Computing Science and Statistics

INTERFACE '91

Proceedings of the 23rd Symposium on the Interface

Critical Applications of Scientific Computing: Biology, Engineering, Medicine, Speech...

April 21–24, 1991

Elaine M. Keramidas
Editor

INTERFACE
FOUNDATION
OF NORTH AMERICA

MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1992	Final 15 Mar 91- 14 Mar 92

4. TITLE AND SUBTITLE

Computing Science and Statistics, Interface'91

5. FUNDING NUMBERS

DAAL03-91-G-0085

6. AUTHOR(S)

J.R. Kettenring (principal investigator)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Interface Foundation of North America, Inc.
Fairfax Station, VA 22039-7460

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARO 28534.1-MA-CF

11. SUPPLEMENTARY NOTES

The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The extremely successful workshop on Computational Molecular Biology featured world-renowned speakers. The workshop served as a focused example of the very real interface between biology, statistics, and computing science. Much of the success of a conference can be measured in terms of the number of attendees and the number of contributed talks, which, for this Symposium, were approximately 400 and 116, respectively. These Proceedings include 78% of the contributed papers and 65% of the invited papers that were given in Seattle - a more than adequate representation of the work presented at the Symposium.

14. SUBJECT TERMS

Symposium, Scientific Computating, Computing Science, Statistics, Biology

15. NUMBER OF PAGES

647

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

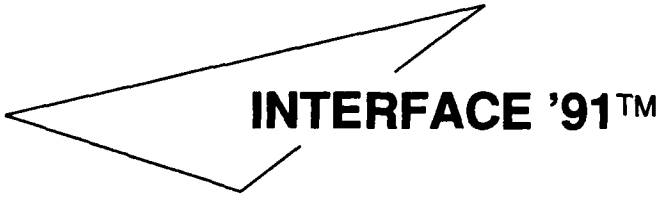
COMPUTING SCIENCE AND STATISTICS

Proceedings of the 23rd Symposium on the Interface

Seattle, Washington, April 21-24, 1991

Editor
ELAINE M. KERAMIDAS
Bellcore, Morristown, New Jersey

Assistant Editor
SELMA M. KAUFMAN
Bellcore, Morristown, New Jersey



INTERFACE FOUNDATION OF NORTH AMERICA

AVAILABILITY OF PROCEEDINGS

22nd (1990)	Springer-Verlag New York, Inc. 175 Fifth Ave. New York, New York 10010
20, 21st (1988, 1989)	American Statistical Association 1429 Duke Street Alexandria, VA 22314-3402
also	Interface Foundation P.O. Box 7460 Fairfax Station, VA 22039-7460
18, 19th (1986, 1987)	ASA 1429 Duke Street Alexandria, VA 22314-3402

Interface Foundation of North America, Inc.
P.O. Box 7460
Fairfax Station, VA 22039-7460

PRINTED IN THE U.S.A.

PREFACE

1991 Interface Proceedings

The 23rd Symposium on the Interface between Computing Science and Statistics was held on April 21-24, 1991, at the Seattle Sheraton Hotel, Seattle, Washington. The conference theme was "Critical Applications of Scientific Computing: Biology, Engineering, Medicine, Speech...". The Symposium was preceded by a workshop on Computational Molecular Biology.

Bellcore hosted the Symposium with Jon R. Kettenring serving as Program Chair. He assembled an outstanding program with a committee that selected topics and invited speakers who collectively made the Symposium a forum for the exchange of exciting new ideas and provided a spectrum of applications for scientific computing. The members of the program committee were Mary Ellen Bock, Andreas Buja, William DuMouchel, Nicholas Fisher, Gene Golub, Joe Hill, John McDonald, John Nash, Daryl Pregibon, Werner Stuetzle, Michael Tarter, Luke Tierney, Paul Tukey, Paul Young, and myself. John Nash devoted much time and effort to organizing a special multi-media session comprised of posters, videos, and demonstrations. Tutorials were presented by Joe Hill, William Eddy and Mark Schervish.

The extremely successful workshop on Computational Molecular Biology was organized by Simon Tavaré and featured world-renowned speakers. The workshop served as a focused example of the very real interface between biology, statistics, and computing science. This theme was evident in the keynote address, "Opportunities for Statisticians and Computer Scientists in Biology", that was presented by Eric Lander. Burton Smith transported those that attended the banquet into the computing world of tomorrow by speaking on "Future Supercomputing". The talks presented in the workshop as well as the keynote and banquet addresses are not included in these Proceedings.

Much of the success of a conference can be measured in terms of the number of attendees and the number of contributed talks, which, for this Symposium, were approximately 400 and 116, respectively. However, a significant indicator of the lasting enthusiasm that remains with the speakers after a conference has ended is their commitment to undertake the task of completing the manuscripts that will comprise the proceedings of that conference. These Proceedings include 78% of the contributed papers and 65% of the invited papers that were given in Seattle - a more than adequate representation of the work presented at the Symposium.

Organizing such a conference is an Herculean feat that necessarily requires the cooperation and dedication of many people. I would like to thank all of those people at Bellcore and the University of Washington who assisted in a myriad of ways. I would also like to thank Selma Kaufman for serving as Assistant Editor of these Proceedings.

Elaine M. Keramidas

Elaine M. Keramidas, Editor

HOST

Bellcore

COSPONSORS OF THE COMPUTATIONAL MOLECULAR BIOLOGY WORKSHOP

National Science Foundation

National Institutes of Health

Department of Energy

COSPONSORS OF THE 23rd SYMPOSIUM ON THE INTERFACE

National Security Agency

Office of Naval Research

Army Research Office

National Science Foundation

National Institutes of Health

Air Force Office of Scientific Research

Department of Energy

COOPERATING SOCIETIES AND INSTITUTIONS

American Statistical Association (ASA)

Institute of Mathematical Statistics (IMS)

Society for Industrial and Applied Mathematics (SIAM)

The Biometrics Society (ENAR and WNAR)

Operations Research Society of America (ORSA)

University of Washington

George Mason University

EXHIBITORS

BMDP Statistical Software

SYSTAT, Inc.

DYAD Software

Brooks Cole Publishing

IBM Corp.

SAS Institute Inc.

IMSL Inc.

Statistical Sciences, Inc.

John Wiley & Sons

Numerical Algorithm Group

Wolfram Research, Inc.

Workshop in Computational Molecular Biology

9:00-9:05	S. Tavaré, University of Southern California. *Introduction*
9:05-9:20	D. Galas, Department of Energy. *Overview*
9:20-10:00	F. Cohen, UC San Francisco. *Computational aspects of the protein folding problem*
10:00-10:40	T. Schlick, NYU. *New computational techniques for computing biomolecular structures and their dynamics*
10:40-11:10	Coffee Break
11:10-11:50	E. Branscomb, Lawrence Livermore National Labs. *Building physical genome maps by random clone overlap; a progress assessment of work on human chromosome 19*
11:50-12:30	E.A. Thompson, University of Washington. *Monte Carlo methods for linkage analysis and complex models*
12:30-1:50	Lunch
1:50-2:30	E.S. Lander, Whitehead Institute. *Dissecting complex inheritance: statistical and computational issues*
2:30-3:10	E. Myers, University of Arizona. *Practical and theoretical advances in sequence comparison*
3:10-3:40	Coffee Break
3:40-4:20	R.J. Roberts, Cold Spring Harbor Labs. *Error detection in DNA sequences*
4:20-5:00	M.S. Waterman, University of Southern California. *Computer methods for locating kinetoplastid cryptogenes*

SYMPOSIUM SCHEDULE

Sunday, April 21, 1991

8:00 a.m. - 9:00 a.m.	Registration (Pre - Function Area)
9:00 a.m. - 5:00 p.m.	Workshop on Computational Molecular Biology (.sp .7en)
5:00 p.m. - 8:00 p.m.	Registration (Area in front of Metropolitan Ballroom)
5:00 p.m. - 8:00 p.m.	Board of Directors' Business Meeting and Dinner (Cedar)
8:00 p.m. - 10:00 p.m.	Opening Reception (Metropolitan Ballroom)

Monday, April 22, 1991

8:30 a.m. - 9:45 a.m.	Keynote Address: "Opportunities for Statisticians and Computer Scientists in Biology" (Grand Ballroom C)
9:45 a.m. - 10:15 a.m.	Break (Grand Ballroom A)
10:15 a.m. - 12:00 p.m.	Invited A : Speech and Language (Grand Ballroom B) Invited B : Scientific Computing Problems in the Aircraft Industry (Metropolitan Ballroom) Invited C : Uncertainty and Graphical Models (West Ballroom) Contributed A : Statistical Graphics (Douglas) Contributed B : Multivariate Analysis (Juniper) Contributed C : Random Number Generators - Simulation (Madrona)
12:00 p.m. - 2:00 p.m.	Lunch
2:00 p.m. - 3:45 p.m.	Invited A : Relational Databases: A Tutorial for Statisticians (Grand Ballroom B) Invited B : Computing Problems in Environmental and Industrial Statistics (West Ballroom) Contributed A : Software Testing (Douglas) Contributed B : Computing and Graphics in Applications (Juniper) Contributed C : Robustness (Madrona)
3:45 p.m. - 4:15 p.m.	Break (Grand Ballroom A)
4:15 p.m. - 6:00 p.m.	Invited A : Massive Databases (Metropolitan Ballroom) Invited B : Engineering Applications of Computing-Intensive Methods (West Ballroom) Invited C : Computational Methods in Spatial Statistics (Grand Ballroom B) Contributed A : Artificial Intelligence - Belief Functions (Douglas) Contributed B : Issues in Interactive Graphics (Juniper) Contributed C : Time Series Prediction - Function Estimation (Madrona)

Tuesday, April 23, 1991

8:00 a.m. - 9:45 a.m.	Invited A : Computationally Intensive Methods for Discrete Data (Grand Ballroom C) Invited B : Data Visualization and Sonification (Grand Ballroom B) Contributed A : Classification - Density Estimation (Douglas) Contributed B : Statistical Inference (Juniper) Contributed C : Genetics - DNA (Madrona)

9:45 a.m. - 10:15 a.m. Break (Grand Ballroom A)

10:15 a.m. - 12:00 p.m. Invited A : Realistic Rendering : A Tutorial for Statisticians (Grand Ballroom B)
Invited B : Computer Modeling, Experimental Design and Data Analysis (Grand Ballroom C)

Contributed A : Neural Nets - Biological Systems (Douglas)
Contributed B : Bootstrap and Related Methods (Juniper)
Contributed C : Optimization - Genetic Algorithms (Madrona)

12:00 p.m. - 2:00 p.m. Poster/Video/Demo Session (.sp .7en)

2:00 p.m. - 3:45 p.m. Invited A : Virtual Interface Technology (Grand Ballroom C)
Invited B : Neural Networks (Grand Ballroom B)
Invited C : Computational Statistical Genetics (East Ballroom)

Contributed A : Tree-Based Methods (Douglas)
Contributed B : Information Retrieval - Record Linkage (Juniper)
Contributed C : Allocation Problems - Sequential Design (Madrona)

3:45 p.m. - 4:15 p.m. Break (Grand Ballroom A)

4:15 p.m. - 6:00 p.m. Invited A : Dynamic Statistical Graphics (Grand Ballroom B)
Invited B : Research Opportunities at the Interface of Biology, Statistics and Computing (Grand Ballroom C)

Contributed A : Integration - Probability Computations (Douglas)
Contributed B : Databases and Information Processing (Juniper)
Contributed C : Problems Relating to Skewness and Kurtosis (Madrona)

6:30 p.m. - 7:30 p.m. Reception (Pre - Function Area)

7:30 p.m. - 10:00 p.m. Banquet (Grand Ballroom C)
Banquet Address: "Future Supercomputing"

Wednesday, April 24, 1991

8:00 a.m. - 9:45 a.m. Invited A : Computational Problems in Biomedical Imaging (Grand Ballroom B)
Invited B : Parallel Computing: A Tutorial for Statisticians (Grand Ballroom C)

Contributed A : Spatial Data - Shape Analysis (Douglas)
Contributed B : Programming Environments (Juniper)
Contributed C : Estimation Problems I (Madrona)

9:45 a.m. - 10:15 a.m. Break (Grand Ballroom A)

10:15 a.m. - 12:00 p.m. Invited A : Multivariate Statistics and Visualization for Labelled Point Data (Grand Ballroom C)
Invited B : Statistical Computing Environments for the 21st Century (Cirrus)
Invited C : Bayesian Computing (Grand Ballroom B)

Contributed A : Image Analysis (Douglas)
Contributed B : Applications Areas (Juniper)
Contributed C : Estimation Problems II (Madrona)

12:00 p.m. End of Conference

TABLE OF CONTENTS

XLV. APPLICATIONS AREAS

XLVI. ESTIMATION PROBLEMS II

Tree-based Models of Speech and Language

Michael D. Riley

AT&T Bell Laboratories
Murray Hill, NJ 07974

Abstract

We describe here the application of classification and regression trees to some problems in speech and language. We begin with a brief overview of the technique. We then describe their application to:

(1) End of sentence detection: The not-so-simple problem of deciding when a period in text corresponds to the end of a declarative sentence (and not an abbreviation) is produced with trees using the Brown corpus as input. The result is 99.8% correct classification.

(2) Segment duration modelling in speech synthesis: 400 utterances from a single speaker and 4000 utterances from 400 speakers were used to build decision trees that predict segment durations based on features such as lexical position, stress, and phonetic context. Over 70% of the durational variance for the single speaker and over 60% for the multiple speakers was accounted by these methods.

(3) Phoneme to phone prediction: A lattice of possible close phonetic transcriptions given a phonemic transcription (from the orthography and a dictionary) is produced using the 4000 TIMIT database as input. The most likely phone corresponding to a phoneme can be predicted 83% correctly. The five most likely phones can be predicted 99% correctly.

1. Introduction

Several applications of statistical tree-based modelling are described here to problems in speech and language. Classification and regression trees are well suited to many of the pattern recognition problems encountered in this area since they (1) statistically select the most significant features involved, (2) provide "honest" estimates of their performance, (3) permit both categorical and continuous features to be considered, and (4) allow human interpretation and exploration of their result. First the method is summarized, then its application to end-of-sentence detection in text, phonetic segment duration prediction, and phoneme-to-phone classification are described. We conclude with some general remarks on the strengths and shortcomings of this method. For other applications to speech and language, see [Lucassen 1984], [Bahl, et al 1987].

2. Classification and Regression Trees

An excellent description of the theory and implementation of tree-based statistical models can be found in *Classification and Regression Trees* [L. Breiman, et al, 1984]. A brief introduction to these ideas will be provided in this section for those who may not be familiar with them.

Consider the not-so-simple problem for deciding when a period in text corresponds to the end of a declarative sentence. This is not as trivial a classification problem as it may first seem. While a period, by convention, must occur at the end of a declarative sentence, one can also occur in abbreviations. Abbreviations can also occur at the end of a sentence. The tagged Brown corpus [Kucera and Francis 1967] of a million words indicates that about 90% of periods occur at the end of sentences, 10% at the end of abbreviations, and about 1/2% in both. The two space rule after an end stop is often ignored and is never present in many text sources (e.g., the AP news).

Figure 1 shows a classification tree for this problem trained on the Brown corpus. Let us first see how to use such a tree for classification. Then we will see how the tree was generated.

The decision of when a period occurs at the end of a sentence will depend on factors such as whether the word following the period is capitalized or if the word containing the period is a common abbrevation. Suppose we see the text fragment "Smith. The". Does the period after "Smith" occur at the end of a sentence?

Starting at the root node in Figure 1, the first decision is whether the word after the period, "the" (case ignored here), is more likely than 27% of the time to occur at the beginning of a sentence relative to its frequency in text. The answer is no (estimated from a database described below), so the left branch is taken. The next split is whether the word containing the period, "smith", is more likely than 1% to occur at the end of a sentence relative to its frequency in text. The answer is yes, so the right branch is taken. The next split concerns the case of the word after the period. Since it is a capitalized word the left split is taken. Finally, the last question is whether the word containing the period, "Smith", is one of several common abbreviation types. Since it is not, the left branch is taken to a terminal node that classifies this

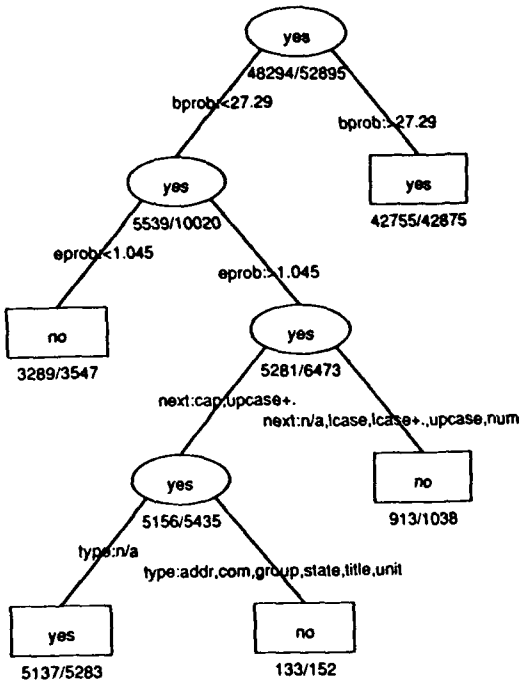


Figure 1. *Classification tree for end-of-sentence detection.*

case as indeed at the end of a declarative sentence.

In the training set, 5137 of the 5283 examples that reached this node were correctly classified. This tree is a subtree of a better classifier to be described in the next section; this example was pruned for illustrative purposes.

This is an example of a *classification* tree, since the decision is to choose one of several classes; in this case, there are two classes: $\{end-of-sentence, not-end-of-sentence\}$. In other words, the predicted variable, y, is categorical. Trees can be created for continuous y also. In this case they are called regression trees with the terminal nodes labelled with a real number (or, more generally, a vector).

Classifying with an existing tree is easy; a more difficult issue is how to generate the tree for a given problem. There are three basic questions that have to be answered when generating a tree: (1) what are the splitting rules, (2) what are the stopping rules, and (3) what prediction is made at each terminal node?

Let us begin answering these questions by introducing some notation. Consider that we have N samples of data, with each sample consisting of M features, $x_1, x_2, x_3, \ldots x_m$. In the end-of-sentence detection example, x_1 might be the case of the word following the period, x_2 the probability that the following word begins a sentence, etc. Just as the y (dependent) variable can be continuous or categorical, so can the x (independent) variables. E.g., word case is categorical (can not be usefully ordered). while beginning word probability is continuous.

The first question — what stopping rule? — refers to what split to take at a given node. It has two parts: (a) what candidates should be considered, and (b) which is the best choice among candidates for a given node?

A simple choice is to consider splits based on one x variable at a time. If the independent variable being considered is continuous $-\infty \leq x < \infty$, consider splits of the form:

$$x \leq k \quad \text{vs.} \quad x > k, \quad \forall k.$$

In other words, consider all binary cuts of that variable. If the independent variable is categorical $x \in \{1, 2, ..., n\} = X$, consider splits of form:

$$x \in A \quad \text{vs.} \quad x \in X - A, \quad \forall A \subset X.$$

In other words, consider all binary partitions of that variable. More sophisticated splitting rules would allow combinations of a such splits at a given node; e.g., linear combinations of continuous variables, or boolean combinations of categorical variables.

A simple choice to decide which of these splits is the best at a given node is to select the one that minimizes the estimated classification or prediction error after that split based on the training set. Since this is done stepwise at each node, this is not guaranteed to be globally optimal even for the training set.

In fact, there are cases where this is a bad choice. Consider Figure 2, where two different splits are illustrated for

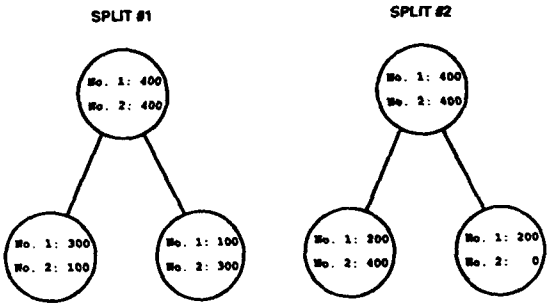


Figure 2. *Two different splits with the same misclassification rate [after Breiman, et al. 1984].*

a classification problem having two classes (No. 1 and No. 2) and 800 samples in the training set (with 400 in each class). If we label each child node according to the greater class present there, we see that the two different splits illustrated both give 200 samples misclassified. Thus, minimizing the error gives no preference to either of these splits [after Breiman, et al. 1984].

The split on the right, however, is better because it cre-

ates at least one very pure node (no misclassification) which needs no more splitting. At the next split, the other node can be attacked. In other words, the stepwise optimization makes creating purer nodes at each step desirable. A simple way to do this is to minimize the entropy at each node for categorical y. Minimizing the mean square error is a common choice for continuous y.

The second question — what stopping rule? — refers when to declare a node terminal. Too large trees may match the training data well, but they won't necessarily perform well on new test data, since they have overfit the data. Thus, a procedure is needed to find an "honest-sized" tree.

Early attempts at this tried to find good stopping rules based on absolute purity, differential purity from the parent, and other such "local" evaluations. Unfortunately, good thresholds for these are hard to find and vary from problem to problem.

A better choice is as follows: (a) grow an over-large tree with very conservative stopping rules, (b) form a sequence of subtrees, $T_0, \ldots, T_n$, ranging from the full tree to just the root node, (c) estimate an "honest" error rate for each subtree, and then (d) choose the subtree with the minimum "honest" error rate.

To form the sequence of subtrees in (b), vary α from 0 (for full tree) to ∞ (for just the root node) in:

$$\min_T [R(T) + \alpha |T|].$$

where $R(T)$ is the classification or prediction error for that subtree and $|T|$ is the number of terminal nodes in the subtree. This is called the cost-complexity pruning sequence.

To estimate an "honest" error rate in (c), test the subtrees on data different from the training data, e.g., grow the tree on 9/10 of the available data and test on 1/10 of the data repeating 10 times and averaging. This is often called cross-validation.

Figure 3 shows misclassification rate vs. tree length for the end-of-sentence classification problem using a subset of the input features describe below. The bottom curve shows misclassification for the training data, which continues to improve with increasing tree length. The higher curve shows the cross-validated misclassification rate, which reaches a minimum with a tree size of about 20 and then rises again with increasing tree length.

The last question — what prediction is made at a terminal node? — is easy to answer. If the predicted variable is categorical, choose the most frequent class among the training samples at that node (plurality vote). If it is continuous, choose the mean of the training samples at that node.

The approach described here can be used on quite large problems. We have grown trees with hundreds of thousands of samples with a hundred different independent variables. The (expected) time complexity, in fact, grows only linearly with the number of input variables (worst case is quadratic). The one expensive operation is forming all binary partitions for categorical $x's$. This increases exponentially with the number of distinct values the variable can assume.

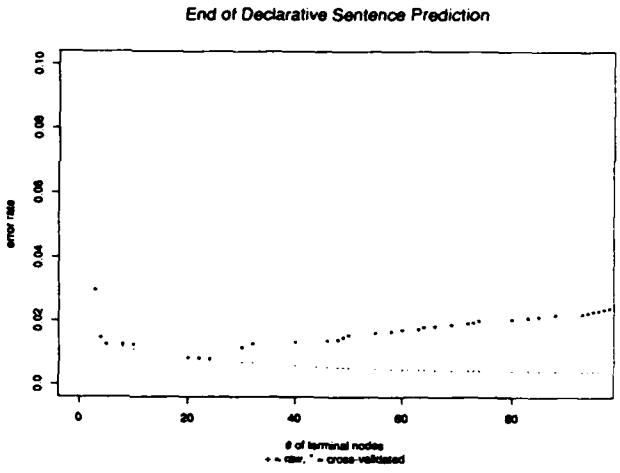


Figure 3.

Let us now discuss in detail the applications of these ideas to some problems in speech and language.

3. End of sentence detection

As the first example, let us look again at the end-of-sentence detection problem described above. A more comprehensive tree was generated using the the following features:

- Prob[word with "." occurs at end of sentence]
- Prob[word after "." occurs at beginning of sentence]
- Length of word with "."
- Length of word after "."
- Case of word with ".": Upper, Lower, Cap, Numbers
- Case of word after ".": Upper, Lower, Cap, Numbers
- Punctuation after "." (if any)
- Abbreviation class of word with ".":

  — e.g., month name, unit-of-measure, title, address name, etc.

The choice of these features was based on what humans appear to use (at least when constrained to looking at a few words around the "."). Facts such as "Is the word after the '.' capitalized?", "Is the word with the '.' a common abbreviation?", "Is the word after the "." likely found at the beginning of a sentence?", etc. can be answered with these features.

The word probabilities indicated above were computed from the 25 million words of AP news, a much larger (and independent) text database. (In fact, these probabilities were for the beginning and end of paragraphs, since these are explicitly marked in the AP, while end of sentences, in general, are not.)

The resulting classification tree correctly identifies whether a word ending in a "." is at the end of a declarative sentence in the Brown corpus with 99.8% accuracy. The majority of the errors are due to difficult cases, e.g. a sentence that ends with "Mrs." or begins with a numeral (it can happen).

4. Segment duration modelling for speech synthesis

400 utterances from a single speaker and 4000 utterances from 400 speakers (the TIMIT database [Fisher, et al. 1987]) of American English, both which are manually hand-segmented and phonetically labelled, were used separately to build regression trees that predict the duration of the phonetic segments. Predicting these durations is important both in work on speech synthesis and recognition. The following features were used:

- Segment Context:
  - — Segment to predict
  - — Segment to left
  - — Segment to right
- Stress (0, 1, 2)
- Word Frequency: (rel. 25M AP words)
- Lexical Position:
  - — Segment count from start of word
  - — Segment count from end of word
  - — Vowel count from start of word
  - — Vowel count from end of word
- Sentence Position:
  - — Word count from start of sentencee
  - — Word count from end of sentencee
- Dialect: N, S, NE, W, SMid, NMid, NYC, Brat
- Speaking Rate: (rel. to calibration sentences)

Coding the phonetic context required special considerations since more than 50 phones (using the TIMIT labelling) can precede a stop in this context. If this were treated as a single feature, more than 2^{50} binary partitions would have to be considered for this variable at each node, clearly making this approach impractical. Chou [1987] proposes one solution, which is to use k-means clustering to find sub-optimal, but good paritions in linear complexity.

The solution adopted here is to classify each phone in terms of 4 features, *consonant manner*, *consonant place*, *"vowel manner"*, *and "vowel place"*, each class taking on about a dozen values. Consonant manner takes on the usual values as *voiced fricative, unvoiced stop, nasal, etc.* Consonant manner takes on values such as *bilabial, dental, velar, etc.* "Vowel manner" takes on values such as *monopthong, diphthong, glide, liquid, etc.* and "vowel place" takes on values such as *front-low, central-mid-high, back-high, etc.* All can take on the value *n/a* if they do not apply; e.g., when a vowel is being represented, consonant manner and place are assigned *n/a*. In this way, every segment is decomposed into four multi-valued features that have acceptable complexity to the classification scheme and that have some phonetic justification.

The word frequency was included as a continuoulsy graded "function word" detector and was based on six months of AP news text. The stress was obtained from a dictionary (which is easy, but imperfect). The last two features were used only for the multi-speaker database. The dialect information was coded with the TIMIT database. The speaking rate is specified as the mean duration of the two calibration sentences, which were spoken by every speaker.

Over 70% of the durational variance for the single speaker and over 60% for the multiple speakers were accounted for by these trees. Figure 4 shows durations and duration residuals for all the segments together. The large tree sizes here, many hundreds of nodes, make them somewhat uninteresting to display.

These trees were used to derive durations for a text-to-speech synthesizer. This approach offers a promising alternative to heuristically derived duration rules [e.g., Klatt

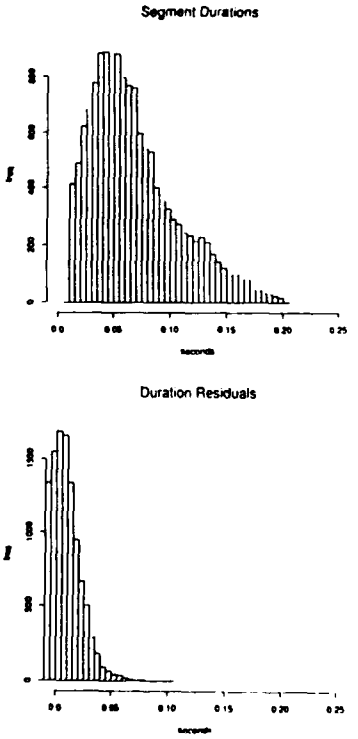


Figure 4.

1976]. Since tree building and evaluation is rapid once the data are collected and the candidate features specified, this technique can be readily applied to other feature sets and to other languages.

This approach is very data-intensive, though. Our databases have tens or hundreds of thousands of segments. We believe really good duration modelling will involve at least an order of magnitude more data. This presents not so much a computational problem, given the efficient algorithms for tree construction available, but a data collection problem. We believe that automatic transcription [Ljolje and Riley] may ultimately be the way to proceed.

5. Phoneme-to-phone prediction

The task here is given a phonemic transcription of an utterance, e.g., based on dictionary lookup, predict the phonetic realization produced by a speaker [see also Lucassen, et. al. 1984; Chou, 1987; Riley 1989, 1991; Chen 1990; Randolph 1990]. For example, when will a T be flapped (as in American English pronunciation of 'pretty') or released (as in phrase-initial T's). We used the following features to decide this problem extracted from the TIMIT database:

- Phonemic Context:
  - — Phoneme to predict
  - — Three phonemes to left
  - — Three phonemes to right
- Stress (0, 1, 2)
- Word Frequency: (rel. 25M AP words)
- Dialect: N, S, NE, W, SMid, NMid, NYC, Brat
- Lexical Position:
  - — Phoneme count from start of word
  - — Phoneme count from end of word
- Phonetic Context: phone predicted to left

The phonemic context was coded in a seven segment window centered on the phoneme to realize, again using the 4 feature decomposition described above. The other features are similar to the duration prediction problem. Ignore the last feature, for the moment.

The tree for all phonemes grown on these features predicts on the average 83% of the TIMIT labellings exactly. A large percentage of the errors are on the precise labelling of reduced vowels as either IX or AX.

A list of alternative phonetic realizations can also be produced from the tree, since the relative frequencies of different phones appearing at a given terminal node can be retained. Figure 5 shows such a listing for the utterance, *Would your name be Tom?* . (We use the TIMITBET phonetic symbols in these examples [Fisher, et al. 1987]). It indicates, for example, that the D in "would" is most likely uttered as a DCL JH in this context (59% of the time), followed by DCL D (28%). On the average five alternatives per phoneme are sufficient to cover 99% of the possible phonetic realizations. This can be used, for example, to greatly constrain the number of alternatives that must be considered in automatic segmentation when the orthography is known.

Would your name be Tom?

Phoneme	Prob	Phone						
w	97.9	w	1.7	-				
uh	79.9	uh	9.2	ix	2.2	uw	2.0	ax
d	59.4	dcl jh	28.1	dcl d	9.4	dcl	3.1	jh
y	76.1	y	22.8	-				
uh	79.9	uh	9.2	ix	2.2	uw	2.0	ax
er	52.6	axr	23.2	r	15.8	er	6.3	-
n	79.8	n	18.6	nx	1.5	-		
ey	95.7	ey	1.3	eh	0.8	ih	0.7	ix
m	96.1	m	3.4	-				
b	87.5	bcl b	4.5	pau b	3.9	bcl	2.5	b
iy	90.5	iy	4.9	ix	2.3	ih	1.2	-
t	92.9	tcl t	5.6	dx	0.6	t		
aa	82.3	aa	7.4	ao	3.4	axr	2.2	ah
m	96.1	m	3.7	-				

Figure 5. *Phonetic alternatives for "Would your name be Tom?*

These a *priori* probabilities, however, do not take into account the *phonetic* context, only the *phonemic*. For example, if DCL JH is uttered for the phoneme D in the example in Figure 5, then the Y is most likely deleted and not uttered. However, the overall probability that a Y is uttered in that phonemic context (averaging both D going to DCL JH, D, etc.) is greatest. The point is that to incorporate the fact that "D goes to DCL JH implies Y usually deletes" is that *transition* probabilities should be taken into account.

This can be done by including an additional feature for the phonetic identity of the previous segment. The output listing then becomes a transition matrix for each phoneme. The best path through such a lattice can be found by dynamic programming.

This, coupled with a dictionary, can also be used for letter-to-sound rules for a synthesizer (when the entry is present in the dictionary). The effect of using the TIMIT database for this purpose is a somewhat folksy sounding synthesizer. Having the D "Would your" uttered as a JH may be appropriate for fluent English, but it sounds a bit forced with existing synthesizers. Too much else is wrong. A very carefully uttered database by a professional speaker would give better results for this application of the phoneme-to-phone tree.

6. Discussion

On the whole, we have found classification and regression trees quite useful in modelling a variety of phenonema in speech and language. In part, it is their ability to han-

dle both categorical and continuous inputs and outputs that makes them attractive to us. The fact that they offer efficient algorithms, a well-established cross-validation procedure, and a relatively perspicuous representation makes them more appealing to us than, say, back-propogation neural networks for the problems we have described.

The principal difficulty we have found with this and similar statistical approaches is that while the trees classify well most of the time, they occasionally make egregious errors. When noticed, it is possible to correct these errors by hand modification of the trees. This is, however, quite tedious. Further, if new data are used or new input features are tried, the editing has to be redone (if the error remains).

What would be most appealing to us would be techniques that would allow easy mixing of statistical learning with hand specification. The user could hand specify what he is sure of and leave to the statistics to fill in the rest the best it can, letting us have our cake and eat it too.

7. References

Bahl, L., et. al. 1987. A tree-based statistical language model for natural language speech recognition. *IBM Researh Report 13112.*

Brieman, L., et. al. 1984. *Classification and regression trees.* Monterey, CA: Wadsworth & Brooks.

Chen, F. 1990. Identification of contextual factors for pronunciation networks. *Proc. ICASSP '90.* S14.9.

Chou, P. 1988. *Applications of information theory to pattern recognition and the desing of decision trees and trellises.* Ph.D. thesis, Stanford University, Stanford, CA.

Fisher, W., Zue, V., Bernstein, D. and Pallet, D. 1987. An acoustic-phonetic data base. *J. Acoust. Soc. Am.* **81.** Suppl. 1.

Klatt, D. 1976. Linguistic uses of segmental duration in English: acoustic and perceptual evidence. *J. Acoust. Soc. Am.* **59.** 1208-1221.

Kucera, H. and W. Francis. 1967 *Computational analysis of present-day Americal English.* Providence, R.I.. Brown Univ. Press.

Ljolje, A. and Riley, M. 1990. Automatic segmentation and labeling of speech. *Proc. ICASSP '91.* S7.17.

Lucassen, J.M. & Mercer, R.L. 1984. An information theoretic approach to the automatic determination of phonemic baseforms. *Proc. ICASSP '84.* 42.5.1-42.5.4.

Randolph, M. 1990. A data-driven method for discover and predicting allophonic variation. *Proc. ICASSP '90.* S14.10.

Riley, M. 1989. Some applications of tree-based modelling to speech and language. *Proc. DARPA Speech and Natural Language Workshop.* Oct 1989, Cape Cod, MA, pp. 339-352.

Riley, M. 1991. A statistical model for generating pronunciation networks. *Proc. ICASSP '91.* S11.1.

Some Statistical Opportunities in Speech and Language

Kenneth Ward Church
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
church@isi.edu

Abstract

Text analysis is a hot topic, and for good reason. Text is more available than ever before. Just ten years ago, the one-million word Brown Corpus (Francis and Kucera, 1982) was still considered large, but even then, there were much larger corpora in use such as the 18 million word Birmingham Corpus (Sinclair 1987a, 1987b). These days, there are many places that regularly use samples of text running into the hundreds of millions of words. And it is very likely that billions of words will be available very soon.

All of this data provides a great research opportunity; it easier these days to corpus data much more effectively than it was in the 1950s, the last time that empiricism was in fashion. Text analysis focuses on broad (though possibly superficial) coverage of unrestricted text, rather than a deep analysis of a restricted domain. This pragmatic view toward coverage and performance distinguishes text analysis from so-called "intelligent" approaches such as natural language understanding. This approach has produced a number of tools such as spelling correctors and part of speech taggers that work on unrestricted text, with reasonable accuracy and efficiency.

1. Recognition Applications

Recognition applications are perhaps the most obvious applications for large bodies of text. Three examples of recognition applications will be mentioned here: (1) Speech Recognition, (2) Optical Character Recognition (OCR), and (3) Spelling Correction.

Imagine a noisy channel, such as a speech recognition machine that almost hears, an optical character recognition (OCR) machine that almost reads, or a typist that almost types. Good text (W_i) goes into the channel, and corrupted text (W_o) comes out the other end.

$$W_i \rightarrow \textit{Noisy Channel} \rightarrow W_o$$

How can an automatic procedure recover the good input text, W_i, from the corrupted output, W_o? In principle, one can recover the most likely input by hypothesizing all possible input texts, W_i, and selecting the input text with the highest score. Using a classic Bayesian argument, the score is computed by taking the product of the prior probability, $Pr(W_i)$, and the channel probability, $Pr(W_o \mid W_i)$. This procedure can be written as:

$$\underset{W_i}{ARGMAX}\, Pr(W_i)\, Pr(W_o \mid W_i)$$

where ARGMAX finds the argument with the maximum score.

The prior probability, also known as the language model, is the probability that the W_i would be input to the channel. For example, in the speech recognition application, it is the probability that someone would utter W_i, whereas in the spelling correction application, it is the probability that someone would type W_i. In practice, the prior is approximated by computing various statistics over a large sample of text.

The channel probability is the probability that the channel would transform the word sequence W_i into the sequence W_o. This is relatively high if W_i is the same as or very "similar" to W_o, where the definition of "similar" depends on the application. The channel for speech recognition, for example, will have a high probability of mapping words that sound similar (e.g., "writer" and "rider" in many American dialects) into the same output representation. However, in other applications such as optical character recognition, "writer" and "rider" are unlikely to be confused by the channel because these words are optically quite distinct. Thus, the channel model clearly depends on the application as illustrated in the Table 1.

Table 1: Examples of Channel Confusions in Different Applications

Application	Input	Output
Speech Recognition	writer	rider
	here	hear
Optical Character Recognition	all	all (*A-one-L*)
	of	o{
	form	farm
Spelling Correction	government	goverment
	occurred	occured
	commercial	commerical

It is convenient to partition the prior and the channel in this way, so that the same prior can be used for a variety of recognition applications including speech recognition, optical character recognition and spelling correction. The channel, of course, generally cannot be ported from one application to another.

2. Spelling Correction

I have found that spelling correction is a good application to look at because it is analogous to many important recognition applications based on a noisy channel model (such as speech recognition), though somewhat simpler and therefore possibly more amenable to detailed statistical analysis. In (Kernighan, Church, and Gale, 1990), we described a program called *correct* which inputs a misspelled word such as *absurb*, and outputs a list of candidate corrections sorted by probability: *absorb* (56%), *absurd* (44%). The probability scores are the novel contribution; there are have been many programs in the past that generated a (long) list of candidate corrections, but few have attempted to score the candidates by a stochastic model of the prior probability of observing the candidate correction $Pr(c)$ and a channel probability of observing a particular typo given the candidate correction $Pr(t \mid c)$. Both of these probabilities were estimated from about 50 million words of Associated Press newswire (which includes about 15,000 typos which are used to train the channel model).

In evaluating the program, we restricted our attention to 564 typos that had exactly two candidate corrections. A panel of three judges were given the typo (e.g., *absurb*), the two candidate corrections (e.g., *absorb* and *absurd*) and a concordance line (e.g., *it is absurb and probably obscene for...*), and were asked to select one of the two corrections (or none-of-the-above). The judges found this task more difficult than they had anticipated, and very time consuming (it took each judge about four hours to grade the 564 examples). In addition, the judges felt that the task would have been much harder without the concordance line, suggesting that context should be incorporated into the program.

Table 2 shows that *correct* agrees with the majority of the judges in 87% of the 332 cases of interest.[1] In order to help calibrate this result, we compared *correct* to three inferior methods: *channel-only*, *prior-only* and *chance*. Table 2 shows that both the channel-only and the prior-only models provide a significant contribution over chance, and that *correct*, which is a combination of the two, is significantly better than either in isolation.

[1] We restricted our attention to those cases where at least two judges selected one of the two candidate corrections, and they agreed with each other.

Table 2 also shows that the judges are significantly better than all of the programs, indicating that there is room for improvement.

Table 2: Evaluation of *Correct*

Method	Discrimination	%
correct	286/329	87 ±1.9
Judge 1	271/273	99 ±0.5
Judge 2	271/275	99 ±0.7
Judge 3	271/281	96 ±1.1
channel-only	263/329	80 ±2.2
prior-only	247/329	75 ±2.4
chance	172/329	52 ±2.8

The program, of course, is not making use of context whereas the human judges did have access to a concordance line. The following examples show that the task is extremely difficult without context.

Table 3: Hard without Context

Typo	Choice 1	Choice 2
actuall	actual	actually
constuming	consuming	costuming
conviced	convicted	convinced
confusin	confusing	confusion

Of course, the task becomes much easier if the context is provided as demonstrated by the following four concordance lines.

1. in determining whether the defendant ***actuall*** will die. In the 1985 decision, the...
2. on Friday night, a show as lavish in ***constuming*** and lighting as those the late Liberace used to...
3. of the area. "When we're ***conviced*** and the Peruvians are convinced (the base camp)...
4. The political situation grew more ***confusin*** today, with an official media report indicating...

Both (Mays *et al.*, 1990) and (Church and Gale, 1991a) have found that statistical n-gram models of context can help considerably, although performance is still far below that of the human judges. A quick look at the concordance lines above shows (a) that the relevant contextual clues are often fairly close to the typo, and (b) that there are relatively few cases that make use of long-distance syntactic dependencies. (a) suggests that simple n-gram methods might work fairly well in many cases, and (b) suggests that more complicated "intelligent" parsing methods might not be worth the trouble.

3. The Trigram Model

One of the simpler and more popular priors is the n-gram model. This model makes the simplifying assumption that word probabilities depend on only the previous $n-1$ words, and that long-distance dependences

which extend beyond this limited window can be ignored. Jelinek (1985) uses the example shown in Table 4 to illustrate the power of the trigram model. In the sentence, *We need to resolve all the important issues within the next two days*, most of the words are extremely predictable from the trigram context (the current word plus the previous two). Note that *we* is the 9th most likely word to begin a sentence in his model; the words *the, this, one, ..., in* are more likely to begin a sentence than *we*. The word *need* is found to be the 7th most likely word to follow *we*; the words *are, will, ..., do* are more likely than *need*. And so on. Jelinek uses this example to argue that the rank is usually very small in comparison to the vocabulary size, which was 20,000 words in this example.

Table 4: Example of Trigrams (Jelinek, 1985)

The This One Two A Three Please In *We*	9
are will the would also do *need*	7
to	1
know have understand ... *resolve*	98
the this these problems ... *all*	9
issues problems *the*	3
necessary data information ... *important*	641
role thing that ... *issues*	9
and from in to are with ... *within*	66
the	1
next	1
be *two*	2
meeting months years ... *days*	7

Note that function words (e.g., *to, the*) are generally more predictable than content words (e.g., *resolve, important*). This turns out to be important in speech recognition because the shorter function words are more easily confused by the channel model and so it is fortunate that they are more predictable from context.

Some of the content words also have relatively small ranks. Consider, the content word *issues*, for example. It turns out that there are relatively few words that follow the word *important* (at least, in the subdomain of IBM office correspondences). This kind of *collocational* (or co-occurrence)[2] constraint between words are often not captured very well with a syntactic parser. Perhaps this is the reason why trigram models have tended to out-perform so-called "intelligent" approaches, when performance is measured in terms of entropy.

4. Word Frequencies and Word Association Norms

The trigram model does a good job of modeling word frequencies which are very important, as any psycholinguist knows. Generally speaking, subjects respond more quickly and more accurately to a high frequency word (e.g., a word that appears relatively often in a sample of text such as the Brown Corpus) than to an unusual low frequency word. The word association effect is similar except that it involves pairs of words. In general, subjects respond more quickly and more accurately to a word like *doctor* if it follows a highly associated word such as *nurse* (Meyer, Schvaneveldt and Ruddy, 1975, p. 98).

Word frequencies are fairly easy to estimate from a sample of text such as the Brown Corpus. Hanks and I have argued that word associations should also be estimated by computing various statistics over large corpora (Church and Hanks, 1990). It is more common in the psycholinguistic literature to find a study like (Palermo and Jenkins, 1964); they estimated word association norms for 200 words by asking a few thousand subjects (psychology undergraduates) to write down a word after each of the words to be measured. Results were reported in tabular form, indicating which words were written down, and by how many subjects, factored by grade level and sex. The word *doctor*, for example, is reported on pp. 98-100, to be most often associated with *nurse*, followed by *sick, health, medicine, hospital, man, sickness, lawyer*, and about 70 more words.

5. Strengths and Weaknesses

The main advantage of the trigram model is that it has very low entropy, 1.76 bits per character (Brown *et al.*, 1991). Parsers generally don't do as well because they tend to ignore word frequencies. The trigram model is also able to capture some collocations and word associations.

The most obvious weakness with the trigram model is the lack of syntax; the model makes no attempt to capture long-distance dependencies such as syntactic agreement, conjunction and wh-movement. In fact, the lack is syntax is probably not the most serious problem with the model. The sparse-data problem is extremely serious since many trigrams do not appear very often in the training corpus, if at all. In addition, the trigram model assumes that trigrams have a binomial distribution, an assumption which is often violated in practice.

[2] Halliday (1966, p. 150) was very interested in the difference between *strong* and *powerful*. Although both words have very similar syntax and semantics, there do seem to be some contexts where one word is much more appropriate than the other, e.g., *strong tea* vs. *powerful drugs*. The terms *collocation, co-occurrence* and *lexis* have been used to describe these kinds of constraints on pairs of words.

6. Parsers May Not Help Very Much

It has been common practice, especially during the first Darpa Speech Understanding Project (Klatt, 1977), to try to use a syntactic parser to take advantage of contextual constraints. Unfortunately, there has not been very much success. If I tell you that the next word is going to be a noun, then I really haven't told you very much. The following example illustrates the problem.

In the Optical Character Recognition (OCR) application, it is likely that the words *form* and *farm* might be confused by the channel model. Imagine, for example, that they were found in one of the following two contexts:

$$federal \left(\begin{array}{c} farm \\ form \end{array} \right) credit$$

$$some \left(\begin{array}{c} farm \\ form \end{array} \right) of$$

Most people would have little trouble deciding that *farm* is much more likely in the first context and that *form* is much more likely in the second context. In fact, trigram models also have little difficulty with this example. However, a syntactic parser wouldn't help very much. The parser might tell us that the missing word is a noun, but that wouldn't help distinguish between *form* and *farm* because they are both nouns. In general, if one were to compare the relative importance of local context versus long-distance dependencies, one would almost certainly find that the local context is much more important, at least in terms of predicting the next word.

The linguistic notion of syntax (constraints on nouns, verbs, subjects, objects, phrases, etc.) was not intended to be used in a noisy channel model. Chomsky has always been more interested in linguistic *competence* (an idealization of syntax) than *performance* (deviations that are found in the real world including: word frequencies, word association norms, collocations, statistical preferences, memory and computational limitations, etc). It should not be surprising that performance issues are important in recognition applications, and consequently, models that are based too closely on idealized notions of syntactic competence are likely to run into trouble when they are tested on real data.

7. Entropy

It is common practice to evaluate a language model on the basis of its entropy. The standard ascii code uses 8 bits to represent a character. Obviously, many of these bits are unnecessary since some letters are much more common than others. If one were to take advantage of letter frequencies using a Huffman code to encode each letter one at a time, then it would take about 5 bits to code each character. This very simple code does almost as well as the Unix(TM) *compress* program, which uses the Lempel-Ziv algorithm (Welch, 1984).

In general, models based on words achieve much better compression than models based on characters. A unigram model (a Huffman code based on word probabilities) requires about 2.1 bits per character (Brown, personal communication). Note that the unigram model out-performs Lempel-Ziv by a considerable margin, indicating that the standard Unix(TM) compress program could be improved significantly.

The trigram model achieves even better compression, 1.76 bits per character (Brown *et al.*, 1991). This last model is remarkably close to Shannon's estimate for the entropy of English. However, it isn't exactly fair to compare these estimates since Shannon's estimate was based on a 27 character alphabet whereas these other estimates are based on a 256 character alphabet. Nevertheless there does seem to be some reason to believe that the trigram model is doing quite well, and that it might be almost as good as native speakers in predicting the next letter.

Table 5: Entropy of Various Language Models

Model	Bits / char
Ascii	8
Huffman code each char	5
Lempel-Ziv (Unix(TM) compress)	4.43
Unigram	2.1
Trigram	1.76
Shannon's Estimate	1.25

8. Sparse Data "Fixes"

As mentioned above, the sparse data problem is probably the most serious weakness with the trigram model. In fact, there are usually many more parameters than data points. Let V be the number of types in the vocabulary and N be the number of tokens in the corpus. Then there are V^3 parameters, which is generally much much larger than N, the size of the training set. For example, in the Brown Corpus, there are $V^3 \approx 1.25 \times 10^{14}$ trigrams, and only $N \approx 10^6$ tokens to train from. Obviously, most of the possible trigrams will not be observed in the training corpus.

One might think that one could fix the sparse data problem by collecting more data, but ironically V^3 generally grows much faster than N. That is, if you collect a larger corpus (more tokens), then you will also find more types (vocabulary items). It isn't exactly clear how these two function grow, but I believe that the vocabulary grows almost linearly with corpus size. In any case, V^3 grows much much faster than N, so collecting more data is not a solution to the sparse data

problem.

Something has to be done about the sparse data. Katz (1987) suggests "backing-off" from the trigram estimates when there isn't enough data. Basically, the idea is to replace trigram estimates with a combination of unigram, bigram and trigram estimates. This is obviously a good idea.

One can also try to reduce the number of parameters by grouping words into classes (e.g., parts of speech, synonym sets, etc.) Brown *et al.* (1990b) suggest building classes with a self-organizing procedure which joins words based on a mutual information criterion. The criterion has the effect of joining together words that have similar distributions (e.g., days of the week, months of the year, etc). Although this particular suggestion is very intriguing, it probably won't help too much with the sparse data problem because it isn't possible to determine that two words have a similar distribution unless you have a fair number of examples of both words. The real problem is what to do with words that you haven't seen very often in the training set. Worse, what do you do with words that you haven't seen at all. The criterion for joining words cannot depend on data that is unavailable.

9. MLE, ADD1, GT and HO

Finally, one can "adjust" frequency counts, especially when they are small. In principle, n-gram probabilities can be estimated from a large sample of text by counting the number of occurrences of each n-gram of interest and dividing by the size of the training sample. This method, which is known as the "Maximum Likelihood Estimator," (MLE) is very simple. However, it is unsuitable because n-grams which do not occur in the training sample are assigned zero probability. This is qualitatively wrong for use as a prior model, because it would never allow the n-gram, while clearly some of the unseen n-grams will occur in other texts. For non-zero frequencies, the MLE is quantitatively wrong.

Three alternatives will be mentioned here. These methods all take the observed counts (r) and produce an adjusted count (r^*). The last two methods also make use of N_r, the number of types that occur exactly r times.

$$r^* = r \qquad \text{MLE}$$

$$r^* = (r+1)\frac{N}{N+S} \qquad \text{ADD1}$$

$$r^* = (r+1)\frac{N_{r+1}}{N_r} \qquad \text{GT}$$

$$r^* = C_r/N_r \qquad \text{HO}$$

The first method, ADD1 (Jeffreys, 1948), simply adds one to all of the observed counts and then adjusts the total appropriately by multiplying by $N/(N+S)$ where S is the number of types (e.g., V^3). This method is generally a disaster, especially when S is much larger than N, which is most of the time. In a spelling correction application, Gale and I have found that this method produced very misleading estimates and concluded that estimating the context badly can be worse than not estimating the context at all (Church and Gale, 1990).

The second method, GT (Good, 1953), depends only on the modest assumption that ngrams have binomial distributions. Unfortunately, even this modest assumption turns out to be highly problematic. Words and ngrams are like busses in New York City; they are social animals and like to travel in packs. The word *earthquake*, for example, has a very bursty distribution in the Associated Press (AP) Newswire, depending on whether or not there has recently been an earthquake. The word *turkey* also has a bursty distribution in the AP, with a burst appearing once a year in late November. In fact, one can show that the binomial assumption is often seriously off depending on what happens to be in the news, among other things.

The last method, HO held-out estimate (Jelinek and Mercer, 1985), assumes the least, merely that the training and test corpora are generated by the same process. This method splits the text into two halves and uses the first half to determine N_r, the number of types that occur r times, and the second half to determine their total mass C_r. r^* is then simply set to C_r/N_r. For example, to determine 0*, the adjusted count for ngrams that did not occur in the first half, one would compute C_0, the total count in the second half for ngrams that did not appear in the first half, and divide by N_0, the number of ngram types that did not appear in the first half.

In (Church and Gale, 1991b), we compared the GT and HO methods for estimating bigram frequencies in 22 million words of Associated Press Newswire and found that the GT method was slightly better when the binomial assumption was appropriate. Tables 6 and 7 show that both methods produce remarkably similar estimates for r^*.

Table 6: Good-Turing (GT) Estimate

r	Nr	r*
0	74,671,100,000	0.0000270
1	2,018,046	0.446
2	449,721	1.26
3	188,933	2.24
4	105,668	3.24

Table 7: Held-Out (HO) Estimate

r	Nr	Cr	r*
0	74,671,100,000	2,019,187	0.0000270
1	2,018,046	903,206	.448
2	449,721	564,153	1.25
3	188,933	424,015	2.24
4	105,668	341,099	3.23

The agreement of the two methods, though, is partly due to the fact that we took extraordinary measures to control for the New York City bus effect. That is, we spit the text into two samples by randomly assigning each bigram to one of the two samples. This effectly destroyed any time structure that might have existed in the two samples. If we had split the text into two halves sequently by assigning the first six months of the newswire to the first half and the second six months to the second half, then we would have observed significant differences due to the non-binomial nature of the news.

Table 8 shows that there is considerable agreement when the text is split randomly. The *t*-scores are possibly somewhat larger than we would like, but they are really not too bad considering that we are dealing with extremely infrequent events. The *t*-scores are computed using an estimate of variances which is described in (Church and Gale, 1991b). Table 9 shows that there is considerable disagreement if the texts are split sequentially.

Table 8: Split Text Randomly

r	HO	GT	t
0	.000027041	.000027026	-.7
1	.4476	.4457	-2.9
2	1.254	1.260	2.5
3	2.244	2.237	-1.5
4	3.228	3.236	1.0
5	4.21	4.23	1.8
6	5.23	5.19	-2.8

Table 9: Split Text Sequentially

r	HO	GT	t
0	0.00001684	0.0001132	479.4
1	0.4076	0.5259	113.
2	1.0721	1.2378	47.0
3	1.9742	2.2685	37.8
4	2.8632	3.1868	26.4
5	3.7982	4.2180	25.8
6	4.7822	5.2221	15.4

In summary, there are quite a number of very powerful techniques such as GT and HO for estimating the probability of an n-gram that did not appear very many times in the training corpus, if at all. These methods appear to work remarkably well when the assumptions are met, but unfortunately, there are serious problems with the assumptions. There has recently been some interest in adaptive models, models that can take advantage of recency effects and forgetting effects. If words were binomially distributed, then the probability of a word should be independent of how long it has been since it was last mentioned. In the AP wire, it appears that the probability increases dramatically when a word has been mentioned recently, and drops fairly consistently with the length of time since the last mention.

10. Translation Applications

Section 1 discussed the use of noisy channel methods in recognition applications. This section will show how the same methods can be used to address translation applications such as Machine Translation (MT). The approach was first suggested by Weaver in 1949 and is currently being revived by Brown *et al.* (1990a). If you would like to translate words in a source language, W_s (e.g., French) into words in a target language, W_t (e.g., English), you imagine that the source words W_s were the output of a noisy channel. The translation task is to find the most likely input to the noisy channel given the observed outputs.

$$W_t \rightarrow \textit{Noisy Channel} \rightarrow W_s$$

Viewed in this way, translation is very similar to recognition. In principle, one can recover the most likely input by hypothesizing all possible target language texts, W_t, and selecting the target text with the highest score, where scores are computed by basically the same formula as above:

$$\underset{W_t}{ARGMAX}\, Pr(W_t)\, Pr(W_s \mid W_t)$$

This information theoretic approach to machine translation is extremely controversial among researchers in machine translation because it questions many of the basic assumptions that have dominated the field since the 1950s when Chomsky (1957) and others pointed out that statistical n-gram methods are incapable of modeling certain syntactic constraints such as agreement over long distances. Brown *et al.* (1990a) argue that the statistical approach is more tractable than it was in the 1950s. Computers are certainly faster than they were then. In addition, and probably much more importantly, it is now possible to find large amounts of *parallel text*, text such as the Canadian parliamentary debates which are available in multiple languages. Brown *et al.* estimate $Pr(W_t)$ and $Pr(W_s \mid W_t)$ by computing various statistics over these parallel texts. Although the approach may be deeply flawed for many of the reasons

that were discussed in the 1950s, there is, nevertheless, a growing community of researchers in corpus-based linguistics such as (Klavans and Tzoukermann, 1990) who are becoming convinced that the approach is worth pursuing because there is a very good chance that it will produce a number of lexical resources that could be of great value to their research.

11. Part of Speech Tagging

This description of the machine translation problem is fairly general and can be applied to quite a number of transduction problems. Consider, part of speech tagging, for example. A part of speech tagger takes an input sequences of words such as *The table is ready.* and outputs a sequence of parts of speech such as: *Article Noun Verb Adjective.* The problem is non-trivial because it is well-known that part of speech depends on context. The word "table," for example, is usually a noun, but it can also be a verb in some contexts such as: *The chairman will table the motion.*

The tagging problem can be viewed as a translation problem, not unlike machine translation. Imagine that we have a sequence of parts of speech P that go into the channel and produce a sequence of words W. Our job is to try to determine the hidden parts of speech P given the observed words W.

$$P \rightarrow \textit{Noisy Channel} \rightarrow W$$

As before, in principle, one can hypothesize all possible inputs to the channel and score them by:

$$\underset{P}{ARGMAX}\, Pr(P)\, Pr(W|P)$$

Again, the parameters in this model are generally estimated by computing various statistics over large bodies of text. Both Church (1988) and DeRose (1988) have used the Tagged Brown Corpus (Francis and Kucera, 1982) for this purpose, which is particularly convenient because it comes with parts of speech that were check by hand. deMarcken (1990) used the Tagged Lancaster/ Oslo-Bergen Corpus (LOB) which also comes with parts of speech. Others such as Jelinek (1985) have used the Baum-Welch Algorithm (Baum, 1972) to estimate the parameters from raw untagged text.

I have always felt that hand-tagged text produces more reliable estimates, and recently Merialdo (1990) performed an experiment which seems to back-up my suspicion. He estimated the parameters using some hand-tagged data and then ran the re-estimation procedure and compared performance before and after re-estimation. One might have thought that re-estimation ought to improve performance, but he found just the opposite. He concludes that one should use as much tagged text as possible to estimate the parameters, and one should resort to re-estimation only when it is not possible to find a sufficient amount of tagged training material.

There are, of course, many other "translation" applications that are very analogous to machine translation and part of speech tagging where one wants to transduce one tape of symbols into another. In speech recognition, for example, it is common to use these noisy-channel methods to translate a sequence of acoustic labels (e.g., the output of a filter bank) into a sequence of phonetic labels (e.g., consonants and vowels).

12. Conclusions

Quite a number of applications have been mentioned in just a few pages: spelling correction, speech recognition, optical character recognition, text compression, machine translation and part of speech tagging. Of course, there are many other applications that should have been discussed, especially information retrieval (Salton, 1989) and author identification (Mosteller and Wallace, 1964), but there just wasn't enough space to say everything.

All of this work points very strongly to the fact that 1950-style empiricism is back in fashion. I have been asked to explain why, and I'm not sure that I have a good answer. Of course, it is possible that the current interest in empiricism is just a fad that will soon fade away. But, I would like to believe that there are good reasons for the revival. One can point to huge advances in computational power since the 1950s. But, even more importantly, the electronic culture has now permeated the publishing sector to such an extent that it is no longer difficult to find hundreds of millions of words of text in electronic form. And there is promise of billions of words in the very near future. The availability of data on such a massive scale has made it possible to carry out experiments that just weren't possible back in the 1950s. Indeed, many of the experiments discussed in this paper would not have been possible without the availability of very large corpora.

References

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Lafferty, J., Mercer, R., Rossin, P. (1990a), "A Statistical Approach to Machine Translation," *Computational Linguistics.*

Brown, P., Della Pietra, V., deSouza, P., Lai J., Mercer, R. (1990b) "Class-based N-gram Models of Natural Language," unpublished ms., IBM.

Brown, P., Della Pietra, S., Della Pietra, V., Lai J., Mercer, R. (1991) "An Estimate of an Upper

Bound for the Entropy of English," submitted to *Computational Linguistics*.

Chomsky, N. (1957) *Syntactic Structures*, The Hague: Mouton & Co.

Church, K. (1988) "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," Second Conference on Applied Natural Language Processing, Association for Computational Linguistics, Austin, Texas.

Church, K., and Hanks, P. (1990) 'Word Association Norms, Mutual Information, and Lexicography,' *Computational Linguistics*, 16:1.

Church, K., and Gale, W. (1990) "Poor Estimates of Context are Worse than None," Darpa Workshop, Hidden Valley, PA.

Church, K., and Gale, W. (1991a) "Probability Scoring for Spelling Correction," *Statistics and Computing*.

Church, K., and Gale, W. (1991b) "A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams," *Computer Speech and Language*, vol. 5, pp. 19-54.

DeRose, S. (1988) "Grammatical Category Disambiguation by Statistical Optimization," *Computational Linguistics*, Vol. 14, No. 1.

deMarcken, C. (1990) "Parsing the LOB Corpus," Association for Computational Linguistics.

Firth, J. (1957) "A Synopsis of Linguistic Theory 1930-1955" in *Studies in Linguistic Analysis*, Philological Society, Oxford; reprinted in Palmer, F. (ed. 1968) *Selected Papers of J. R. Firth*, Longman, Harlow.

Francis, W., and Kucera, H. (1982) *Frequency Analysis of English Usage*, Houghton Mifflin Company, Boston.

Good, I., (1953), "The population frequencies of species and the estimation of population parameters," *Biometrika*, v. 40, pp. 237-264.

Jeffreys, H., (1948) *Theory of Probability*, second edition, section 3.23, Oxford: Clarendon Press.

Jelinek, F. (1985) "Self-organized Language Modeling for Speech Recognition," IBM Report, also available in, Waibel, A. and Lee, K. (eds.) (1990) *Readings in Speech Recognition*, Morgan Kaufmann Publishers, San Mateo, California.

Jelinek, F., and Mercer, R. (1985) "Probability distribution estimation from sparse data," *IBM Technical Disclosure Bulletin*, v. 28, pp. 2591-2594.

Katz, S. M., (1987), "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. ASSP-35, pp. 400-401.

Kernighan, M., Church, K., Gale, W. (1990) "A Spelling Correction Program Based on a Noisy Channel Model," Coling, Helsinki, Finland (proceedings are available from the Association for Computational Linguistics).

Klavans, J., and E. Tzoukermann (1990) "The BICORD System," Coling, Helsinki, Finland (proceedings are available from the Association for Computational Linguistics).

Mays M., F. Damerau and R. Mercer (1990) "Context Based Spelling Correction," IBM internal memo, RC 15803 (#730266).

Merialdo, B. (1990) "Tagging Text with a Probabilistic Model," in *Proceedings of the IBM Natural Language ITL*, Paris, France, pp. 161-172.

Mosteller, F. and Wallace, D. (1964) *Inference & Disputed Authorship: The Federalist*, Addison-Wesley, Reading, Massachusetts.

Salton, G. (1989) *Automatic Text Processing*, Addison-Wesley Publishing Co.

Shannon, C. (1948) "The Mathematical Theory of Communication," *Bell System Technical Journal*.

Shannon, C. (1951) "Prediction and Entropy of Printed English," *Bell Systems Technical Journal*, vol. 30, pp. 50-64.

Sinclair, J., Hanks, P., Fox, G., Moon, R., Stock, P. (eds), (1987a), *Collins Cobuild English Language Dictionary*, Collins, London and Glasgow.

Sinclair, J., (1987b) (ed.), "Looking Up: an account of the COBUILD Project in lexical computing," Collins, London and Glasgow.

Welch, T. (1984) "A Technique for High Performance Data Compression," *IEEE Computer*, vol. 17, no. 6, pp. 8-19.

Weaver, W. (1949) "Translation," reproduced in Locke, W. and Booth, A., (1955) (eds.), *Machine Translation of Languages*, MIT Press, Cambridge, Mass.

An Application of Neural Networks to Group Technology

THOMAS P. CAUDELL, SCOTT D. G. SMITH, AND STANLEY TAZUMA
Research and Technology
Boeing Computer Services
P.O. Box 24346
MS 7L-22
Seattle, WA. 98124-0346

Email: tpc@espresso.boeing.com

Abstract

Adaptive resonance theory (ART) neural networks are being developed for application to the industrial engineering problem of group technology -- the reuse of engineering designs. Two and three dimensional representations of engineering designs are input to ART-1 neural networks to produce groups or families of similar parts. These representations, in their basic form, amount to bit maps of the design, and can become very large when the design is represented in high resolution. We describe a "neural database" system under development. This system demonstrates the feasibility of training an ART-1 network to first cluster designs into families, and then to recall the family when presented a similar design. This application is of large practical value to industry, making it possible to avoid duplication of design efforts.

Introduction

Money and time can be saved by manufacturing companies when engineering designs are reused. This is particularly true in companies producing large systems, such as aircraft, that must be customized to varying layouts. Often the same design is inadvertently redesigned at great expense. This can happen frequently in large systems which involve teams of designers. A new designer will have no knowledge of a previous designer's work unless the technology exists to retrieve and compare designs. In industrial engineering, the study and implementation of such retrieval systems is referred to as group technology.

Several basic requirements must be met for the practical implementation of group technology. First, the designs must exist in, or be convertible to, an electronic description. Second, an appropriate criterion must be designed to determine similarity of designs. Third, the search algorithm must exceed a threshold of performance on the host computer to provide timely responses for the user. Fourth, a retrieval system should output the best few matches for consideration by the human designer. Fifth and final, the database must be easily maintainable and updateable. Few traditional database technologies provide all of these, particularly a criterion for measuring the similarity of geometrical shapes.

In the following, we will address the general application of neural networks to the group technology problem, where the designs are derived from a CAD system. Later in the paper, we will discuss the results of a specific neural database architecture that finds similar marker (ie. decals) designs. Markers are found in the passenger compartments and service bays of commercial airliners, and indicate locations of services, warnings, and restrictions to people who move and work in and around the aircraft. In this specific system, the data is not derived from a CAD system, but is acquired from paper drawings of the markers with the help of a PC based optical scanner, and is transferred to the network in raster format.

In the next section, we describe how a specific artificial neural network can meet all of the requirements of a group technology implementation. We will assume that there exists an electronic description of the design information. First we will introduce the ART-1 algorithm. We will then discuss the process of information translation into the binary representations needed for input to the network. A modification of the simulation is mentioned that makes use of data compression techniques. Finally, the markers retrieval system will be described.

ART-1 Algorithm

The adaptive resonance theory (ART) neural network model was developed by Carpenter and Grossberg[1]. The version of this model that processes binary input patterns is referred to as ART-1. The ART-1 neural network model is canonically represented by a coupled set of ordinary nonlinear differential equations[1]. If appropriate assumptions are made about the relationship between the learning rates and the dynamical time constants, this system of equations can be replaced by a procedural algorithm[2]. This "fast learning" mode of learning requires that the learning process stabilize each time before the next input pattern is presented. The impact of this assumption on both hardware and

software implementation is large: the computational steps of the algorithm can now be directly mapped onto an algorithmic processor. For this model, there is no need to become embroiled in the implementation issues of dynamical systems.

The basic functionality of this algorithm is to autonomously place input patterns into clusters or families. These patterns are represented as binary vectors. Clusters are formed and modified during the training process, often referred to as "self-organizing" learning. The number of clusters is not preset at the beginning, but is determined by the underlying structure of input patterns used during training and by a small set of network parameters. After training, the network is used as a "neural database", being queried by new input patterns to find the closest family. Again, the input patterns must be represented as binary vectors.

A characteristic of this self-organizing neural network is the formation of memory templates or archetypes during the repeated exposure of the network to the training set. A template isolates a conjunctive generalization[3] of the attributes representing the member patterns in that cluster. If the input pattern, denoted **I**, is found to be a member of an existing cluster after a search of neural memories, then this pattern is added to the membership list for that cluster, and the template associated with this cluster is updated to include the features of the new pattern. The updated template is a conjunction, or an "and", of the matching template and the newly added member input vector.

On the other hand, if **I** is new to the system's memories, then a new cluster is formed with **I** being the first member. In this case, the new template representing the new cluster becomes **I**. (That is, the archetype for a group with one member is the member itself). This process proceeds automatically with no outside supervision, finding order and structure in the stream of input patterns. For the learning process to stabilize, the training set of input patterns is repetitively presented to the network. In summary: when a new input vector is presented, it is then either placed into one of the existing clusters, or classified as a novel pattern and added to a new cluster.

During the search of the memory templates, the dot product of each memory template with the input vector is computed, as are the vector norms of each template and the input vector. That is,

$$|\mathbf{I}|$$

$$\{\ |\mathbf{T}_k|,\ 1 \le k \le n_c\ \}$$

$$\{\ (\mathbf{I} \bullet \mathbf{T}_k),\ 1 \le k \le n_c\ \}$$

where **I** is the current input vector, $\mathbf{T}_k$ is the k^{th} memory template, n_c is the current number of groupings, $\bullet$ is the dot product, and $|\ .\ |$ is the L_1 norm. During learning, the conjunction of the template and the input vector must be calculated as well. That is,

$$\mathbf{T}_k \leftarrow (\mathbf{I} \cap \mathbf{T}_k),$$

where $\cap$ is bit-wise "and". These calculations constitute a major portion of the processing load of the ART-1 algorithm.

The Neural Network Approach

Healy and Caudell have further developed the understanding of the logical functionality of the ART-1 network and have developed a methodology for the design of macrocircuits of ART-1 network modules[4]. Through the study of these logical architectures, we have applied ART-1 to the group technology problem. In this application, the network is trained on design representations derived directly from descriptions generated by such computer aided design (CAD) packages such as CATIA and CAD-KEY. Two, three, and higher dimensional descriptions are being used to represent features of designs.

The CAD system usually stores a "constructive description" of the part. That is, a list of instructions that tell a graphics rendering program how to draw a diagram of the part. The diagram tells the design engineer how this part fits into the overall system, the manufacturing engineering how to design the manufacturing process for the part, and the field service engineer how to maintain the part in the system. From this constructive description, a transformed representation must be produced by a preprocessing system to become the input for the neural network. The description of the design may come in other forms, including raster scanned images as mentioned above. In this later case, no preprocessing is required.

For a 2D designs, such as a sheet metal floor stiffener in an aircraft, the simplest transformed representation is a binary pixel map or silhouette; ones where there is solid material and zeros where there is none, defined over a predefined 2D graphical view port. This is shown in Figure 1a. The view port is a window on 2D space. The binary pixel map is strung-out or rasterized into a binary vector by concatenating rows of pixels from the view port. This vector is subsequently fed to the ART-1 neural network simulator for clustering into families.

Other forms of information may be represented as binary patterns. For example, Figure 1b illustrates how the position of fastener holes can be represented in a view port with the same dimensions of the silhouette, but with ones in the neighborhood of a hole, and zeros otherwise. The locations and degree of metal bends can be represented in a three dimensional "Hough Space", where the first two axis code the slope and intercept of the bend line, while the third axis codes the bend angle.

In this case, each bend line would be represented as a single point in a 3D space. If the angle of the bend is not important, then a bend line could be represented directly in a viewport as with the silhouette. This is shown in Figure 1c.

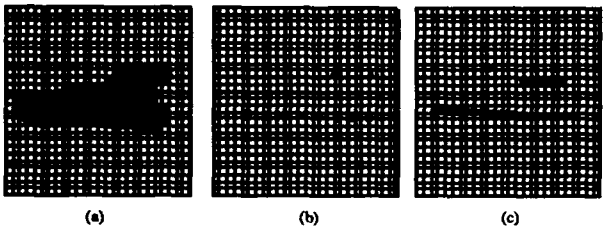


Figure 1. Three representations of features of a design. (a) is the silhouette of the part, (b) is the location of fastener holes, and (c) is the location of bend lines. Each of these are converted into linear binary vectors for input to the neural network.

The limitation of this type of sparce representation is in the explosion of the length of the binary input vector. The resolution of the pixelization determines the overall length of the binary input vector. The resolution also determines the accuracy of the object representation, and if too coarse it will strongly affect the way the network groups the designs. Even though the bits in the binary vector can be "packed" into 32 bit integers for storage and manipulation, when many clusters are formed, the total size of the vectors will tax the limits of small engineering workstations.

In our normal simulation of the ART-1 algorithm, the vectors and templates are in binary form before the dot products, norms, and conjunctions are calculated. A practical group technology parts retrieval system might be expected to require many ART-1 modules running with many hundreds of memory templates each. A compounding fact is that the range of engineering workstations on which the system might possibly be deployed include relatively low-end PCs. The following section briefly introduces a modification to the ART-1 algorithm that allows direct operation on data compressed input vectors and memory templates.

A Compressed ART-1 Algorithm

There are significant advantages to applying data compression techniques to the binary representations used in this ART-1 system. First of all, there is no random "noise" in designs, making accurate compression possible. Second, a bit map of a design will quite frequently have long strings of 1's and 0's as the material of the part is transited, producing potentially large data compression ratios. Finally, the neural network simulation will have fewer actual numbers to process per part, reducing the execution times.

In this work, standard run-length encoding is used. For an example, see Figure 2. Although other more sophisticated techniques are available, it is the low conversion overhead and basic simplicity of run-length encoding that makes it ideal for this application. A run-length algorithm returns a list of integers that represents the lengths of runs of consecutive 1's and/or 0's in the binary vector. Efficient linear algorithms exist to compress data into this format. With the assumption that the starting value of the list is known, the fact that the 1's and 0's alternate allows this list to be stored without the actual values of the runs.

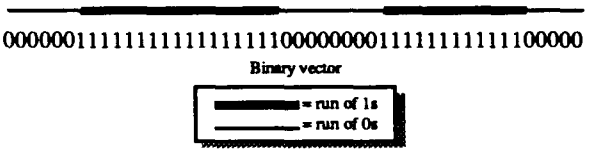


Figure 2. An example of a short binary vector. The run-length code C for this string is {6,17,8,12,5} with byte compression ratio of 8/5. This ratio assumes that the uncompressed vector is stored in compact form in 8-bit bytes, and that the maximum length of a single run is 256. The bar above the vector symbolically indicates the location of strings of 0's and 1's, and is used to explain the compressed algorithms later in the text.

The ART-1 simulation used for this research was modified to include compressed versions of the vector operations described above. The input patterns are compressed before presentation to the network. The memory templates are created and updated directly in compressed form. Data compression ratios and execution times were measured for both compressed and uncompressed versions of the simulations. In these experiments, compression ratios of up to 20 were found using 2-D CAD designs. In addition, speedups of the ART-1 algorithm of upwards of 100 were measured for 3-D CAD designs. These improvements are important to the developers and end-users of these neural retrieval systems because it makes deployment of practical applications on existing engineering workstations possible.

Neural Database Architecture

For the group technology applications considered so far in our research group, a generic system architecture is emerging. This can be seen in Figure 3. The basic components are 1) CAD System Interface, 2) Parser, 3) Representation Generator, 4) Neural Network macrocircuits, and 5) User Interface.

In a group technology system the lists of parts which form each cluster are maintained during training. When the user queries the system with a new design, that design is presented to the network and the list of parts which previously grouped in the same cluster are returned.

The functionality of the Parser is to extract the salient information from the CAD System Interface. Typically, this interface is an ASCII data file containing the constructive description of the part. It may also be a raster file of an image. The extracted information might be a list of lines and arcs defining the border of the part, the location of fastener holes, or a bit map of the design. Unfortunately, the structure of the data files usually depends on the style and consistency of the user of the CAD program, making multiple searches of the data necessary. Sometimes information on a substructure of the part will be distributed in many locations in the CAD file. The Parser is the only component specific to the brand of CAD program being used, and must be redesigned for each new system.

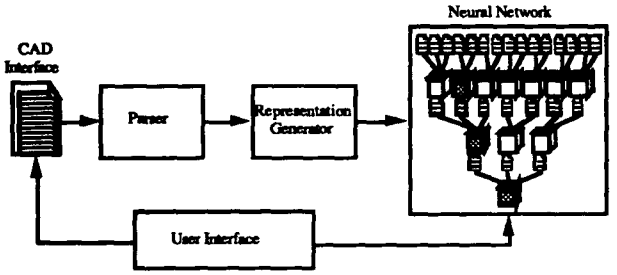


Figure 3. A schematic of the components of a neural database system for group technology. The User Interface provides control of the level of abstraction of recall in the network.

The Representation Generator converts and compresses the information extracted by the Parser into a form usable by the neural network. This includes operations such as the generation of the 2D viewports, generating silhouettes by filling in boundaries, computing the location of points in Hough spaces, and the compression of each representation into run-length codes. This component is independent of the type of CAD program used in design generation, but will vary depending on the types of representations required to capture the significant features that best discriminates the design families.

The structure of the ART-1 Macrocircuits component is also dependent on the representations, and will vary according to requirements of the database users. A macrocircuit is a collection of neural network modules, connected together in a larger and more functional network. These are necessary if a network is to give the user a range of query options. For example, the user may choose to query the database for designs that have the same general size, represented by a bounding rectangle or box. After limiting the choices of families by this step, the user may next want to discriminate according to the the specific shape of the object. The structure of the macrocircuit strongly effects the range of functionality provided by the neural database. (See Figure 6 for a diagram of the macrocircuit used in the demonstrations system discussed in the following section.)

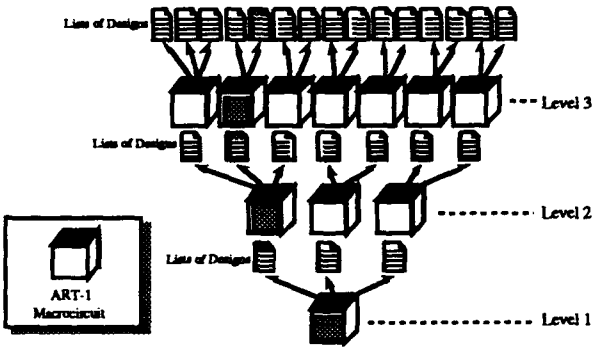


Figure 4. An example of a ART Tree database structure. Each cube represents a macrocircuit of ART-1 neural networks to provide "don't care" option to query.

Another user requirement might be the ability to vary the degree of the family discriminators, allowing on-line specification of the closeness of a match in the search for similarity. This can be implemented with a hierarchical abstraction tree of macrocircuit modules, as shown in Figure 4. Each module in the tree is trained separately with the input patterns associated only with that branch cluster, and each module receives the complete set of representations. The modules at the top of the tree have the greatest discrimination, while the one at the bottom has the least. When a query occurs, the lowest module places the design into one of its families or clusters. Families at this level represent the most general abstraction of the possible set of designs stored in the system. When a winning cluster is selected at the first level, the module up the branch of the tree associated with this group is activated. This module then places the design into one of it's clusters, and the process repeats. The user selects the level of abstraction at retrieval time according to the current requirements.

An Application to Marker Retrieval

As an illustration of the types of systems currently under development, more detail will be given on the marker design retrieval system. Figure 5 gives two markers that are similar in size and textual content, but differ in the graphical information. It is possible that only one of these need be saved. Often new markers are needlessly designed because no retrieval system exists to aid the designer. The markers designs are produced and stored on sheets of paper bound in volumes, complicating electronic access.

For this demonstration, approximately 50 markers were digitized on a Macintosh optical scanner to capture the

graphical shape. These images were then converted to raster file bit maps for input to the neural network. In addition, the cut-out die size and textual content of the marker were recorded with the image. Figure 6 gives the details of how sets of ART-1 modules are connected to implement the database system.

The detailed structure of this macrocircuit evolves during the learning process, where a training set of marker designs are repetitively presented to the network. The die size and textual information are used to form families. When a new size/text family forms, an ART-1 module is created to cluster the graphics associated with this family into subfamilies of similar shapes. In Figure 6, the shape representation is considered last by the highest ART-1 module.

One advantage of this sort of hierarchical structure is that it could be easily incorporated into a traditional database system. The categorization that occurs before presenting the graphical images to the neural network could be performed by querying an existing database. Thus, any attributes of the markers that have been entered into a database could be used prior to graphical grouping.

This demonstration system mentioned above is implemented in the C language on Sun SPARC workstations. Training for this small system takes less than ten minutes, and retrieval time for a new design is less than a second. The ART Tree structure has not been implemented for this application. Figure 7 shows a screen-dump of a trained network.

A neural network grouping system for airplane markers could be used in a number of ways. The existing markers could be grouped and then the groups examined by a human to locate and purge duplicate markers. This would save money in maintenance. Also, such a system could be used for group technology to return the closest existing markers to a new one being designed. This would help avoid the future proliferation of duplicate markers. Finally, additions to traditional databases could be constructed which would graphically group the markers returned to the user in response to a query.

Figure 5. Two markers that are the same size and have the same message, but contain different graphical information.

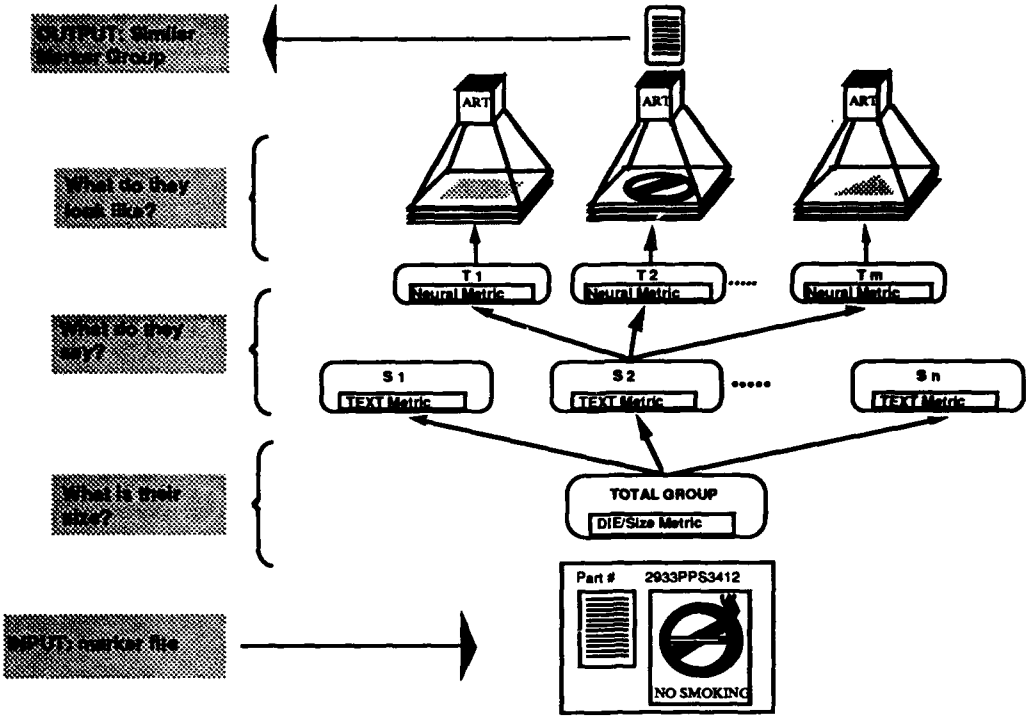


Figure 6. The macrocircuit of ART-1 modules that implements the markers design retrieval system.

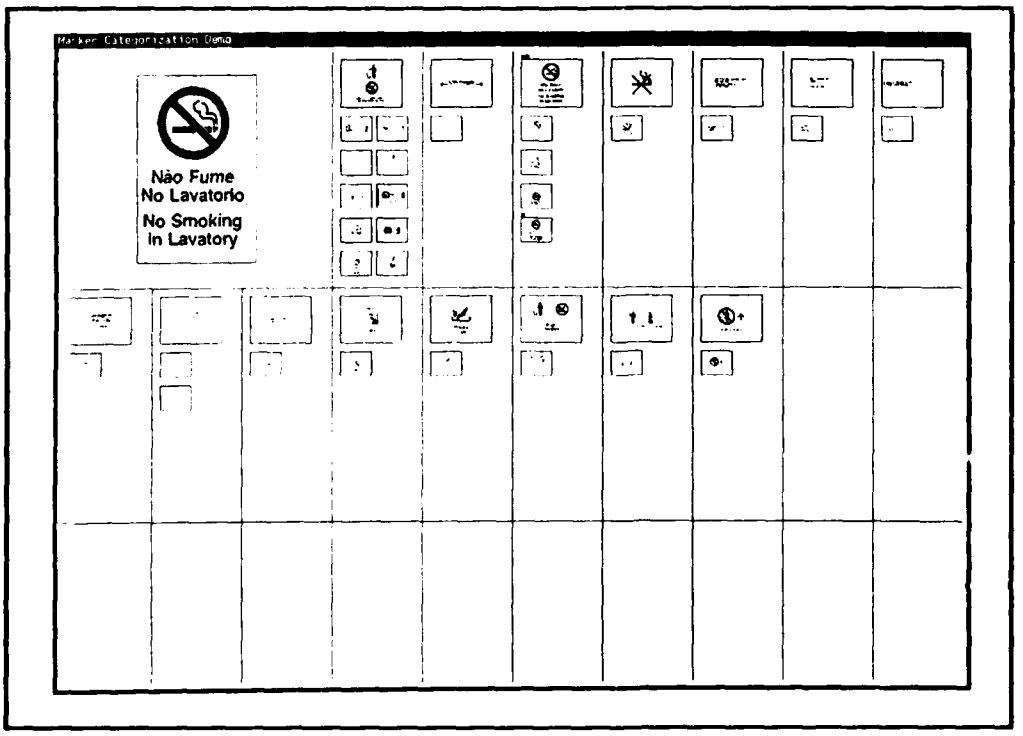


Figure 7. A screen dump from a Sun SPARC1 simulation of the sheet metal floor stiffener retrieval system. The three representations of shape, holes, and bends appear in viewports across the top of the image. The set of silhouettes in the upper middle of the figure are the memory templates for the shape ART-1 module. The lower set of rectangular windows show the results of the holes and bends modules for each shape cluster.

Conclusions

Artificial neural networks have been applied to design retrieval. ART-1 networks are used to adaptively group together similar engineering or graphical designs. The information used to group is coded into a binary representations which, in their basic form, amounts to bit maps of design descriptors. We have used this technology to build neural databases for the retrieval of two and three dimensional engineering designs. We have discussed in detail a feasibility level system that learns to group airliner markers into families, and then to recall the family when presented a similar marker. The input to these networks may be generated directly from CAD designs of the parts or other sources of object features.

An addition to the algorithmic form of ART-1 was introduced that allows it to operate directly on run-length encoded vectors, and to generate compressed memory templates. When compared to the regular uncompressed algorithm on real engineering designs, the performance of this compressed algorithm demonstrated a significant savings in storage of the input vector and the memory templates. A surprising result was the size of the speed up in execution of the simulation on larger input vectors. Issues of object scale, orientation, and reflection have not been discussed here, although they have been dealt with in the working systems. The code for a system that groups aircraft floor stiffener sheet metal parts has been transferred to a PC based engineering workstation for beta testing. The application of neural networks to group technology is of large practical value to industry, by making it possible to avoid duplication of design efforts and save many down stream costs.

Acknowledgements

The authors would like to acknowledge the assistance of Vicky Lane, Craig Johnson, Mike Healy, Richard Escobedo, Mark Singer, Kathryn Chalfan, Dennis Campbell, and John Gravendyke in this project.

References

1. Carpenter, G.A. and Grossberg, S., "A massively parallel architecture for a self-organizing neural pattern recognition machine", Computer Vision, Graphics, and Image Processing, #37, pp 54-115, Academic Press, 1987.

2. Moore, B., "ART-1 and Pattern Clustering", Proceedings of the 1988 Connectionist Summer School at Carnegie Mellon University, Touretzky and Hinton Ed., Morgan Kaufman, 1989.

3. Healy, M.J., "An investigation of knowledge representations in a neural network", in the proceedings of Northcon/88, Vol II, Western Periodicals Co., Oct. 1988.

4. Healy, M.J. and Caudell, T.P., "On the Semantics of Pattern Recognition Neural Networks", in the proceedings of Northcon/88, Vol II, Western Periodicals Co., Oct. 1990.

Advances in Probabilistic Reasoning*

Dan Geiger
Northrop Research and Technology Center
One Research Park
Palos Verdes, CA 90274

David Heckerman
Departments of Computer Science and Pathology
University of Southern California
HMR 204, 2025 Zonal Ave, LA, CA 94305

Abstract

This paper discuses multiple Bayesian networks representation paradigms for encoding asymmetric independence assertions. We offer three contributions: (1) an inference mechanism that makes explicit use of asymmetric independence to speed up computations, (2) a simplified definition of similarity networks and extensions of their theory, and (3) a generalized representation scheme that encodes more types of asymmetric independence assertions than do similarity networks.

Introduction

Traditional probabilistic approaches to diagnosis, classification, and pattern recognition face a critical choice: either specify precise relationships between all interacting variables or make uniform independence assumptions throughout. The first choice is computationally infeasible except in very small domains, while the second, which is rarely justified, often yields inadequate conclusions.

Bayesian networks offer a compromise between the two extremes by encoding independence when possible and dependence when necessary. They allow a wide spectrum of independence assertions to be considered by the model builder so that a practical balance can be established between computational needs and adequacy of conclusions.

Although Bayesian networks considerably extend traditional approaches, they are still not expressive enough to encode every piece of information that might reduce computations. The most obvious omissions are *asymmetric independence* assertions stating that variables are independent for some but not necessarily for all of their values. Such asymmetric assertions cannot be represented naturally in a Bayesian network. Several researchers observed this limitation, however, until recently no effort was made to remove it.

Similarity network paradigm is the first major effort towards the r··resentation of asymmetric independence [Heckerman, 1990]. Contingent influence diagrams is an alternative approach [Fung and Shachter, 1991]. Both schemes employ asymmetric independence to ease the elicitation and improve the quality of probabilistic models.

This article offers three contributions: (1) an inference mechanism that makes explicit use of asymmetric independence to speed up computations, (2) a simplified definition of similarity networks and extensions of their theory, and (3) a generalized representation scheme that encodes more types of asymmetric independence assertions than do similarity networks.

These contributions address problems of knowledge representation, inference, and knowledge acquisition. In particular, Section 2 describes *Bayesian multinets* and how to use them for inference, Section 3 describes knowledge acquisition using *similarity networks* and how to convert them to Bayesian multinets, Section 4 extends these representation schemes to the case where hypotheses are not mutually exclusive and section 5 summarizes the results. We assume the reader is familiar with the definition and usage of Bayesian networks. For details consult [Pearl, 1988].

Representation and Inference

Bayesian Multinets

The following example demonstrates the problem of representing asymmetric independence by Bayesian networks:

> A guard of a secured building expects three types of persons to approach the building's entrance: workers in the building, approved visitors, and spies. As a person approaches the building, the guard notes its gender and whether or not the person wears a badge. Spies are mostly men. Spies always wear badges in order to fool the guard. Visitors don't wear badges because they don't have one. Female-workers tend to wear badges more often than do male-workers. The task of the guard is to identify *the type of person* approaching the building.

A Bayesian network that represents this story is shown in Figure 1. Variable h in the figure represents the cor-

*This paper is reprinted from the proceedings of the 7th Uncertainty in Artificial Intelligence conference, Los Angeles, California.

rect identification. It has three values w, v, and s respectively denoting worker, visitor, and spy. Variables g and b are binary variables representing, respectively, the person's gender and whether or not the person wears a badge. The links from h to g and from h to b reflect the fact that both gender and badge-wearing are clues for correct identification, and the link from g to b encodes the relationship between gender and badge-wearing.

Unfortunately, the topology of this network hides the fact that, independent of gender, spies always wear badges and visitors never do. The network does not show that gender and badge-wearing are conditionally independent given the person is a spy or a visitor. A link between g and b is drawn merely because gender and badge-wearing are related variables when the person is a worker.

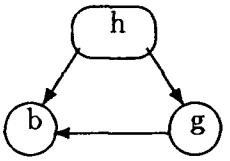


Figure 1: A Bayesian network for the secured-building example.

We can more adequately represent this story using two Bayesian networks shown in Figure 2. The first network represents the cases where the person approaching the entrance is either a spy or a visitor. In these cases, badge-wearing depends merely on the type of person approaching, not on its gender. Consequently, nodes b and g are shown to be conditionally independent (node h blocks the path between them). The links from h to b and from h to g in this network reflect the fact that badges and gender are relevant clues for distinguishing between spies and visitors. The second network represents the hypothesis that the person is a worker, in which case gender and badge-wearing are related as shown.

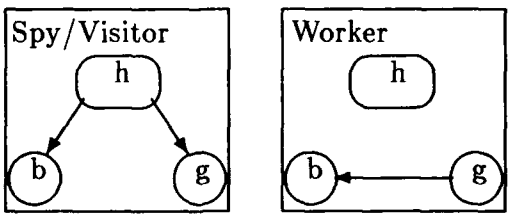


Figure 2: A Bayesian multinet representation of the secured-building story.

Figure 2 is a better representation than Figure 1 because it shows the dependence of badge-wearing on gender only in context in which such a relationship exists, namely, for workers. Moreover, the former representation requires 11 parameters while the representation of Figure 2 requires only 9. This gain, due to asymmetric independence, could be substantially larger for real-sized problems because the number of parameters needed grows exponentially in the number of variables, whereas the overhead of representing multiple networks grows only linearly.

We call the representation scheme of figure 2, a *Bayesian multinet.*

Definition Let $\{u_1 \ldots u_n\}$ be a finite set of variables each having a finite set of values, P be a probability distribution having the Cartesian product of these sets of values as its sample space, and h be a distinguished variable among the u_i's that represents a mutually-exclusive and exhaustive set of hypotheses. Let $A_1, ..., A_k$ be a partition of the values of h. A directed acyclic graph D_i is called a *local network* of P (associated with A_i) if it is a Bayesian network of P given that one of the hypotheses in A_i holds, i.e., D_i is a Bayesian network of $P(u_1 \ldots u_n | A_i)$. The set of k local networks is called a *Bayesian multinet* of P.[1]

In the secured-building example of Figure 2, $\{\{spy, visitor\}, \{worker\}\}$ is a partition of the values of the hypothesis node h, one local network is a Bayesian network of $P(h, b, g | \, worker)$ and the other local network is a Bayesian network of $P(h, b, g |$ {spy, visitor}). [2]

The fundamental idea of multinets is that of *conditioning*; each local network represents a distinct situation conditioned that hypotheses are restricted to a specified subset. Savings in computations and space occur because, as a result of conditioning, asymmetric independence assertions are encoded in the topology of the local networks. In the example above, conditional independence between gender and badge-wearing is encoded as a result of conditioning on h.

Notably, conditioning may also destroy independence relationships rather then create them [Pearl, 1988]. However, if the distinguished variable is a root node (i.e., a node with no incoming links), conditioning on its values never decreases and often increases the number of independence relationships, resulting in a more expressive graphical representation. Other situations are addressed below where the hypothesis variable is not a root node or where more than one node represents hypotheses.

Representational and Computational Advantages

The vanishing dependence between gender and badge-wearing is an example of an *hypothesis-specific* independence because it is manifest only when conditioning on

[1] A Bayesian multinet roughly corresponds to an *hypothesis-specific similarity network* as defined in Heckerman's dissertation (1990, page 76).

[2] The conditioning set $\{spy, visitor\}$ is a short hand notation for saying that h draws its values from this set, namely, either $h = spy$ or $h = visitor$.

specific hypotheses, that is, for spies and visitors, but not for workers. The following variation of the secured-building example demonstrates an additional type of asymmetric independence that can be represented by Bayesian multinets as well.

> The guard of the secured building now expects *four* types of persons to approach the building's entrance: executives, regular workers, approved visitors, and spies. The guard notes gender, badge-wearing, and whether or not the person arrives in a limousine (l). We assume that only executives arrive in limousines and that male and female executives wear badges just as do regular workers (*to* serve as role models).

This story is represented by the two local networks shown in Figure 3. One network represents a situation where either a spy or a visitor approaches the building, and the other network represents a situation where either a worker or an executive approaches the building. The link from h to l in the latter network reflects the fact that arriving in limousines is a relevant clue for distinguishing between workers and executives. The absence of this link in the former network reflects the fact that it is not relevant for distinguishing between spies and visitors.

The vanishing dependence between gender and the hypothesis variable h when h is restricted to a subset of hypotheses $\{worker, executive\}$ is an example of *subset independence*. Similarly, badge-wearing is independent of h when restricted to $\{worker, executive\}$, and arriving in limousines is independent of h when restricted to $\{spy, visitor\}$. [3]

Subset independence is a source of considerable computational savings. For example, in lymph-node pathology less than 20% of the potential morphological findings are relevant for distinguishing any given pair of disease hypotheses (among over 60 diseases) [Heckerman, 1990].

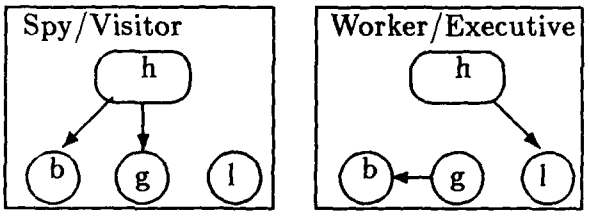


Figure 3: A Bayesian multinet representation of the augmented secured-building story.

Below we demonstrate these computational savings using the simple secured-building example; more savings are obtained in real domains such as lymph-node pathology.

[3] Heckerman coined the terms subset independence and hypothesis-specific independence in his dissertation.

Suppose the guard sees a male (g) wearing a badge (b) approaches the building and suppose the guard doesn't notice whether or not the person arrives in a limousine. A computation of the posterior probability of each possible identification (*executive, worker, visitor, spy*) based on the Bayesian network of Figure 1 simply yields the chaining rule:

$$P(h|\mathbf{g}, \mathbf{b}) = K \cdot P(h) \cdot P(\mathbf{g}|h) \cdot P(\mathbf{b}|\mathbf{g}, h). \quad (1)$$

where K is the normalizing constant.

Using the representation of Figure 3, however, the following more efficient computations are done instead:

$$P(spy|\mathbf{g}, \mathbf{b}) = K \cdot P(spy) \cdot P(\mathbf{g}|spy) \cdot P(\mathbf{b}|spy) \quad (2)$$

$$P(visitor|\mathbf{g}, \mathbf{b}) = K \cdot P(visitor) \cdot P(\mathbf{g}|visitor) \cdot P(\mathbf{b}|visitor) \quad (3)$$

$$P(worker|\mathbf{g}, \mathbf{b}) = K \cdot P(worker) \cdot P(\mathbf{g}|worker) \cdot P(\mathbf{b}|\mathbf{g}, worker) \quad (4)$$

$$P(\mathbf{g}, \mathbf{b}|executive) = P(\mathbf{g}, \mathbf{b}|worker). \quad (5)$$

Equations 2 and 3 take advantage of an hypothesis-specific independence assertion, namely, that g and b are conditionally independent given, respectively, that $h = spy$ and $h = visitor$. Equation 5 uses a subset independence assertion, namely, that b and g are independent of h restricted to $\{worker, executive\}$.

More generally, calculating the posterior probability of each hypothesis based on a set of observations $e_1, ..., e_m$ is done in two steps. First, for each hypothesis h_i, the probability $P(e_1, ..., e_m|h_i)$ is computed via standard algorithms such as Spiegelhalter and Lauritzen's (88) or Pearl's (88). Second, these results are combined via Bayes' rule:

$$P(h_i|e_1 ... e_m) = K \cdot p(h_i) P(e_1 ... e_k|h_i). \quad (6)$$

Notably, the computation of $P(e_1 \ldots e_k|h_i)$ in the first step uses the local networks as done in Eqs. (2) through (5) and does not use a single Bayesian network as done in Eq. (1). Consequently, when the values of h are properly partitioned, the extra independence relationships encoded in each local network could considerably reduce computations.

The parameters needed to perform the above computations consist, as we shall see next, of the prior of each hypothesis h_i and the parameters encoded in the local networks:

Theorem 1 *Let $\{u_1 \ldots u_n\}$ be a finite set of variables each having a finite set of values, P be a probability distribution having the Cartesian product of these sets of values as its sample space, h be a distinguished variable among the u_is, and M be a Bayesian multinet of P. Then, the posterior probability of every hypothesis given any value combination for the variables in $\{u_1 \ldots u_n\}$ can be computed from the prior probability of h's values and from the parameters encoded in M.*

According to Eq. 6 above, the only parameters needed for computing the posterior probability of each hypothesis h_i, aside of the priors, are $p(v_2 \ldots v_n|h_i)$ where $v_2 \ldots v_n$ are arbitrary values of $u_2 \ldots u_n$ (assuming without loss of generality that $h = u_1$). Let D_i denote a local network in M, A_i be the hypotheses associated with D_i, and h_i be an hypothesis in A_i. Clearly, $p(v_2 \ldots v_n|h_i)$ is equal to $p(v_2 \ldots v_n|h_i, A_i)$ because h_i logically implies the disjunction over all hypotheses in A_i. The latter probability is computable from the local network D_i by any standard algorithm (e.g., [Pearl, 1988]), thus, the former is also computable as needed. □

For example, $P(\mathrm{g}|worker, \{worker, executive\})$ is equal to the probability $P(\mathrm{g}|worker)$ because *worker* logically implies the disjunction *worker* $\vee$ *executive*. In fact, $P(\mathrm{g}|worker, \{worker, executive\})$ is also equal to $P(\mathrm{g}|\{worker, executive\})$ because g and *worker* are independent given $\{worker, executive\}$ as shown in Figure 3. In this example, the needed probability $P(\mathrm{g}|worker)$ is equal to the given one $P(\mathrm{g}|\{worker, executive\})$, however in general, the needed probabilities are computed via standard inference algorithms.

Overcoming some Limitations

The multinet approach described thus far is especially beneficial when the hypothesis variable can be modeled as a root node because, then, no dependencies are ever introduced by conditioning on the different hypotheses. However, the hypothesis node cannot always be modeled as a root node. For example, in the secured-building story, suppose there are two independent reports indicating possible spying, say, for military and economical reasons respectively. Such a priori factors for correct identification are modeled as parent nodes of h, called, say, *economics* and *military* having no link between them to show their mutual independence. The resulting network in this case is simply *economics* $\rightarrow h \leftarrow$ *military*.

However when h assumes the value *spy*, an induced link is introduced between its parents *economics* and *military*; one explanation for seeing a spy changes the plausibility of the other explanation, thus making the two variables economics and military be not independent conditioned on h = *spy*. Consequently, an induced link must be drawn between the *economics* and *military* nodes in the local network for spies vs. visitors to account for the above dependency. This link would not appear in the full Bayesian network because economics and military are marginally independent (they become dependent only when conditioning on h = *spy*). Such induced links are often hard to quantify and therefore, constructing a single local network is sometimes harder than constructing the full network, as is the case in the above example.

One approach to handle this situation is to first construct a Bayesian network that represents only a priori factors that influence the hypotheses, ignoring any evidential variables (such as gender, badge-wearing, and limousines). In our example, this network would be *economics* $\rightarrow h \leftarrow$ *military*. Then, use this network to revise the a priori probabilities of the different hypotheses. Finally, construct local networks ignoring a priori factors (as done in Figure 2) and use the resulting multinet with the revised priors of h to compute the posterior probability of h as determined by the evidential clues. This decomposition technique works best if a priori factors are independent of all clues conditioned on the different hypotheses. That is, in situations that can be modeled with Bayesian networks of the form shown in Figure 4 where all paths between a priori factors r_i's and evidential clues f_i's pass through h.

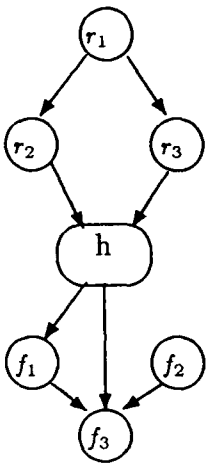


Figure 4: A Bayesian network where all paths between a priori factors r_i's and evidential clues f_i's pass through h.

When a network of this form cannot serve as a justifiable model, another approach can be used instead; compose a Bayesian multinet ignoring a priori factors, construct a Bayesian network from the local networks by taking the union of all their links (e.g., the union of all links in Figure 2 yields the Bayesian network of Figure 1). Finally, add a priori factors to the resulting network. This approach was proposed in [Heckerman, 1990].

The disadvantage of this method is that in the process of generating a Bayesian network from a multinet, one encodes asymmetric independence in the parameters rather than in the topology of the Bayesian network. Consequently, these asymmetric assertions are not available to standard inference algorithm to speed up their computations.

Nevertheless, this approach is still the best alternative for decomposing the construction of large Bayesian networks having topologies more complex than that of Figure 4. Such decomposition techniques are crucially

needed due to the overwhelming details of real-life problems. Additional issues of knowledge acquisition are discussed below.

Knowledge Acquisition/ Representation

Similarity Networks

Recall the guard that must distinguish between workers, executives, visitors and spies. In this story, some variables do not help distinguish between certain hypotheses. For example, gender and badges do not help distinguish between workers and executives, and limousines do not help distinguish between spies and visitors. In richer domains, large numbers of variables are often not relevant for distinguishing between certain hypotheses.

Unfortunately, the Bayesian multinet approach requires full specification of all variables in each local network even when they are not relevant to distinguish between the hypotheses associated with that local network. For example the relationship between b and g is encoded in the local network for spies vs. visitors although these variables do not help distinguish between this pair of hypotheses (Figure 3). Assessing such relationships, in contexts where they are not relevant, imposes insurmountable burden on the expert consulted as is demonstrated by the following quote [Heckerman, 1990]:

> "When the expert pathologist was asked questions of the form
>
> > Given any disease, does observing feature x change your belief that you will observe feature y ?
>
> the expert sometimes would reply
>
> > I've never thought about these two features at the same time before. Feature x is relevant to only one set of diseases, while feature y is only relevant to another set of diseases. These sets of diseases do not overlap, and I never confuse the first set of diseases with the second."

The solution is to simply include in each local network only those variables that are relevant for distinguishing between the hypothesis covered by that local network.

However, by doing so, valuable information for correct identification might be lost. For example, the relationships between badge-wearing and gender in Figure 3 would be lost. To compensate for such losses of information, additional local networks must be constructed.

For example, the secured-building can be represented with three local networks shown in Figure 5 rather than two as in Figure 3. One network is used to distinguish between spies and visitors, another between visitors and workers, and a third between workers and executives. In each local network we include only those variables relevant to distinguishing the hypotheses covered by that local network. In particular, the relationship between badge-wearing and gender is not included in the local network for workers vs. executives as in Figure 3. This relationship, however, is included in the local networks for visitors vs. workers because it helps distinguish between these two hypotheses. The reason for not loosing needed information is that the three local networks are based on a *connected cover* of hypotheses (rather than a partition).

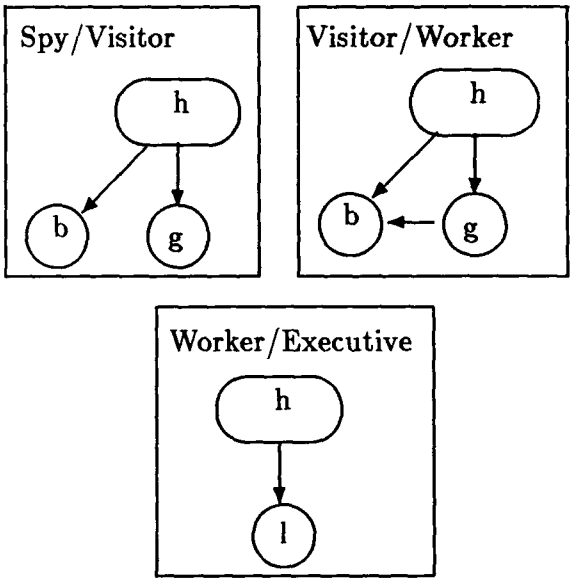


Figure 5: A similarity network representation of the secured-building story.

Definition A *cover* of a set A is a collection $\{A_1, ..., A_k\}$ of non-empty subsets of A whose union is A. Each cover is a hypergraph, called the *similarity hypergraph*, where the A_i's are edges and elements of A are nodes. A cover is *connected* if the similarity hypergraph is connected.

In Figure 5, {*spy, visitor*}, {*visitor, worker*}, {*worker, executive*} is a cover of the hypotheses set. This cover is connected because it is simply a four-nodes chain *spy—visitor—worker—executive* which, by definition, is a connected hypergraph. The set {{*spy, visitor*}, {*worker, executive*}} is also a cover but it is not connected. The set {{*worker, executive, visitor*}, {*visitor, spy*}} is an example of a connected cover that is a hypergraph which is not a graph.

Definition Let $U = \{u_1 \ldots u_n\}$ be a finite set of variables each having a finite set of values, P be a probability distribution having the cross product of these sets of values as its sample space, and h be a distinguished variable among the u_i's that represents a mutually-exclusive and exhaustive set of hypotheses. Let $A_1, ..., A_k$ be a connected cover of the values of h. A directed acyclic graph D_i is called a *comprehensive local network* of P (associated with A_i) if it is a Bayesian network of P assuming one of the hypotheses in A_i holds, i.e., D_i is a Bayesian

network of $P(u_1 \ldots u_n|A_i)$. The network obtained from D_i by removing nodes that are not relevant to distinguishing between hypotheses in A_i is called an *ordinary local network*. The set of k ordinary local networks is called an *(ordinary) similarity network* of P.

For example, the local networks of Figure 5 are ordinary, and together form an ordinary similarity network. Notably, hypotheses covered by each local network are often similar (e.g., spies and visitors), [4] a choice that maximizes the number of asymmetric independence relationships encoded.

Heckerman (1990) shows that under several assumptions, if a cover is connected, one can always remove from each local network variables that do not help distinguish between hypotheses covered by that local network and yet not loose the information necessary for representing the full joint distribution. These assumptions consist of 1) the hypothesis variable is a root node, 2) the cover is a graph and not a hypergraph, 3) the local networks are constrained by the same partial order, and 4) the distribution is strictly positive. Theses assumptions are relaxed below.

Theorem 2 *Let $\{u_1 \ldots u_n\}$ be a finite set of variables each having a finite set of values, P be a probability distribution having the Cartesian product of these sets of values as its sample space, h be a distinguished variable among the u_is, and S be a similarity network of P. Then, the posterior probability of every hypothesis given any value combination for the variables in $\{u_1 \ldots u_n\}$ can be computed from the parameters encoded in S provided $p(h_i) \neq 0$ for every value h_i of h.*

To prove the above theorem, it suffices to consider the case where h is a root node in all the local networks of S because, otherwise, *arc-reversal* transformations [Shachter 1986] can be applied until h becomes one.

Also note that since the similarity hypergraph is connected, it imposes $n-1$ independent equations among the following n: $p(h_i) = p(h_i|A_i) \cdot \sum_{h_j \in A_i} p(h_j)$, $i = 1 \ldots n$. In addition, $\sum_1^n p(h_i) = 1$. The values for $p(h_i)$ are the unique solution of these linear equations provided $p(h_i) \neq 0$ for $i = 1 \ldots n$.

Aside of the priors, the only remaining parameters needed for computing the posterior probability of each hypothesis h_i, are $p(v_2 \ldots v_n|h_i)$ where $v_2 \ldots v_n$ are arbitrary values of $u_2 \ldots u_n$ (assuming without loss of generality that $h = u_1$). Due to the chaining rule, $p(v_2 \ldots v_n|h_i)$ can be factored as follows:

$$p(v_2 \ldots v_n|h_i) = P(v_2|h_i) \cdot P(v_3|v_2\, h_i) \ldots p(v_n|v_1 \ldots v_{n-1}\, h_i).$$

Thus, it suffices to show that for each variable u_j, $p(v_j|v_2 \ldots v_{j-1}\, h_i)$ can be computed from the parameters encoded in S.

[4] Hence the name: similarity network.

Let D_i denote a local network in S, A_i be the hypotheses associated with D_i, and h_i be an hypothesis in A_i. There are two cases; either u_j is depicted in D_i or it is not. Let $A_i, A_{i+1} \ldots A_m$ be a path in the similarity hypergraph where A_m is the only edge on this path associated with a local network that depicts u_j as a node. If u_j is depicted in D_i, then the path consists of one edge A_i which is equal to A_m. If u_j is not depicted in any local network, then u_j does not alter the posterior probability of any hypothesis and is therefore omitted from the computations.

Let D_k be the local netowrk associated with A_k for $k = i+1 \ldots m$ and let $h_{i+1}, h_{i+2} \ldots h_m$ be a sequence of hypotheses such that $h_k \in A_{k-1} \cap A_k$. Due to the definition of similarity networks, since u_j is not depicted in D_k where $k < m$, the following equality must hold:

$$p(v_j|v_2 \ldots v_{j-1}\, h_{k-1}) = p(v_j|v_2 \ldots v_{j-1}\, h_k).$$

Since this equation holds for every k between $i+1$ and m, we obtain,

$$p(v_j|v_2 \ldots v_{j-1}\, h_i) = p(v_j|v_2 \ldots v_{j-1}\, h_m).$$

Moreover,

$$p(v_j|v_2 \ldots v_{j-1}\, h_m) = p(v_j|v'_1 \ldots v'_l\, h_m)$$

where $u'_1 \ldots u'_l$ are the variables depicted in D_m (a subset of $\{u_2 \ldots u_{j-1}\}$) because, due to the definition of similarity network, the variables deleted are conditionally independent of v_j, given the other variables; they are disconnected from all the other variables in D_m. [5]

Finally,

$$p(v_j|v'_1 \ldots v'_l\, h_m) = p(v_j|v'_1 \ldots v'_l\, h_m, A_m),$$

because h_m logically implies the disjunction over all hypotheses in A_m.

The latter probability is computable from the local network D_m by any standard algorithm (e.g., [Pearl, 1988]), thus, due the three equalities above, $p(v_j|v_2 \ldots v_{j-1}\, h_i)$ is also computable as needed. □

For example, to compute $P(g, b, l|spy)$ we use the following two equalities implied by Figure 5: From the first local network, $P(g, b, l|spy) = P(g|spy) \cdot P(b|spy) \cdot P(l|spy)$ and from the absence of l in the first and second local networks, $P(l|spy) = P(l|worker)$. Thus, $P(g, b, l|spy) = P(g|spy) \cdot P(b|spy) \cdot P(l|worker)$, where all the needed probabilities are encoded in the similarity network. In fact, the proof of Theorem 2 provides a general way of factoring any desired probability, thus, the full joint distribution $P(g, b, l, h)$ is encoded in the ordinary similarity network of Figure 5.

Similarity networks have another important advantage not mentioned so far: protecting the model builder from omitting relevant clues. For example, suppose workers

[5] Geiger and Heckerman (1990) discuss weaker definitions of being irrelevant other than being disconnected.

and executives often arrive with a smile to work (because the secured building is such a great place to be in) while spies and visitors arrive seriously. Such a clue, smile, is likely to be forgotten when constructing the local networks for spies vs. visitors and for visitors vs. executives because it does not help distinguish between these pairs of hypotheses. However, when constructing the similarity network of Figure 5, which includes a local network for distinguishing visitors from workers, smile is more likely to be recalled because the distinctions between visitors and workers are explicitly in focus.

Redundancy

Basing the construction of local networks on covers of hypotheses raises the problem of *redundancy*, namely, that some parameters are specified in more than one local network. For example, in Figure 5, the parameter $P(\text{g}|visitor)$ should, in principle, be specified both in the first and in the second local network. This problem is particularly crucial because local networks are actually constructed from expert's judgments rather than from a coherent probability distribution as implied by the definition of similarity networks.

One way to remove redundancy is to automatically-translate a similarity network as it is being constructed to a Bayesian multinet which is never redundant. For example, instead of storing Figure 5, we can actually store Figure 3 which contains no redundant information.

The translation is done by the following algorithm.

Conversion Algorithm

Input: A similarity network S of a probability distribution P.
Output: A Bayesian multinet of P.

1. For each ordinary local network L in S:
   - Add a node for each variable not represented in L.
   - For each added node x, set the parents of x in L to be the union of all parents of x in all other local networks where x originally appeared, excluding variables that were originally in L.
2. Remove enough local networks from S and enough hypotheses from the remaining local networks until a Bayesian multinet is obtained.

(A finer version of this algorithm is forthcoming).

Notably, the user of a similarity network need not know about the conversion to a Bayesian multinet which can be thought of as an internal representation. The user benefits from both the advantages of similarity network for knowledge acquisition, and from an inference algorithm (Section 2) that uses the Bayesian multinet produced by the conversion algorithm.

Generalized Similarity Networks

Previous sections assume all hypotheses are mutually exclusive and are, therefore, represented as values of a single hypothesis variable denoted h. Here this assumption is relaxed. We allow several variables to represent hypotheses, as needed by the following example:

> Consider the guard of Section 2 who has to distinguish between workers, visitors, and spies. A *pair* of people approach the building and the guard tries to classify them as they approach. Assume that only workers converse (c) and that workers often arrive with other workers (because they must car-pool to conserve energy).

A Bayesian network representing this situation is shown in Figure 6 where nodes h_1 and h_2 stand for the respective identity of the two persons. (The direction of the link between h_1 and h_2 is arbitrary.)

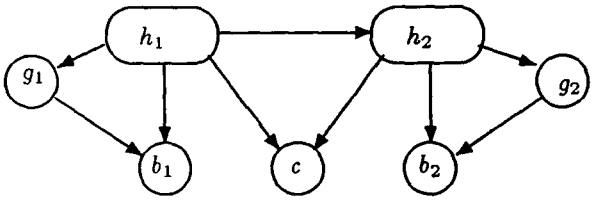


Figure 6: A Bayesian network with two hypothesis nodes h_1 and h_2.

Alternatively, we can represent this example using a *generalized similarity network*, or a *generalized Bayesian multinet*.

Definition Let $\{u_1 \ldots u_n\}$ be a finite set of variables each having a finite set of values, P be a probability distribution having the cross product of these sets of values as its sample space, and H be a subset of distinguished variables among the u_i's each representing a set of hypotheses. Denote the Cartesian product of the sets of values of the distinguished variables by *domain(H)*. Let $A_1, ..., A_k$ be a connected cover of *domain(H)*. A directed acyclic graph D_i is called a *comprehensive local network* of P if it is a Bayesian network of $P(u_1 \ldots u_n|A_i)$. The network obtained from D_i by removing nodes that are not relevant to distinguishing between hypotheses in A_i is called an *ordinary local network*. The set of k local networks is called a *generalized similarity network* of P. When $A_1, ..., A_k$ is a partition of *domain(H)*, then the set of k comprehensive local networks is called a *generalized Bayesian multinet*.

For example, the secured-building story is represented in the generalized similarity network of Figure 7. Note, $H = \{h_1, h_2\}$ and *domain(H)* consists of nine elements (x, y) where both x and y are drawn from the set $\{w, v, s\}$. A connected cover of *domain(H)* upon which Figure 7 is based consists of: $\{(s,s)(v,s)(s,v)(v,v)\}$, $\{(v,v)(w,v)(v,w)(w,w)\}$, and $\{(s,s)(s,w)(w,s)\}$. This cover is connected.

Most asymmetric independence assertions encoded in Figure 7 were either explained in previous sections or are obvious from the verbal description of the story.

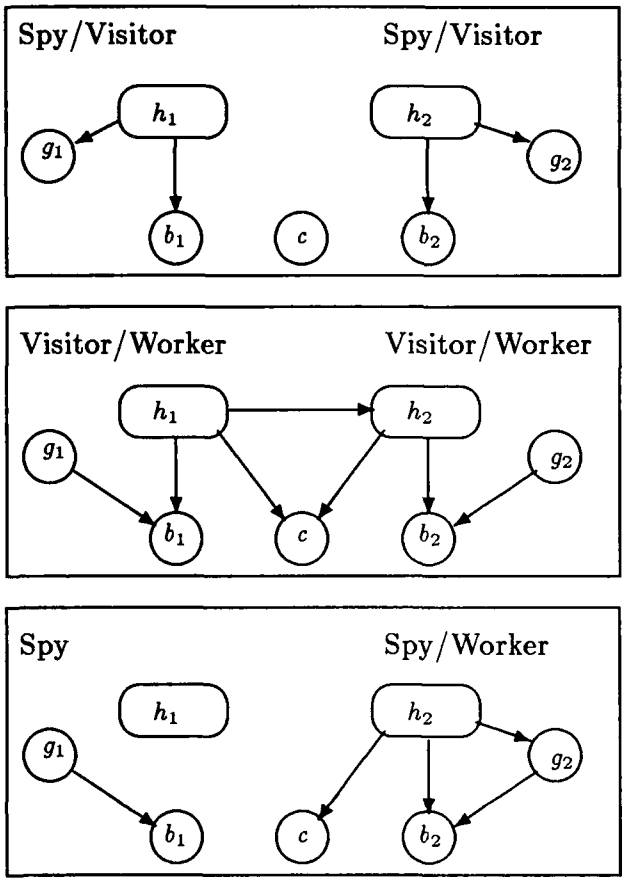


Figure 7: A generalized similarity network with two hypothesis nodes.

The absence of a link between h_1 and h_2 in the top network encodes the fact that if the guard knew that one person is a spy, this knowledge would not help him/her decide whether the other person is a spy or a visitor. The existence of a link between h_1 and h_2 in the middle network encodes the fact that workers come in pairs more often than do visitors. Hence the knowledge that one person is a worker is a clue for classifying the other person.

The vanishing dependence between hypothesis variables h_1 and h_2 in case of spies vs. visitors is an example of *inter-hypothesis independence*. Such asymmetric assertions cannot be encoded in ordinary similarity networks.

Summary

This paper proposes an efficient format for encoding and using asymmetric independence assertions for inference. The model builder is asked to express knowledge about independence by constructing multiple local networks using informal guidelines of causation and time ordering. Like any Bayesian network, local networks possess precise semantics in terms of independence assertions and these can be used to verify 1) whether the network faithfully represents the domain and 2) whether the input is consistent.

Multiple local networks have several advantages compared to a single Bayesian network. The elicitation of several small networks is easier than eliciting a single full-scale Bayesian network because the expert can focus his/her attention to particular subdomains, and hence, provide more reliable judgments. Multiple networks represent a domain better because more knowledge about independence is qualitatively encoded. Algorithms for finding the most likely hypothesis run faster when using multiple networks. And finally, the overall storage requirement of multiple networks is often smaller than that of a single Bayesian network because as independence assertions become more detailed, less numeric parameters are needed for describing a domain.

Notably, when independence assertions in the domain are symmetric, a single Bayesian network is preferable.

The challenges remain to 1) devise additional graphical representation schemes of salient patterns of independence assertions, (2) provide computer-aided elicitation procedures for constructing these representations, and (3) devise efficient inference procedures that make use of the encoded assertions.

References

Geiger D., and Heckeman, D. (1990). Separable and transitive graphoids. Sixth Conference on Uncertainty in Artificial Intelligence.

Heckerman, D. (1990). *Probabilistic Similarity Networks.* PhD thesis, Program in Medical Information Sciences, Stanford University, Stanford, CA.

Contingent Influence Diagrams. Submitted for publication.

Lauritzen, S.L.; and Spiegelhalter, D.J. 1988. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems (with discussion). *Journal Royal Statistical Society*, B, 50(2):157-224.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA.

Shachter, R. (1986). Evaluating Influence Diagrams. Operations Research 34:871-882.

Verma, T. and Pearl, J. (1988). Causal networks: Semantics and expressiveness. In *Proceedings of Fourth Workshop on Uncertainty in Artificial Intelligence,* Minneapolis, MN, pages 352–359. Association for Uncertainty in Artificial Intelligence, Mountain View, CA.

Graphical Models and Their Representation

Colin Goodall*
Columbia University
Princeton University

H. Mathis Thoma†
Ciba-Geigy
Summit, NJ 07901

1 Introduction

In a multivariate Gaussian model, the presence of a zero in the inverse variance matrix, or in the partial correlation matrix, implies that the two variables are independent given the rest. Thus the dependence between variables can be fully represented by a graph, in which the absence of an edge implies conditional independence. This leads to the term graphical Gaussian model, and further to theorems concerning the equivalence of the local, global and pairwise Markov properties of the graphical model. For discrete distributions (or other multivariate continuous distributions), this graphical representation is ambiguous, as the interactions may involve more than two variables at a time. By convention, the presence of a clique of k variables in a graph representing a cross-classified multinomial distribution implies that the joint distribution includes a term in all k variables. The distribution does not in general factorize into $\binom{k}{2}$ pairwise components. However, a hypergraph gives a natural, unambiguous, representation.

A hypergraph comprises a set of nodes (or variables) together with a set of hyperedges. Each hyperedge is a subset of the set of nodes, with the constraints that no hyperedge is the empty set (ϕ), and the union of all hyperedges is the set of nodes. Thus, the presence of a given hyperedge implies a corresponding factor, involving one or more variables.

To demonstrate the flexibility and utility of hypergraphs, we consider hypergraph representations of graphical association/conditional Gaussian (CG) models for both discrete and continous variables (Lauritzen and Wermuth, 1989), and their generalization to hierarchical interaction/CG models (Edwards 1990). Edwards (1990, p.5) gives the example of two discrete and two Gaussian variables and draws the independence graph for the model in which the two discrete variables are conditionally independent, and likewise the two continuous variables. With other models for these four variables the graphical representation breaks down, but the hypergraph representation does not. CG models provide some of the most exciting applications of graphical modeling; we focus on the special case of ANOVA, allowing heterogeneous variances.

The Gibbs-Markov equivalence says that if a strictly positive distribution satisfies the conditional independences induced by a graph (through graph separation), then this distribution is the product of functions carried by the cliques of the graph, and vice-versa. Later in the paper we will reformulate this equivalence in terms of hypergraphs as follows. We will replace the conditional independences with their equivalent factorizations into two factors (2-factorizations), and we will introduce the meet operation on hypergraphs, which will allow us to combine several 2-factorizations into one factorization with $n \geq 2$ factors. This has two advantages: First, it will show that only certain factorizations can be described through the conditional independences they induce. Second, using methods from the theory of relational databases, we give conditions that generalize the equivalence in a weaker form to distributions that are not strictly positive.

2 Conditional Gaussian models

The conditional Gaussian model (Edwards, 1990, Whittaker, 1990) specifies the joint distribution of a set I comprising k discrete variables and a set Y comprising q continuous variables to be

$$f_{IY}(\mathbf{i},\mathbf{y}) = f_I(\mathbf{i})\ f_{Y|I}(\mathbf{y}|\mathbf{i}), \tag{1}$$

the product of a cross-classified multinomial distributions f_I and a multivariate Gaussian density $f_{Y|I}$ separately in each cell $\mathbf{i}$. The moment parametrization of (1) is

$$f_{IY}(\mathbf{i},\mathbf{y}) \;=\; f_I(\mathbf{i})\ \frac{1}{(\sqrt{2\pi})^q\ |\sum_{\mathbf{i}}|}\ \times \tag{2}$$

*Supported by Columbia University and Army Research Office grant DAAL03-88-K-0045 to Princeton University

†Supported by Office of Naval Research grany N00014-85-K-0745, Army Research Office grant DAAL03-86-K-0042, and National Science Foundation grants DMS85-03362 and DMS85-04332 to Harvard University

$$\exp\left\{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}_i)^T \sum_{\mathbf{i}}^{-1} (\mathbf{y}-\boldsymbol{\mu}_i)\right\}$$

and the canonical parametrization is

$$f_{IY}(\mathbf{i},\mathbf{y}) = \exp\left\{\alpha_{\mathbf{i}} + \boldsymbol{\beta}_{\mathbf{i}}^T\mathbf{y} - \frac{1}{2}\mathbf{y}^T D_{\mathbf{i}}\mathbf{y}\right\} \qquad (3)$$

where each scalar parameter included in $(\alpha_{\mathbf{i}}, \boldsymbol{\beta}_{\mathbf{i}}, D_{\mathbf{i}})$ is expanded using all subsets of I,

$$\alpha_{\mathbf{i}} = \sum_{a\subseteq I} \lambda_{\mathbf{i}}^a \qquad \boldsymbol{\beta}_{\mathbf{i}} = \sum_{a\subseteq I} \boldsymbol{\eta}_{\mathbf{i}}^a \qquad D_{\mathbf{i}} = \sum_{a\subseteq I} \psi_{\mathbf{i}}^a. \quad (4)$$

These are called the discrete, linear and quadratic parts respectively. Models are specified by restricting some elements of the $\lambda_{\mathbf{i}}^a$, $\boldsymbol{\eta}_{\mathbf{i}}^a$, and $\psi_{\mathbf{i}}^a$ to zero.

Lauritzen and Wermuth (1989) develop CG graphical association models, for which a graphical representation suffices (with the clique convention for discrete variables). The attractiveness of a graphical association model is that the complete set of conditional independence statements can be read from the graph (hence the term independence graph). The model is fully specified by these ternary statements of the conditional independence of pairs of variables given the rest. However, graphical association models are unnecessarily restrictive. Edwards (1990) gives the theoretical basis for the analysis of conditional independence in hierarchical interaction models. He defines the hierarchical interaction models to be the CG models that satisfy the marginality principle. Briefly, if $\lambda_{\mathbf{i}}^a$ is not identically zero, then neither is each of $\lambda_{\mathbf{i}}^b$ for $b \subseteq a$. If the rth element $\eta_{\mathbf{i}r}^a$ of $\boldsymbol{\eta}_{\mathbf{i}}^a$ is not identically zero than neither are $\eta_{\mathbf{i}r}^b$ and $\lambda_{\mathbf{i}}^b$ for $b \subseteq a$. If the rsth element of $\psi_{\mathbf{i}}^a$ is nonzero, then neither are $\psi_{\mathbf{i}rr}^b$, $\psi_{\mathbf{i}rs}^b$, $\psi_{\mathbf{i}ss}^b$, $\eta_{\mathbf{i}r}^b$, $\eta_{\mathbf{i}s}^b$, and $\lambda_{\mathbf{i}}^b$ for $b \subseteq a$.

Hypergraph representation of hierarchical interaction models. We demonstrate that hierarchical interaction models can be represented using hypergraphs (although, as will be seen, the quadratic parts cause some difficulties). In Section 4 we carry across some of the basic properties of independence graphs to hypergraphs. In so doing, we argue that it is better to emphasize factorizations, read directly from the set of hyperedges than conditional independence statements. In particular, in modeling data by ANOVA it is natural to think in terms of several overlapping subsets of mutually dependent variables, each a hyperedge.

The marginality principle allows us to use the *reduced* hypergraph, that includes a hyperedge corresponding to each maximal subset of variables. The hierarchical interaction model is especially demanding, and requires that there are two types of hyperedges. A type 1 hyperedge (Figure 1) corresponds to a maximal discrete or linear part in (4). A type 1 hyperedge containing only discrete variables corresponds to some $\lambda_{\mathbf{i}}^a$ with a maximal. A type 1 hyperedge containing k' discrete and q' continuous variables corresponds to a $q' \le q$ subvector of $\boldsymbol{\eta}_{\mathbf{i}}^a$ where $|a| = k'$. By convention, the presence of two or more continuous variables in a type 1 edge does not imply an association between them. A type 2 hyperedge (Figure 2) also includes both discrete and continuous variables, and corresponds to a maximal quadratic part $\psi_{\mathbf{i}}^a$. When there is more than one continuous variable in a type 2 edge, the pairwise interactions are implied. Type 2 edges must be nested inside type 1 edges, and where type 1 and type 2 edges coincide, the type 1 edge may be omitted.

If the marginality principle were to be dropped, two types of edges would suffice, but the nesting property would fail, and a reduced hypergraph could not be used.

3 Analysis of variance

To illustrate the hypergraph representation, and to motivate the use of hierarchical interaction models, we turn to the CG regression setting, that is, the conditional part $f_{Y|I}(\mathbf{i}|\mathbf{y})$ keeping $f_I(\mathbf{i})$ fixed. With a single continuous variable Y and k factors, this is the analysis of variance model. Our principal concerns have a practical flavor:

1. The set of possible models includes the lattice of 2^k models for the linear part, each multiplied by some number of models for the quadratic part. How is backward or forward model selection to be viewed as "local operations" on hyperedges?

2. Graphical models, fit by maximum-likelihood, are commonly compared using the analysis of deviance. How adequate are the χ^2 approximations when exact F-ratios are available?

3. Hierarchical interaction models allow unequal variances to be modeled readily. How important is this feature?

4. In the classical approach to ANOVA, the experimental design places restrictions on the models to be selected. What is the analog for graphical models?

5. What useful information is contained in the independence structure of $f_I(\mathbf{i})$?

Example 1. Pilot plant data. Box, Hunter and Hunter (1978) give pilot plant data, of chemical yield Y measured at two replicates of a 2^3 design, with factors

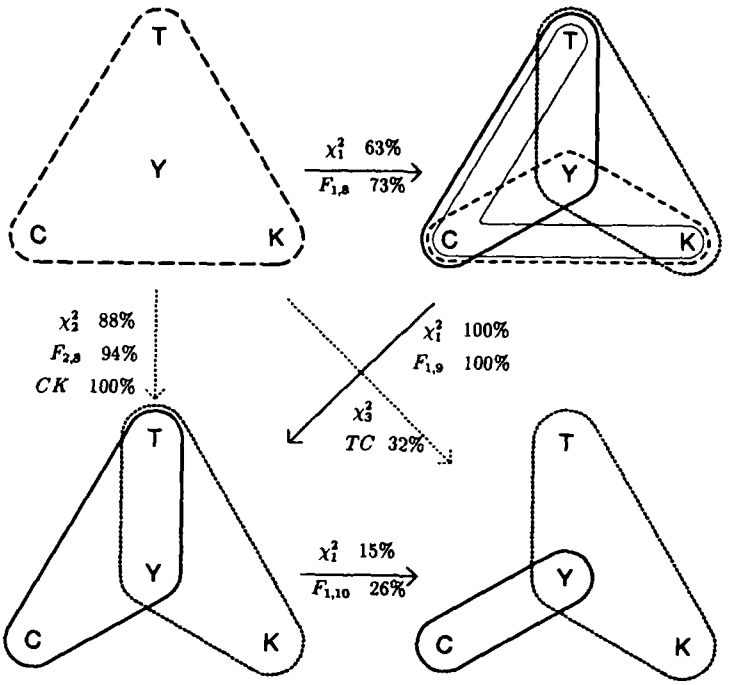


Figure 1: ANOVA of pilot plant data using hypergraphs

temperature T, concentration C, and catalyst K. Conventional ANOVA (Table 1) shows that T, C, and TK are significant at 1%.

Table 1. Pilot plant ANOVA

	DF	SS	F-ratio	P-value
T	1	2116	265	0%
C	1	100	13	1%
K	1	9	1	32%
TC	1	9	1	32%
TK	1	400	50	0.01%
CK	1	0	0	100%
TCK	1	1	0.1	73%
Error	8	64		

We model these data using hypergraphs, with constant variance ($\psi^{\phi}_{\mathbf{i}} = 1/\sigma^2$, $\psi^{a}_{\mathbf{i}} = 0$ otherwise). Each hyperedge includes Y and some subset of T, C, and K. We always include the complete model for the discrete variables, that is, the hyperedge $\{T, C, K\}$, so that the deviance in comparing the $f_{IY}(\mathbf{i}, \mathbf{y})$ is a comparison of the $f_{Y|I}(\mathbf{y}|\mathbf{i})$.

Backward elimination algorithm.

1. Start with the hypergraph containing the single maximal edge $\{Y, I\}$.
2. Replace in turn each maximal hyperedge, containing k' discrete variables and Y, with k' hyperedges each containing $k' - 1$ discrete variables and Y. (This eliminates the k'-factor interaction.)
3. Choose the model at step 2 with the minimum deviance difference. If that difference is statistically significant, stop. Otherwise, reduce the resulting hypergraph and return to step 2.

Modifications are available when $q > 1$, and for forward selection, or a stepwise procedure. The backward elimination algorithm can be implemented using a graphical user interface (see Figure 1), and then the use of hypergraphs would eliminate the need for difficult modeling formulae (Edwards (1990) and Whittaker (1990)).

Figure 1 shows backward elimination for the pilot plant data. The three steps are (1) UL to UR: eliminate TCK interaction (with Y); (2) UR to LL: eliminate CK interaction and reduce; (3) LL to LR: eliminate TC interaction and reduce. Notice that the hyperedge TCK is shown at UR, but is implicitly included in the models at LL and LR also. The arrows are annotated with the

P-values from both the analysis of deviance and F-ratios (Table 1). The two sets of P-values are reasonably close. Each model is also referred to the original model (χ^2_2 and χ^2_3 respectively). As the design is balanced and complete, independent tests of the two-way interactions CK and TC are available from conventional ANOVA (Table 1). Modeling with hypergraphs leads to a hierarchical interaction model that includes the K main effect (contrary to Table 1). Notice that because the hyperedge TCK is implicitly present, we may not conclude from Figure 1, LR that $C \perp K|Y$.

Example 2. Dental golds data. Hoaglin, Mosteller and Tukey (1991) and Goodall and De Veaux (1990) include extensive analyses of data on the hardness Y of dental gold, produced using three methods M at three temperatures T from two alloys A by five dentists D. A model for the linear part is $\{YMTD,\ YMAD\}$. Figure 2 shows hypergraphs with four choices of quadratice part. The analysis of deviance is given in Table 2. The linear part may be refined further. Each choice in Table 2 is permissible with linear part $\{YMTD,\ YMA\}$ but the fourth is not with $\{YTD, YMA\}$.

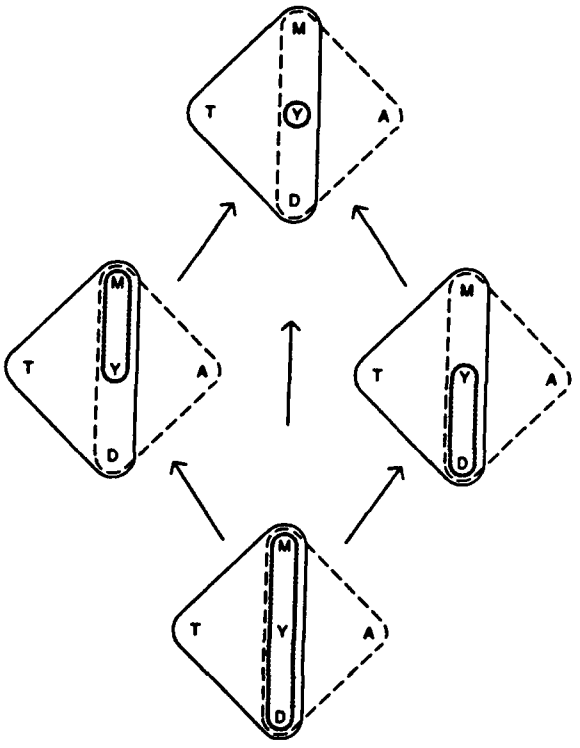


Figure 2: Models for heterogeneous variances in dental golds data

Table 2. Analysis of deviance for dental golds (The P-value compares each deviance to the first, homogeneous model)

Quadratic	Deviance	Deg.Free	P-value
Y	674	119	-
YM	671	117	27%
YD	666	115	9%
YMD	651	105	6%

Role of experimental design. Just as in the classical approach to ANOVA, the experimental design limits the models that can be considered. A k-factor factorial design without replication includes no $(k+1)$-vertex hyperedge. Model selection by backwards elimination in the dental golds example begins with four three-factor interactions.

When there is confounding we assume that some terms, typically the high order interactions, are zero. For example, the resolution V 2^{5-1} fractional factorial with defining relation I = 12345 confounds 1 and 2345, 12 and 345, etc. The maximal model includes $\binom{5}{2}$ three-vertex hyperedges (a factorization). The resolution IV 2^{4-1} design with I = 1234 confounds 1 and 234, 12 and 34, etc. Setting the three-way interactions and three two-way interactions to zero leaves four maximal models each with three two-way interactions.

In the design of experiments, a preliminary factorization of the variables may be used to decide on an appropriate design. For example, if it is believed that the two-way interaction 12 is zero, but the three-way interaction 345 is non-zero, the resolution V design above may be used with a different initial maximal model in the backwards elimination algorithm. In a future paper we will discuss the relationship between factorization and experimental design in greater detail.

Factorizations of the discrete part. Given two discrete variables A and B each at two level, suppose proportional allocation, that is, f_I factorizes. Then it is easy to show that the estimates of A and B main effects are independent (in a main effects only model). More general statements are true: These relate the factorization of f_I to independence statements about β, the regression coefficients, since var $\beta = (\mathrm{X^T X})^{-1}$, where X is the matrix of dummy variables.

4 Hypergraph Factorizations

Factorizations in graphical models. Graphical models are usually defined in terms of conditional independence, and are represented using either directed or undirected graphs (see for example Whittaker, 1990, or Pearl, 1988). However, any conditional independence

statement is equivalent to a factorization of the overall distribution (or one of its margins) into two factors. This equivalence can be exploited to give a new graphical representation of conditional independence statements and the rules that govern them. Such representations are based on hypergraphs instead of graphs, which gives them several advantages: (1) Hypergraphs are mathematically simpler than the ternary conditional independence relation. (2) It is natural to consider factorizations involving more than two factors, but conditional independence does not allow for such a generalization. (3) Factors can be identified with independent but overlapping subsystems where the variables outside the overlap are independent, offering a convenient modeling paradigm.

We argue that the concept of factorization forms a more general and more convenient mathematical foundation for the theory of graphical models than does conditional independence. This point of view has been pursued in Thoma (1989) in a different context and will be the subject of a forthcoming paper by the authors.

Below we will focus on two important aspects of this idea. First we will show how conditional independence relations, respectively their equivalent factorizations, can be represented graphically. Secondly we will focus on the description of arbitrary factorizations through sets of conditional independence statements. This is the content of the Gibbs-Markov Equivalence, a fundamental result in the theory of graphical models. The equivalence holds only for strictly positive distributions. Using ideas from the theory of relational databases it is possible to extend the equivalence, in a weaker form, to arbitrary distributions.

Conditional Independence. Consider the set $U = \{V_1, \ldots, V_n\}$ of random variables. To avoid difficulties with regularity of the underlying measure, and thus to focus on the hypergraph representation, we assume that all variables have finite outcome spaces. However, all properties discussed below can be extended to very general distributions, including hierarchical interaction models. Let X, Y, and Z be three disjoint subsets of variables in U. Set X is *independent of* Y *given set* Z, written as $X \perp\!\!\!\perp Y \mid Z$, if $f_{XY|Z} = f_{X|Z} \cdot f_{Y|Z}$. If X and Y are conditionally independent given Z, then there exist two functions g_{XZ} and h_{YZ}, such that $f_{XYZ} = g_{XZ} \cdot h_{YZ}$. Here g_{XZ} and h_{YZ} are functions that depend only on some of the variables, those in $X \cup Z$ in the case of g_{XZ}, and those in $Y \cup Z$ for h_{YZ}. We will say that these functions are *carried* by their respective sets of variables. Note that g_{XZ} and h_{YZ} are usually not margins of f_{XYZ}.

There are a number of well known rules that govern conditional independence. See for example Whittaker (1990) or Pearl (1988). We give four axioms, which we call the *coarsening*, *projection*, *substitution*, and *intersection* axioms respectively. Let X, Y, Z, W be four disjoint subsets of variables in U. Writing XY for $X \cup Y$, the axioms are

1. $X \perp\!\!\!\perp YW \mid Z \Rightarrow X \perp\!\!\!\perp Y \mid WZ$
2. $X \perp\!\!\!\perp YW \mid Z \Rightarrow X \perp\!\!\!\perp W \mid Z$
3. $X \perp\!\!\!\perp Y \mid WZ$ and $X \perp\!\!\!\perp W \mid Z \Rightarrow X \perp\!\!\!\perp YW \mid Z$
4. $X \perp\!\!\!\perp Y \mid WZ$ and $X \perp\!\!\!\perp W \mid YZ \Rightarrow X \perp\!\!\!\perp YW \mid Z$

The last axiom holds only if the joint distribution f_{XYWZ} is strictly positive. For completeness, two additional axioms must be added to the set of four (Pearl 1988). These are the symmetry axiom, $X \perp\!\!\!\perp Y|X \Rightarrow Y \perp\!\!\!\perp X|Z$, and the trivial independence axiom, $X \perp\!\!\!\perp \phi|Z$. Notice also that Axioms 1 and 2 provide the converse to Axiom 3, and Axiom 1 the converse to Axiom 4.

Graphical Representation. The conditional independence statement $X \perp\!\!\!\perp Y \mid Z$ can be represented graphically via its equivalent factorization as in Figure 3. If the two factors together cover all the variables under consideration (the set U, left side of Figure 3) the factorization is *full*. If they cover only a subset (right side of Figure 3), the factorization is *embedded* since this implies that only a margin of the overall distribution factors.

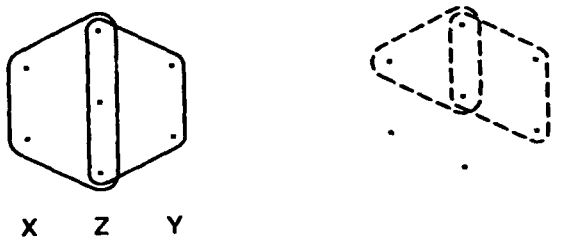


Figure 3: $X \perp\!\!\!\perp Y \mid Z$

It is possible that $f_U = f_A \cdot f_B$, where A and B are two subsets of U, but A and B do not cover U. In this case the variables not in $A \cup B$ have no influence on f_U. This leads to a small problem with our graphical representation, since we can no longer tell whether a factorization is full or embedded by looking at the set of variables covered. Thus, we distinguish the two cases by using a different color or line style to represent embedded factorizations, as shown in Figure 3.

The conditional independence $X \perp\!\!\!\perp Y \mid Z$ is equivalently described through the two sets XZ and YZ,

which indicate the factors of the corresponding factorization. The set $\{XZ, YZ\}$ is called a *scheme*. The term is borrowed from the theory of relational databases. A scheme is equivalent to a *reduced hypergraph* with two (or more) hyperedges. We will use bold capitals, **A**, **B**, ..., to designate schemes.

No conditional independence relation will result in a scheme where one component is a subset of the other. The two components of the scheme are always incomparable. However, we can consider factorizations where one factor is carried by a subset of the other factor. For example $f_{XYZ} = f_Z \cdot f_{XY|Z}$. Since it is always possible to factor in this fashion, from the factorization point-of-view we need only consider maximal factors, and therefore the reduced hypergraph. However, additional factors may aid in interpretation, for example main effects in ANOVA with interactions present.

Graphical Representation of Axioms. Using factorizations and their schemes we can represent the axioms given above in graphical form, in the following four figures. The following terminology is convenient: The *key* of scheme $\mathbf{A} = \{A_1, A_2\}$ is the set $A_1 \cap A_2$, and sets $A_1 \setminus A_2$ and $A_2 \setminus A_1$ are the *wings* of the scheme.

Axiom 1 says that from a given factorization we can derive a new one by moving wing elements to the key. This simply adds variables to the factors that do not influence the distribution.

Figure 4: Axiom 1, Coarsening

Axiom 2 says that we can derive a new factorization by clipping elements from the wings of a given one. However, the new scheme will cover fewer variables. There are simple example showing that we do not derive valid new schemes if we clip elements of the key.

Figure 5: Axiom 2, Projection

Axiom 3 shows that we can replace one factor with a factorization that covers the same variables. One of the resulting three factors can then be absorbed and we end up with a two-component scheme again.

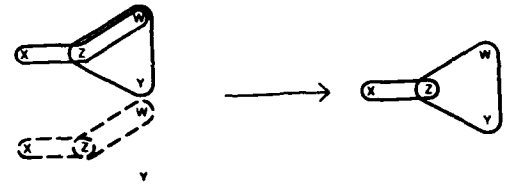


Figure 6: Axiom 3, Substitution

To formulate Axiom 4 we introduce the following definitions: If **R** is an arbitrary set of subsets of U then $\mathbf{R}^\circ$ is its *reduction*, i.e. the set of maximal elements of **R** (a component is maximal if it is not strictly contained in another component). The *meet* of two schemes **A** and **B** is the set $\mathbf{A} \wedge \mathbf{B} := \{A \cap B \mid A \in \mathbf{A}, B \in \mathbf{B}\}^\circ$. i.e. the reduction of all intersections of components of **A** and **B**. Axiom 4 says that if the distribution is strictly positive, we can infer from two given factorizations a new one, the meet of the given schemes. The two schemes must share a component to ensure that the meet comprises only two components.

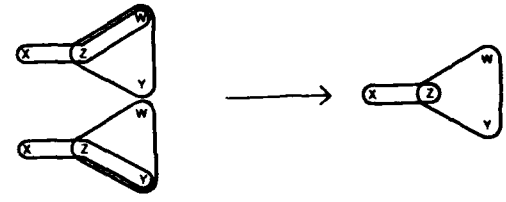


Figure 7: Axiom 4, Intersection

5 General Factorizations

Gibbs-Markov Equivalence. We now consider factorizations that involve more than 2 factors, and, correspondingly, schemes with more than two components. To distinguish the general factorization and schemes from those involving two factors, we will use the terms 2-factorization and 2-scheme for the latter. Our overall strategy is described in the Introduction.

The Gibbs-Markov Equivalence, one of the central results for graphical models, says that for strictly positive distributions a set of conditional independence statements (2-factorizations) is equivalent to a factorization involving more than two factors.

Consider the following example. Let $f = g \cdot h \cdot k$ be defined over the set of variables $U = \{A, B, C, D, E\}$. Let g be carried by margin $\{A, B\}$, h by $\{B, C, D\}$, and k by $\{D, E\}$. The distribution f factors into three components, but it is easy to derive the following three 2-factorizations simply by multiplying two of the factors:

$$\begin{aligned} f &= (g \cdot h) \cdot k \\ f &= g \cdot (h \cdot k) \\ f &= (g \cdot k) \cdot h \end{aligned}$$

Figure 8 shows the corresponding 2-schemes.

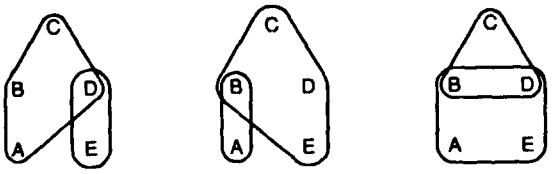


Figure 8: Derived 2-Schemes

In this particular situation we can reconstruct the original factorization from the three 2-factorizations as follows: First clip the element E from the wing of the second 2-scheme, then use the resulting scheme to replace the larger component of the first 2-scheme. The result is the original factorization. In fact, the third 2-scheme is superfluous. Note that this reconstruction is possible even if the distribution is not strictly positive.

It is not always possible to proceed as in the example. Figure 9 shows an example where it is not possible to derive any 2-factorizations.

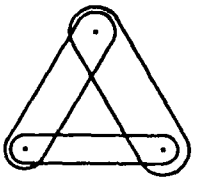


Figure 9: Simple Cyclic Scheme

We are therefore faced with the following questions: (1) Which factorization can be replaced by a set of 2-factorizations? (2) Which sets of 2-factorizations define a factorization? In addition, we need to know how to determine the set of 2-schemes that is equivalent to a given factorization, and how to determine the scheme of a factorization from a given set of 2-schemes.

Results. The results differ depending on whether the overall distribution is strictly positive or not.

If the distribution is *strictly positive*, i.e. $f_U > 0$, then any set of 2-factorizations (all involving the same set of variables) combine to give a factorization with at least two factors. Its scheme is the meet of all 2-schemes of the given 2-factorizations. Furthermore, for any factorization with a *conformal scheme* [1] there is an equivalent set of 2-factorizations. The corresponding 2-schemes can be determined as follows: Divide the components of the n-scheme into two groups, and determine for each group the union of its members. The two resulting sets form a 2-scheme. Each possible way of forming two groups will determine a 2-scheme. Some groupings may not yield a viable 2-scheme, and some groupings may yield the same 2-scheme, but overall they will determine a set of 2-schemes whose meet coincides with the scheme of the original factorization.

If the distribution is *not strictly positive*, i.e. $f_U \geq 0$, than any *conflict-free* [2] set of 2-factorizations (all involving the same set of variables) can be combined into one overall factorization. Its scheme is the meet of the given 2-schemes, and it is *acyclic* [3]. Furthermore, for any acyclic factorization there is an equivalent set of 2-factorizations. The 2-factorizations can be found using the same method as in the strictly positive case.

Distributions that are not strictly positive have a support (the set of arguments for which the distribution has non-zero probability) that does not cover the entire outcome space. Such a distribution will not factor unless its support factors too. It is therefore not surprising that the factorization properties of arbitrary distributions are closely related to those of sets. The set case has been studied extensively in the theory of relational databases, and both, terminology and results, can readily be extended from the set to the distribution case. The support of a strictly positive distribution is the entire

[1] A scheme is *conformal* if its components are equal to the cliques of a graph over the same set of variables, or equivalently, if the scheme is the meet of a set of 2-schemes (Thoma 1989).

[2] For a definition of *conflict-free* sets of 2-scheme we refer the reader to the influential paper by Beeri et al. (1983) and to the forthcoming paper by the authors, which will give a more detailed discussion of the issues involved.

[3] A scheme is *acyclic* if there is a triangulated graph over the same set of variables, such that the cliques of the graph coincide with the scheme components.

outcome space, and these distribution are therefore not subject to the restrictions that apply to sets.

References

1. Beeri, C., Fagin, R., Maier, D., Yannakakis, M. (1983), "On the Desirability of Acyclic Database Schemes," *Journal of the ACM*, **30**, 479-513.

2. Box, G.E.P., Hunter, W.G, Hunter, Hunter, J.S. (1978), *Statistics for Experimenters.* New York: John Wiley.

3. Edwards, D. (1990), "Hierarchical interaction models," *J. Royal Statist. Soc., Series B* **52**, 3-20, with discussion.

4. Goodall, C., De Veaux, R.D. (1990), "Final downsweeping: the use of Paull's rule for aggregation." Manuscript.

5. Goodall, C., Thoma, M. To appear. "Factorization as a Basis for Graphical Models."

6. Hoaglin, D.C., Mosteller, F., Tukey, J.W. (1991). *Fundamentals of Exploratory Analysis of Variance.* New York: John Wiley. In press.

7. Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems.* San Mateo, CA: Morgan-Kaufmann.

8. Thoma, M. (1989), *Factorization of Belief Functions.* Ph.D. Thesis, Harvard University.

9. Whittaker, J. (1990), *Graphical Models,* in *Applied Multivariate Statistics.* Chichester: John Wiley.

TESTGRAF: Some Graphics Tools for the Analysis of Examination Data

J. O. Ramsay
Department of Psychology, McGill University
Montreal, Quebec, Canada

1 Objectives

TESTGRAF is a program designed to graph the performance of examination questions in a way meaningful to statistically naive examiners. It was developed with the college or university instructor in mind who has given a multiple choice exam to a class of a hundred or more students, and who wants to evaluate test items with a view to

- deciding whether or use or reject an item in determining the final grade,
- getting information that will help in the rewriting of items for future use,
- identifying items which might be added to a pool for constructing subsequent exams,
- determining aspects of student performance on the test as a whole.

The program also generates examinee ability estimates which are optimal in the sense that they use the subtantial information provided by which wrong options were chosen for incorrectly answered questions. The ability estimates are also optimal in a statistical sense (maximum likelihood conditional on item characteristics), and thus automatically weight test items by their efficiencies. These ability estimates are therefore substantially more efficient than the traditional percentage correct estimates.

TESTGRAF also has a module aimed at showing examinees how much information is provided by the exam about their ability or proficiency in the subject being tested.

2 Characteristic Curves

The central concept in the modern statistical theory of tests is the *item or option characteristic function*, shown in Figure 1. Ability is viewed as a latent variable which indexes the probability that a specific answer or option will be chosen among those presented for a given test item. The function $P_{im}(\theta)$ plotted in Figure 1 for each of the five options for Item 1 is the probability that option m will be chosen for item i by examinees at or near ability level θ.

In Figure 1 the solid line indicates the probability that the option is chosen that is designated by the examiner as correct, and as one might hope, it shows that examinees with low ability have a small probability of getting the item correct, but that this probability increases rapidly over ability values 55 to 70, after which the probability of chosing the correct answer is very high. The dashed curves show the corresponding probabilities that the vari-

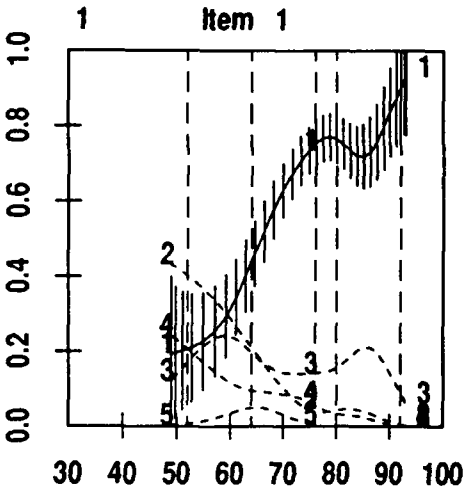


Figure 1: Option Characteristic Curves

ous wrong options will be chosen, and we observe that option 2 is especially popular with the weakest examinees, while option 3 tends to attract those with high ability and hardly anyone chooses option 5. The 5%, 25%, 50%, 75%, and 95% quantiles of the actual distribution of percentage correct (the traditional and usual scoring scheme) are indicated by the vertical dashed lines. Vertical bars on the correct answer curve show 95% pointwise confidence limits for this function.

3 Ability θ

It should be appreciated at the outset that the latent variable θ designed to capture unidimensional variation among examinees in knowledge, proficiency, or ability is not an independent variable, but rather an index for a family of Bernoulli probability distributions. As such it is only defined to within an arbitrary order-preserving transformation g, since if $\xi = g(\theta)$, then defining $P^* = P \circ g^{-1}$ implies $P^*(\xi) = P[g^{-1}(\xi)] = P(\theta)$. This means that the essential task is to estimate the rank of examinee $a, a = 1, \ldots, N$, after which the ability values θ_a can be assigned by any convenient order-preserving transformation of the N ranks.

Consequently, ability values are assigned as follows:

Step 1: Use some statistic T_a to order examinees. By default TESTGRAF uses the conventional proportion correct to do this, but TESTGRAF also permits the user to input any set of values, including the result of some other type of scoring of the exam, results from other exams, or ability estimates from a previous TESTGRAF analysis.

Step 2: Assign the quantiles of the standard Gaussian distribution $\theta_a = z_a$ to the ordered examinees. Since most examination administrations tend to produce approximately Gaussian exam scores, this permits the ability values to roughly reflect the statistical properties of familiar exam scores.

4 Estimation of $P_{im}(\theta)$

The option characteristic function is estimated by kernel smoothing of the bivariate relationship between ability θ_a and the binary variable y_{ima} taking the value 1 if examinee a of ability θ_a chose option m for option i, and zero otherwise. Kernel smoothing with Gaussian Nadaraya-Watson weights is employed, so that

Step 3:

$$\hat{P}_{im}(\theta) = \frac{\sum_a w_a(\theta) y_{ima}}{\sum_a w_a(\theta)}$$

where

$$w_a(\theta) = \exp -[(\theta_a - \theta)/h]^2/2.$$

Since the number of examinees N may number in the thousands, the Fast Fourier Transform (Härdle, 1987) is used to keep the number of calculations to $O(N + M \log M)$, where M is the number of equally spaced values of θ at which the functions P_{im} are to be evaluated.

Extensive experience indicates that the bandwidth parameter h may be set to $N^{-1/5}$ in general, although the user can override this default. However, since a constant bandwidth tends to be somewhat inefficient when the independent variable is not equally spaced, and since Gaussian quantiles become sparse in the tails, this first smoothing step tends to produce rather variable curve values for $|\theta| \geq 2$. Consequently, a second smoothing step is then used:

Step 4:

The estimated function values $\hat{P}_{im}(\theta)$ are now smoothed over the M equally spaced values of θ using the variable bandwidth

$$h^* = h \exp(\theta^2/8)/2.0.$$

Finally, most instructors are familiar with percentage correct as an indicator of ability rather than the admittedly artificial standard Gaussian values. Consequently, indicating the curve for the correct option by $\hat{P}_{iC}$, the transformation $\eta(\theta) = \sum_i \hat{P}_{iC}(\theta)$, which is nearly certain to be strictly monotonic, in effect transforms Gaussian abilities into the expected number of correct items, and, when reexpressed as a percentage, tends to be more intuitive for most instructors.

5 Credibility Curves for θ

TESTGRAF can also plot the posterior density function for ability θ for selected examinees, conditional on the estimated option characteristic curves. For clarity of plotting, these curves are normalized to have a maximum of unity, and are referred to as *relative credibility curves*. Figure 2 shows an example. For examinee 2 taking a 100-item test, we see that the most likely ability value is 61%, even though the observed percent correct is only 56% (indicated by the vertical dashed line). The discrepancy is due to the fact that the maximum credibility curve estimate takes account of wrong option choices and of the efficiency of items answered correct, and hence uses more information than simply counting correct answers. The curve also indicates, by the two dashed lines under the curve, that about 95% of the posterior probability falls between 56% and 68%.

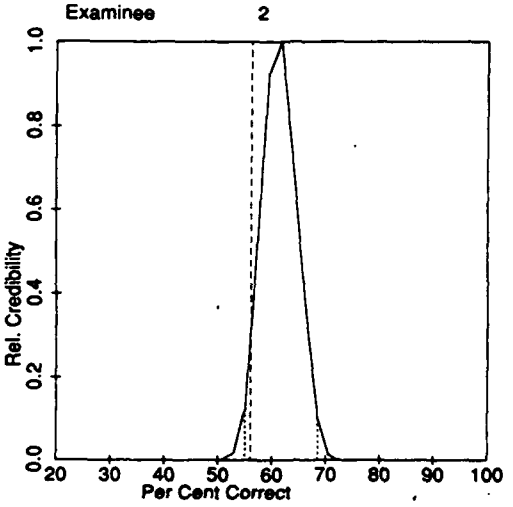


Figure 2: Relative Credibility Curves

6 PCA Display

As a summary display TESTGRAF shows each correct option curve $\hat{P}_{iC}(\theta)$ plotted at a position defined by the principal components scores for a principal components analysis of

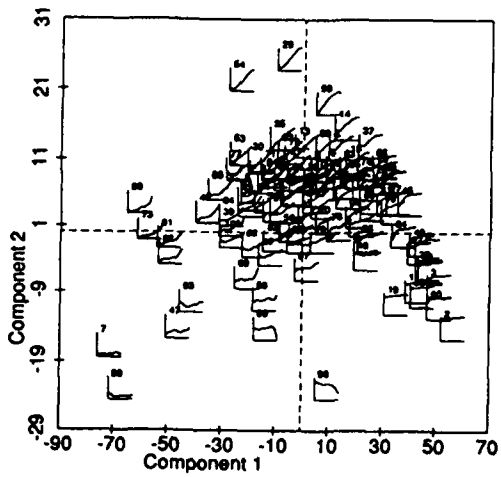


Figure 3: Principal Components of Correct Option Curves

curve values. In this analysis, the M values of θ used to plot the curves play the role of the variables in a conventional analysis, while the cases or replications are the items. Curve values are weighted by the inverse of pointwise error variances in computing the cross-products matrix on which eigenalysis is performed.

Figure 3 shows a display for a 100-item test. Here we see that the very difficult items answered correctly by very few examinees are clustered at the lower left, while the extremely easy items are found at the lower right. Items with flat or even descending correct option curves show up at the lower edges of the plot, while steeply increasing, and hence highly efficient, items are found in the upper regions.

7 Other Results

TESTGRAF also can plot other useful functions. One of these is the *test information function* $I(\theta)$, defined as the expected Hessian with respect to ability θ,

$$I(\theta) = \sum_i \sum_m \frac{[\partial P_{im}(\theta)/\partial\theta]^2}{P_{im}(\theta)}$$

This function indicates the amount of information about θ provided by the test for each level of ability, and can be used to show the ability ranges to which the test tends to be "tuned".

The program can also create a file containing the maximum likelihood estimates of ability for each examinee. These can be used to score the exam, and can also be input to TESTGRAF to provide a more efficient basis for ranking examinees.

Finally, TESTGRAF can create a file of commands which are subsequently processed by another program, TESTLASR, to produce Postscript commands for laser printer hard copies.

The program and documentation are available from the author. A small fee is requested to cover the cost of reproduction and distribution. A more complete discussion of technical aspects of TESTGRAF can be found in Ramsay (1991).

8 References

Härdle, W (1987). Resistant smoothing using the fast Fourier transform. *Applied Statistics, 36*, 104-111.

Ramsay, J. O. (1991). Kernel smoothing approaches to nonparametric item characteristic curve estimation. *Psychometrika*, To appear.

A GRAPHICAL DISPLAY FOR CHOOSING A TRANSFORMATION

Patrick J. Burns*
Statistical Sciences, Inc.
1700 Westlake Ave. N., Suite 500
Seattle, WA 98109

1 Abstract

There are three primary reasons to transform data: lack of symmetry, nonconstant variance, and interaction between factors. We present a display that has separate graphics designed to diagnose each of these conditions. The user is thus free to weigh the importance of each of these three criteria for the problem at hand, and then to choose the transformation that seems most suitable. As is the practice of many data analysts, this system uses only a few select transformations rather than transformations to arbitrary powers.

Although this system is demonstrated with data from designed experiments, it may also be used for regression problems.

KEYWORDS: robustness, symmetry, running scale estimation.

2 Introduction

Transformation can often achieve the assumptions implicit in a regression or other estimation problem. Such assumptions include: the distribution of the errors is symmetric (or Gaussian), and the variance is constant. At times a transformation can also produce a more parsimonious model.

In the present paper we use the power transformations of Box and Cox (1964). This family of transformations, which includes the logarithm, embraces those most commonly used. We also use robust estimation to ensure that the results are not unduly swayed by a few outliers.

For background on transformations, see chapters 4 and 8 (written by Emerson and Stoto) of Hoaglin, Mosteller and Tukey (1983). Also, the Box and Cox (1964) paper (and its discussion) contains many interesting comments. Robustness of transformations is discussed in Carroll (1980, 1982), and nonparametric transformations are explained in Hastie and Tibshirani (1990).

An advantage of the display being introduced is that it shows the effect of transformation on each of the criteria individually. See Sampson and Guttorp (1991) for an example in which it is desirable to attain symmetry and constant variance without destroying interaction.

3 Symmetry

Symmetry is diagnosed graphically by producing a plot based on the residuals from the fit for a particular transformation. As is done in the other plots, the residuals from a robust fit are used by default, but the least squares residuals may optionally be used.

Let $r_{(i)}$ be the ith order statistic of the residuals scaled by the (Gaussian-consistent) median absolute deviation, let n be the number of residuals and let M be the median of the scaled residuals. For each i between $n/2$ and 1, the quantity

$$(r_{(i)} + r_{(n-i)})/2 - M$$

is plotted versus the value of i. If the distribution is symmetric, this will tend to be a flat line at zero.

Since the points in this plot are dependent, the symmetry plots typically show a curve even when samples come from a symmetric distribution. It thus becomes important to have a minimum range that the y-axis spans. A glance at the asymptotic distribution of the points in the plot (Stuart and Ord, 1987, p.452) and the inspection of plots for several sample sizes and distributions led to forcing $\pm 4/\sqrt{n}$ to appear in the plot (a dashed line is drawn at these two values). When several points fall outside the dashed lines and they form a definite curve, then asymmetry may be assumed.

The plot described above is similar to plots proposed by several people; these are reviewed in Fisher (1983).

*Research supported by NSF grant ISI 88-61156

4 Homoscedasticity

To diagnose heteroscedasticity, we plot running scale estimates of the residuals versus the fit. The running scale estimate sorts the fitted values into ascending order. A certain fraction of the data enter into the estimation at each step (we have used one-half in the examples). The location for a step is considered to be the mean of the fitted values that are being used. Both the standard deviation and a robust scale estimate of the residuals (corresponding to the fitted values used) are computed at each step. The robust estimate that is used is the A-estimate of scale based on the bisquare that has an efficiency of 80 percent at the Gaussian distribution, see Burns and Martin (1991).

The test of the null hypothesis that the residuals are homoscedastic is the Spearman rank correlation test of the fit versus the absolute value of the residuals. This test was proposed by Horn (1981).

5 Parsimony

In designed experiments it is possible to make plots of the interaction of two factors; such plots were not chosen for two reasons: simplicity and generality. Since there can be a great number of pairs of factors (not to mention triples and so on), the display of interactions is a complicated task best suited to a specialized procedure. Additionally, the general regression problem is not often thought of in terms of interactions. By producing a different plot, both designed experiments and general regression problems can benefit from the same set of plots.

We selected a barchart that tells how well a simple (user-specified) model does. For both the least squares and the robust fit there is both a standard and a robust estimate of the fraction of variability explained. The standard method is the fraction of the sum of squares explained by the model. The robust method uses a τ-estimate of scale based on a Huber function with tuning constant 1.7 (Burns and Martin, 1991). Let τ denote this scale estimate with the median used as the location estimate, and let $\mathbf{y}$ and $\mathbf{r}$ denote the response and the residuals, respectively. Then the fraction of variability explained is

$$\max\left\{1 - \left(\frac{\tau(\mathbf{r})}{\tau(\mathbf{y})}\right)^2, 0\right\}.$$

6 The Display

The ingredients of the display are the four types of plot — "residual versus fit", heteroscedasticity plot, symmetry plot and parsimony plot. The allowable transformations are square, identity, square root, cube root, logarithm, inverse square root and inverse. An implementation of the display was made in S-PLUS.

All four plots are viewed for a single transformation, or one type of plot is viewed for up to four transformations.

In preparation for the display, the model is fit for each transformation both with least squares and with a robust technique. For the examples, the L_1 solution was used. This has a high breakdown point for balanced designs, but moderately low efficiency at the Gaussian model. A different algorithm should be used for the general regression problem since leverage becomes more of an issue. A high-breakdown, high-efficiency algorithm is preferred.

The user may also choose whether to use the robust residuals (the default) or the least squares residuals.

7 Example

We use the poison data discussed in Box and Cox (1964), and in many subsequent papers on transformation. This dataset consists of 4 observations on each combination of 3 poisons and 4 treatments. The parsimonious model that is used is the additive one — the response is modeled as poisons plus treatments.

Figure 1 shows the display for the response in the original units. There is clearly non-constant variance, and unevenness of the bars in the parsimony plot indicates that there is a problem with non-Gaussian errors. Both the plot for symmetry and the "residual versus fit" plot indicate that there is not symmetry. When the least squares residuals are used, there is slightly less indication that a transformation is needed; the symmetry plot is especially degraded.

Figure 2, using the inverse of the response, is close to the ideal. The fraction of variability explained is much higher and virtually the same on all four bars. The symmetry plot is bent down slightly, indicating that the inverse transform could be too strong. The running scale still has some tendency of a positive slope, which would indicate a transformation that is not quite strong enough.

Symmetry plots for four transformations are shown in figure 3. Only the plot for the identity transform shows a definite trend — the other three plots are indicating no or very slight asymmetry. The inverse square root seems to be close to the optimal transform for symmetry.

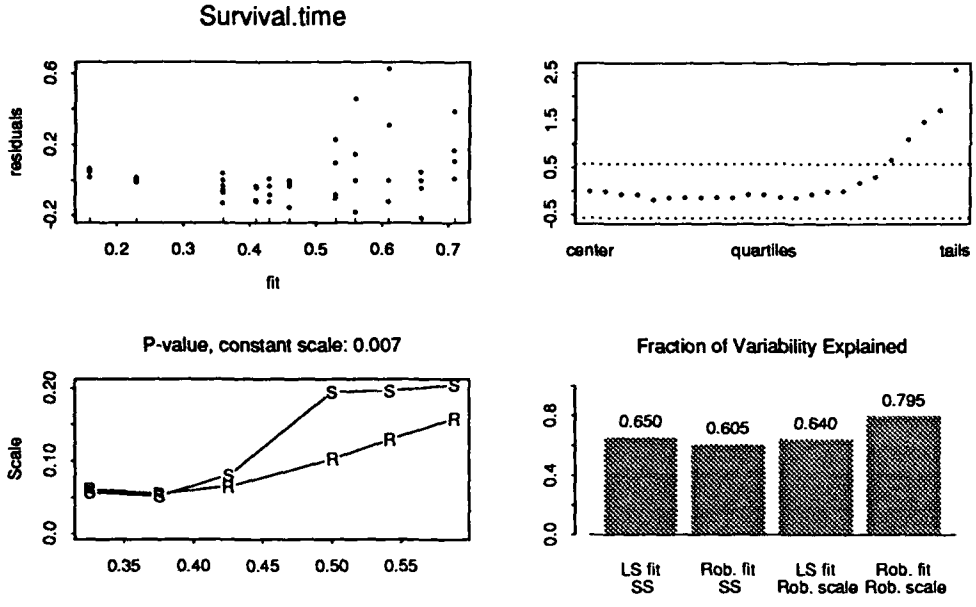


Figure 1: Poison Data, Original Scale

8 Discussion

Transformation is a common data analysis task. With the graphical display introduced in this paper a data analyst can quickly decide on an appropriate transformation or see that transformation will have little effect on the analysis.

The types of plots presented may also be used individually to explore data even when transformation is not being considered. In particular, the plot for symmetry presented here is more usable than those previously proposed because of the additional lines that indicate the significance of a curve in the plot.

9 References

Box, G. E. P. and Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society, Series B* **26** 211–252.

Burns, P. J. and Martin, R. D. (1991). One-sample robust scale estimation in contaminated models. (in preparation).

Carroll, R. J. (1980). A robust method for testing transformations to achieve approximate normality. *Journal of the Royal Statistical Society, Series B* **42** 71–78.

Carroll, R. J. (1982). Two examples of transformations when there are possible outliers. *Applied Statistics* **31** 149–152.

Fisher, N. I. (1983). Graphical methods in nonparametric statistics: a review and annotated bibliography. *International Statistical Review* **51** 25–58.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models.* Chapman and Hall; London.

Hoaglin, D. C., Mosteller, F. and Tukey, J. W. (1983). *Understanding Robust and Exploratory Data Analysis.* Wiley; New York.

Horn, P. (1981). Heteroscedasticity of residuals: a nonparametric alternative to the Goldfeld-Quandt peak test. *Communications in Statistics - Theory and Methods* **10** 795–808.

Sampson, P. D. and Guttorp, P. (1991). Power transformations and tests of environmental impact as interaction effects. *American Statistician* **45** 83–89.

Stuart, A. and Ord, J. K. (1987). *Kendall's Advanced Theory of Statistics, Volume 1.* Oxford University Press; New York.

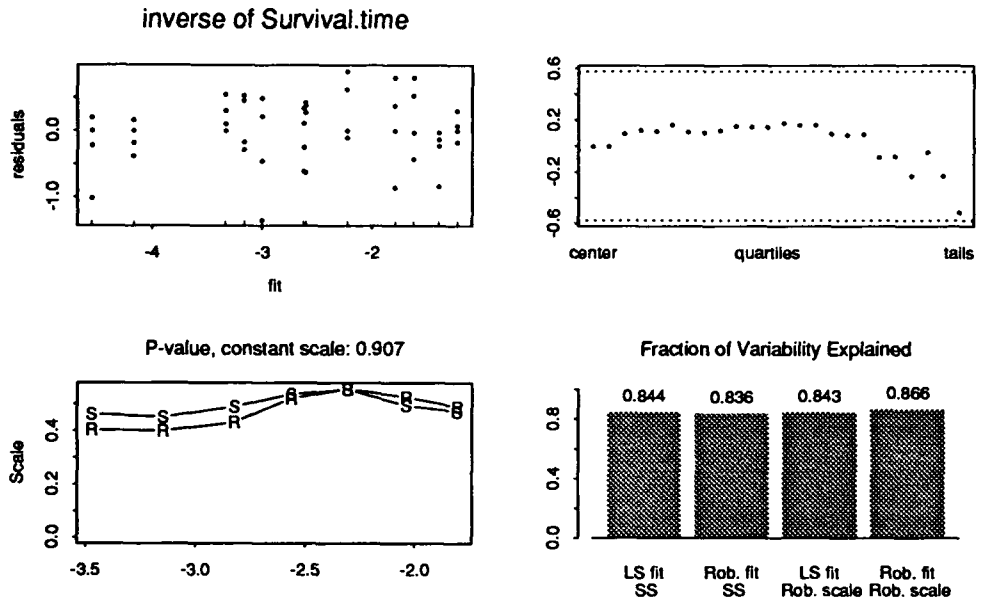


Figure 2: Poison Data, Inverse Scale

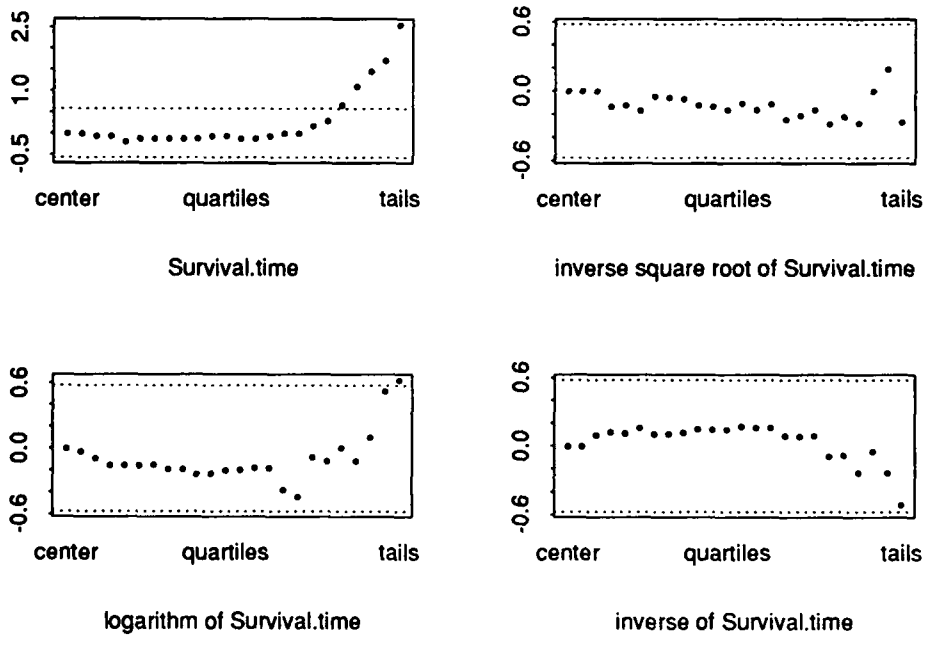


Figure 3: Symmetry Plots for the Poison Data

EXPLORATORY GRAPHICAL TECHNIQUES FOR RANKED DATA

Georgia Lee Thompson
Department of Statistics
Southern Methodist University
Dallas, Texas 75275

Graphical methods are developed for frequency distributions of fully ranked data with pseudoranks. The proposed graphical techniques use permutation polytopes, and are compatible with both Spearman's ρ and Kenmdall's τ. The problem of visualization in higher dimensions is also addressed.

1. INTRODUCTION

Graphical methods are critically needed to display frequency distributions for fully ranked data. Fully ranked data occur, for example, when judges are asked to rank n items, possibly with pseudoranks, in order of preference. Each observation is a permutation of the n distinct pseudoranks, and the resulting set of frequencies is a function on S_n, the symmetric group of n elements. Because S_n does not have a natural linear ordering, graphical methods such as histograms and bar graphs cannot be used to display frequency distributions for ranked data. Other existing graphical methods for rankings include multidimensional scaling, minimal spanning trees, and nearest neighbor graphs as discussed by Diaconis (1988). Cohen and Mallows (1980) propose graphical methods based on multi-dimensional scaling and biplots. Cohen (1990) presents alternate exploratory data techniques for ranked data.

In this paper, graphical methods are developed to display frequency distributions of fully ranked data by using permutation polytopes. A polytope is the convex hull of a finite set of points in $\mathbf{R}^n$, and a permutation polytope is the convex hull of the n! points in $\mathbf{R}^n$ whose coordinates are the permutations of n pseudoranks. To represent a set of ranked data, the frequencies with which the permutation are chosen are displayed, not on a line as is done with histograms, but on the vertices of the permutation polytope. The resulting graphical displays are especially useful as diagnostic tools because they are compatible with two commonly used metrics on S_n: Kendall's τ and Spearman's ρ. Both the τ and ρ are easily interpreted on the permutation polytope.

The permutation polytope on which the n! frequencies are displayed is inscribed in a sphere in an $n-1$ dimensional subspace of $\mathbf{R}^n$, as noted by McCullagh (1990) for ordinary ranks, in such a way to be compatible with both Kendall's τ and Spearman's ρ. Hence, for n>4, the problem of visualization of points on a polytope in higher dimensions must be addressed. One approach to this problem is to explore a higher dimensional polytope by examining its three and four dimensional faces. By defining a permutation polytope as the solution to a finite set of linear inequalities, all of the faces can be characterized. In particular, it is shown that all two-dimensional faces are combinatorily equivalent to either squares or hexagons, and all three dimensional faces are combinatorily equivalent to either truncated octahedrons, cubes, or hexagonal prisms.

2. PERMUTATION POLYTOPES FOR n=3,4

Before developing the concepts needed for the proposed graphics for either n>4 or for pseudoranks, we illustrate the proposed technique with ordinary ranks for n=3 and n=4. Ranked data can be recorded either as an ordering or as a ranking. Items are labeled with letters, and orderings are denoted by permutations of the first n letters, bracketed by < >. For example, <b,c,a,d> means that item b is ranked first, and item d last. A ranking is a permutation of n values written as a row vector $\underset{\sim}{\pi} = (\pi_1, \ldots, \pi_n)$ where π_i is the rank of the i^{th} item. The ranking corresponding to <b,c,a,d> is (3,1,2,4).

Figure 1 shows the orderings and rankings of the 6 elements of S_3. Two adjacent points are connected by an edge if their orderings differ by a pairwise adjacent transposition, or equivalently, if their rankings differ by the inversion of two consecutive values. Hence, the minimum number of edges that must be traversed to get from one vertex to another is equal to Kendall's τ. Formally, if $\underset{\sim}{\pi}$ and $\underset{\sim}{\sigma}$ are two rankings, then $\tau(\underset{\sim}{\pi},\underset{\sim}{\sigma})$ is the number of pairs (i,j) such that $\pi_i<\pi_j$ and $\sigma_i>\sigma_j$. This is equivalent to the minimum number of pairwise adjacent transpositions needed to change the ordering corresponding to $\underset{\sim}{\pi}$ into the ordering corresponding to $\underset{\sim}{\sigma}$. The placement of the vertices in Figure 1 is also compatible with Spearman's ρ:

$$\rho(\underset{\sim}{\pi},\underset{\sim}{\sigma}) = \Big(\sum_{i=1}^{n} (\pi_i - \sigma_i)^2 \Big)^{1/2}.$$

If the edges of the regular hexagon are all of length $\sqrt{2}$, then Spearman's ρ is the Euclidian distance

between two vertices. Note that the two vertices on a common edge have the same item ranked either first or last.

These ideas extend to n=4 by placing the 24 permutations on the vertices of a truncated octahedron, as shown in Figure 2 [Yemelichev et. al. (1984)]. The truncated octahedron has 8 hexagonal faces and 6 square faces. As in Figure 1, τ is the minimum number of edges that must be traversed to get from one vertex to another, and ρ is the Euclidian distance between two vertices if each edge has length $\sqrt{2}$ [cf. Schulman (1979)]. On the truncated octahedron, the 4 vertices of a square have the same 2 items ranked in the first 2 positions and the other 2 items ranked in the last 2 positions. Similarly, the 6 vertices of a hexagon all have the same item ranked either first or last. The idea that each face has a "defining property" is fundamental in the proposed graphical methods for n>4.

For $n = 3$, consider the data of Duncan and Brody (1982) in which 1439 people ranked city, suburban, and rural living in order of preference. The current residence is also recorded as a covariate. For each covariate, the relative frequencies of each permutation were calculated. In Figure 3 these relative frequencies are plotted on the vertices of 3 hexagons. Each hexagon corresponds to a covariate, and the sizes of the circles at the vertices indicate the relative values. It is immediately obvious that rural and suburban residents are similar to each other, but are both different from city dwellers. Those who prefer the city most seem to live in the city. Relatively few rural and suburban dwellers prefer their current location least, while many city dwellers would rather be anyplace else. For $n = 3$, this proposed graphical technique is similar to the graphics of Cohen and Mallows (1980) in which circles with areas proportional to the frequencies are placed at the ends of 6 vectors radiating from the origin.

The plotting of ranked data with $n = 4$ on truncated octahedrons is illustrated by the following example. At the start of a literary criticism course, 38 students read the short story and ranked 4 different styles of literary criticism in order of preference. At the end of the course, they read another short story and again ranked the same four styles of literary criticism. The 4 styles were authorial (a), comparative (c), personal (p), and textual (t); and the question of interest was whether or not the post-course rankings had moved in the direction of the teacher's own preferred ordering <p,c,a,t> [see Critchlow and Verducci (1989)]. The frequencies of the 38 pre-course rankings are shown in Figure 4a and the 38 post-course rankings are shown in Figure 4b. Most obviously, the frequencies do change a great deal between the two sets of rankings. First, there is an increase in the frequencies at the 6 vertices that correspond to orderings that begin with c. The post-course ranking do not seem to have moved *toward* the teacher's preferred ranking, <p,c,a,t>, but as concluded by Critchlow and Verducci (1989), they appear to be, over all, closer to <p,c,a,t> than are the pre-course rankings. The orderings seem to have moved *toward* <c,p,t,a>. McCullaugh and Ye (1990) illustrate a similar conclusion by plotting the vectors of the average pre- and post-course ranking on a truncated octahedron. Other observations are 1) the frequencies at the 6 vertices corresponding to the ordering ending in (c) decrease; 2) style (a) is rarely chosen as either a first or second choice after the course is completed; and 3) the incidence of style (t) as a first choice decreases.

To make the plots perceptually accurate, the areas of the circles in Figures 3 and 4 are based on Steven's Law which says that the perceived scale, p, of the size of an area is

$$p \propto (\text{area})^{.7}$$

(Cleveland, 1985). Hence, the areas of the circles are calculated as

$$\text{area} \propto f^{10/7},$$

where f is the frequency. If the areas are proportional to the values, i.e., area $\propto$ f, then small circles appear too large and large circles appear too small. Conversely, if the radius of the circle is proportional to the frequency, i.e., area $\propto f^2$, then large values are magnified and small values are minimized.

3. PERMUTATION POLYTOPES FOR n > 4

Instead of using the integers from 1 to n, some applications use pseudoranks in which a ranking is a vector whose elements are a permutation of n distinct values, and an ordering is a permutation of the n items such that the i^{th} item is assigned the i^{th} smallest pseudorank. Without loss of generality, assume that the psuedoranks are $a_1 > a_2 > \ldots > a_n > 0$. The ordinary ranks are $a_i = n - i + 1$. To extend Spearman's ρ to pseudoranks, let $\underline{a}(\underline{\pi}) = (a(\pi_1), a(\pi_2), \ldots, a(\pi_n))$ and $\underline{a}(\underline{\sigma}) = (a(\sigma_1), a(\sigma_2), \ldots, a(\sigma_n))$ be two rankings where $\underline{\pi}$ and $\underline{\sigma}$ are elements of S_n. Then

$$\rho(\underline{a}(\underline{\pi}),\underline{a}(\underline{\sigma})) = \Big(\sum_{i=1}^{n} (a_{\pi_i} - a_{\sigma_i})^2 \Big)^{1/2}.$$

Next, as in Schluman (1979), consider the set of vectors in $\mathbf{R}^n$ whose elements are permutations of the

pseudoranks. These points lie in the intersection of the sphere

$$\sum_{i=1}^{n}(x_i-\bar{a})^2=\sum_{i=1}^{n}(a_i-\bar{a})^2$$

and n − 1 dimensional hyperplane

$$\sum_{i=1}^{n}x_i=n\,\bar{a}\quad\text{where}\quad\bar{a}=n^{-1}\sum_{i=1}^{n}a_i\ .$$

The permutation polytope is the convex hull of these points, and it can be mapped into $\mathbf{R}^{n-1}$ via the Helmert transformation. Because it is an orthonormal mapping, it preserves distance and angles, so that the polytope is still inscribed in a sphere in $\mathbf{R}^{n-1}$ and Spearman's ρ (which is the Euclidian distance between two points) is preserved. When n=4 and $a_i=n-i+1$, calculations show that the resulting polytope is a truncated octahedron whose vertices are exactly the vectors of permutations and whose edges are all of length $\sqrt{2}$.

For $n>4$, the proposed graphical methods require that we relate Kendall's τ to the permutation polytope, and that we characterize all the faces, particularly in 3 dimensions. Answers to both problems are found in Chapter 5 of Yemelichev et. al. (1984) in which a permutation polytope is shown to be equivalent to the following system of constraints:

$$(1)\qquad \sum_{i\in\omega}x_i\le\sum_{i=1}^{|\omega|}a_i\quad\text{for all }\omega\in\{1,2,\ldots,n\}$$

$$(2)\qquad \sum_{i=1}^{n}x_i=\sum_{i=1}^{|\omega|}a_i\ .$$

For a given n, the faces are characterized by Theorem 3.4 of Section 5 which proves that the set of solutions to (1) and (2) is an i-dimensional face (i-face), $0\le i\le n\text{-}2$, if and only if for each such solution the inequalities in (1) are satisfied as equalities for subsets $\omega_1,\ \omega_2,\ \ldots,\ \omega_{n-i-1}$ of $\{1,2,\ldots,n\}$ such that $\omega_1\subset\omega_2\subset\ldots\subset\omega_{n-i-1}\subset\omega_{n-j}=\{1,2,\ldots,n\}$.

To use this theorem, first define $Q_k=\omega_{k+1}\backslash\omega_k$, $1\le k\le n\text{-}i$. Then, the 0-faces (vertices) are exactly the n! points whose elements are permutations of the pseudoranks because each Q_i contains exactly one element. Similarly, for a 1-face, one of the Q_k's has 2 elements and the others each have exactly one. Hence, Corollary 3.9 of Section 5 proves that 2 vertices of a permutation polytope are adjacent (on the same 1-face) if and only if they differ by a single transposition of a_k and a_{k+1}, $1\le k\le n-1$. For ordinary ranks, we now have that Kendall's τ is equal to the minimum number of edges (1-faces) that must be traversed to get from one point to another. This also extends Kendall's τ to pseudoranks in an obvious manner that warrents more study. Similarly, Theorem 3.4 shows that every 2-face is either a hexagon if all Q_k have one element except one which has 3 elements (so that all but 3 of the orderings are fixed), or a square if all Q_k have one element except for 2 of them which each have 2 elements. The 3-faces correspond to truncated octahedrons if all Q_k have one element except one (so that all but four of the orderings are fixed), to cubes if all Q_k have one element except 3 which have 2 elements each, and to hexagonal prisms if all Q_k's have one element except 2, one which has 2 and one which has 3 elements.

Thus, all 3-dimensional faces of any permutation polytope can be characterized and the data can be illustrated by a sequence of 3-dimensional polytopes in which the frequencies are plotted on the appropriate vertices. Frequently, it is useful to also plot portions of the 4-dimensional polytopes.

4. REFERENCES

Cleveland, W. S. (1985), *The Elements of Graphing Data*, Monterey: Wadsworth Advanced Books and Software.

Cohen, A. (1990), "Data Analysis of Full and Partial Rankings," Technical Report, Dept. of Industrial Engineering and Management, Technion-Israel Institute of Technology, Haifa, Israel.

Cohen, A. and Mallows, C. (1980), "Analysis of Ranking Data," Technical memorandom, Bell Laboratories, Murray Hill, New Jersey.

Critchlow, D. E. and Verducci, J. S. (1989), "Detecting a Trend in Paired Rankings," Technical Report No. 418, Dept. of Statistics, Ohio State Univ.

Diaconis, P. (1988), *Group Representations in Probability and Statistics*, Hayward: Institute of Mathematical Statistics.

Duncan, O. D. and Brody, C. (1982), "Analyzing n Rankings of Three Items," In R. M. Hansen et. al. (eds.) *Social Structure and Behavior*, New York: Academic Press.

McCullagh, P. (1990), "Models on Spheres and Models for Permutations," Technical Repoert, Department of Statistics, Univ. of Chicago.

McCullagh, P. and Ye, J. (1990), "Matched Pairs and Ranked Data," Technical Report No. 287, Department of Statitsics, Univ. of Chicago.

Schulman, R. S. (1979), "A Geometric Model of Rank Correlation," *The Amer. Statist.*, **33 (2)**, 77-80.

Yemelichev, V. A., Kovalev, M. M., and Kravtsov, M. K. (1984), *Polytopes, Graphs, and Optimisation*, Cambridge: Cambridge Univ. Press.

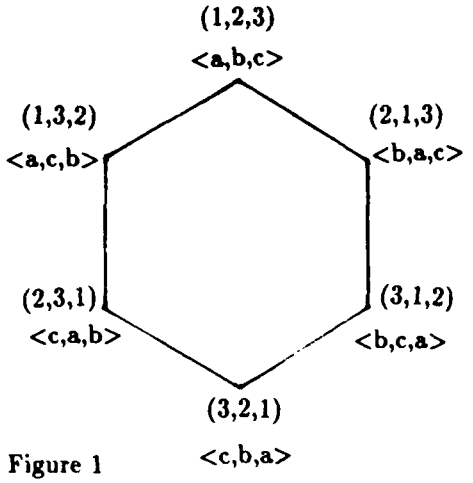


Figure 1

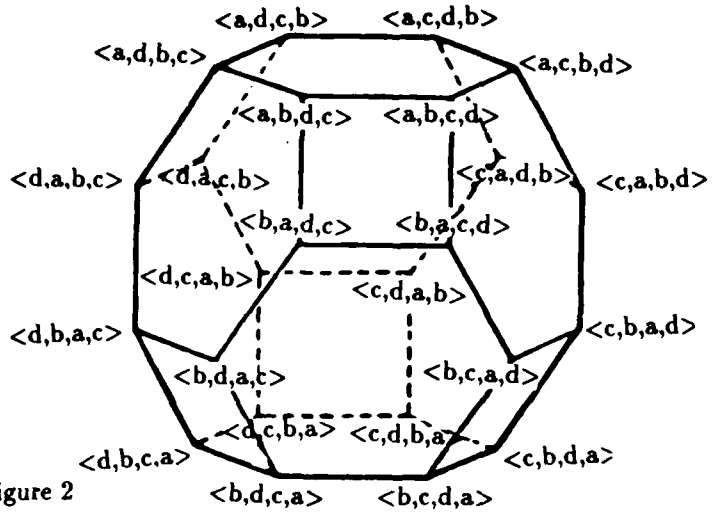


Figure 2

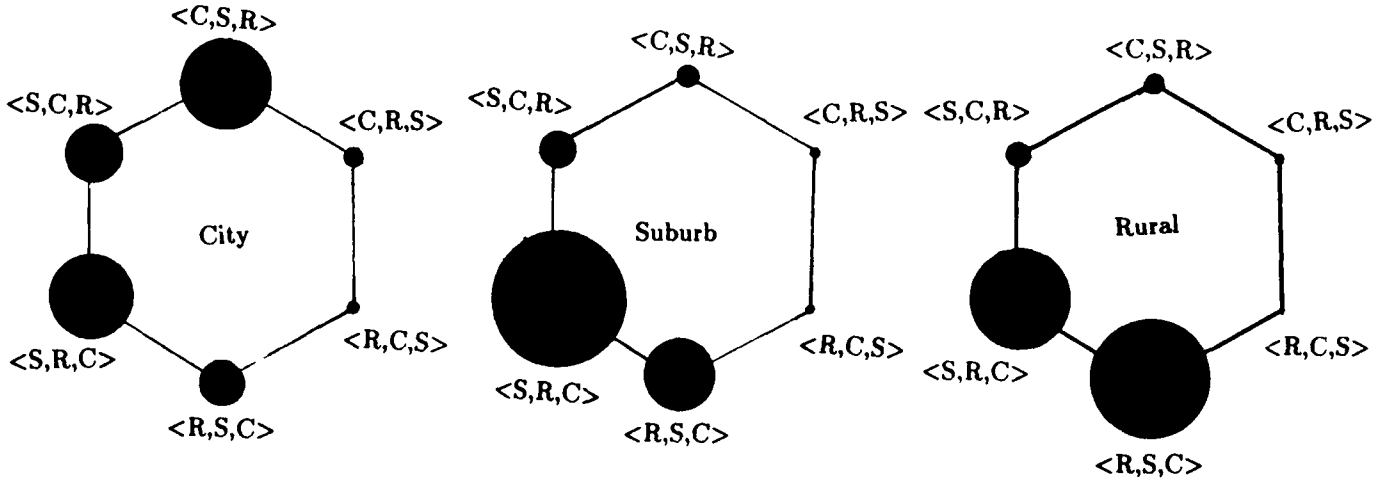


Figure 3

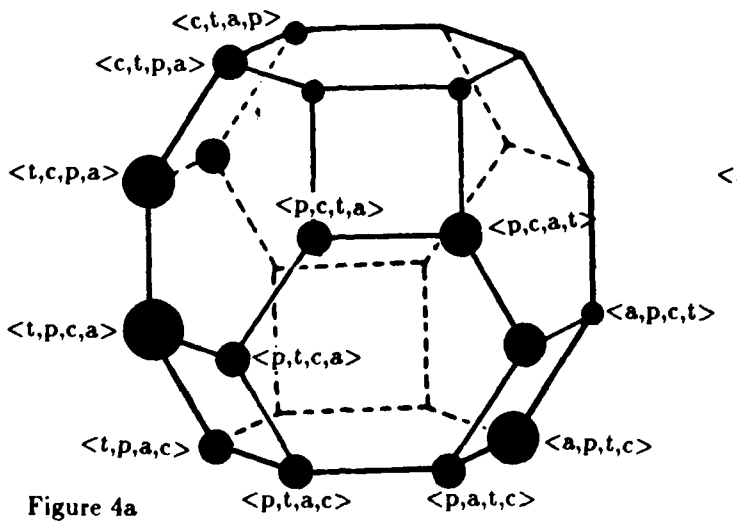


Figure 4a

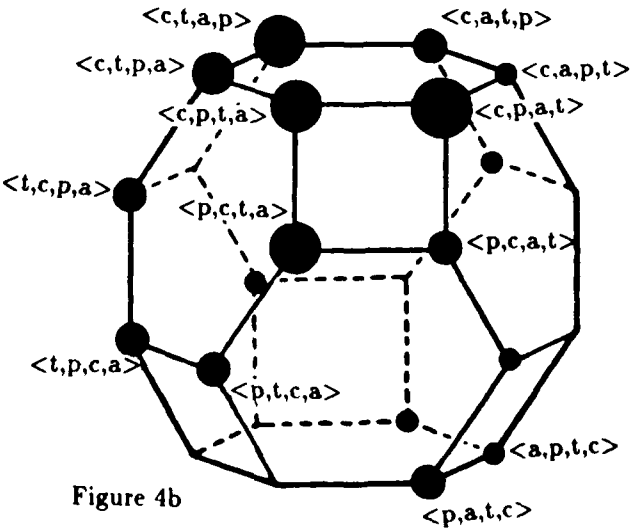


Figure 4b

Some Uses of Quantile Plots to Enhance Data Presentation

David M. Shera
428 Broadway #3, Somerville MA 02145

Abstract

Quantile plots are used to display data for better understanding and comparison of distributions. Splitting the quantile plot by a categorical variable helps one visualize an analysis of variance. Plots of of rank-transformed data corresponds to non-parametric methods and can also aid in the analysis of categorical data. As less abstract and more direct presentations of data than, for example, box plots, quantile plots can be more effective, in particular, when presenting to non-statisticians.

Definition

This paper will present quantile plots as a method of plotting actual data side by side in a way that is easily presentable to anyone, regardless of their statistical training. One simply plots the data points as an empirical quantile function [Parzen, Cleveland] which is the plot of the value of each observation on the vertical scale against the rank within the sample on the horizontal scale. The idea is that the random variable V is a function on the unit interval, [0,1]. It is closely related to the empirical cumulative distribution of V with the horizontal and vertical axes flipped. One should be careful not to confuse the meaning of "quantile plot" in this paper with the common "quantile-quantile plot" or "q-q plot" which has a slightly different definition and different use. Here, the former is a special cases of the latter, i.e., a q-q plot with uniform quantiles on the horizontal axis. This paper also pertains only to the use of quantile plots and does not involve quantile functions or their estimation. ([Parzen] has more sophisticated uses for quantile functions and related constructs.)

In Figure 1, "Score at Week 3", each observation is a single diamond. When N gets to be very large, the points tend to meld together, depending on the resolution of one's graphics device. But with such large N, the empirical distribution should be closer to the true distribution. Any quantile of the distribution is readable from this graph, in particular, the median, which is the quantile at .5 . The local density is the inverse of the local slope of the quantile function so that ranges where the slope of the points is low are regions of higher density. Extreme outliers and multiple modes are often obvious to the eye. The use of points makes the amount of ink used to print the points proportional to the size of the sample, a desirable property. Connecting the points with lines would confuse this ink/observations ratio and also emphasize what could be an incorrect interpolation.

One can easily add many features to represent various summary statistics. The interpretation of the symbols in these quantile plots is as follows:

- The data points themselves are small, hollow diamonds. Diamonds more precisely indicate position than do squares or circles. Also, they are two-dimensional which crosshairs, X's, and asterisks are not. The hollowness allows points to overlap without much loss of ink area.
- For reference, crosshairs are plotted at the quantiles of .05, .25, .50, .75, and .95. They are slightly larger than the diamonds, so as to show up in plots with many points, but do not add any more two-dimensional images to the picture.
- Small dots are placed at the corners of the "box" of the traditional box plot. (These dots may be too small to show well in this printing.)
- At the far left are five cross-hairs which represent the mean and one and two standard errors (not standard deviations). The choice to plot both one and two standard errors was to make it unambiguous as to how many standard errors were represented.
- At the far right, the crosshairs indicate the endpoints of a non-parametric 95% confidence interval for the median.
- A reference line of dots lies on the diagonal for visual anchoring. The diagonal line can be a great aid in comparing different quantile plots.

The primary purpose of these plots is to emphasize the overall shape of the distribution and adding too many extra symbols will distract the eye from this purpose. There are other common statistics which are left out:

- The standard deviation: First, for skewed data, the standard deviation marks could extend over the boundary of the plot on the high or low side, possibly onto other sections of the graph. Second, all information on the variability is contained in the quantile plot itself and the information from the standard deviation will be redundant. If it is important to a specific presentation, the standard deviation is easy to add.
- Quantile points at a variety of locations (.05, .15, .25, ...). When included in a narrow range between .4 and .6, these points tended to clutter the plot .
- Altering shapes, coloring, and shading of points was rejected in favor of having all points have equal visual impact and thus, equal importance.
- Including a smoothed version of the quantile function is certainly possible, but then one must make a choice of smoothing method. A recent example of one such method can be found in [Yang].

Here the reader may ask, "Why not use cumulative distribution plots?" The vertical orientation of the quantile plot brings gravity inherent in the page into play: areas of lower slope are more "stable" spots. In a cdf, a variable which tends to have "higher" values has a "lower" cdf, while in a quantile plot, "higher" really means "higher" in both senses of the word. If we truly think of the random variable as a function, standard convention puts the function value or

range on the vertical axis and domain on the horizontal. One more point about gravity: in a histogram with horizontal bars or a stem-and-leaf plot the larger bars might look as if they will break and fall off.

Splitting Plots and Ranked Values

By splitting the graph into separate, parallel graphs by different groups, one can perform a visual analysis of variance to supplement an ANOVA table and to provide more impact to a presentation. There are four ways to split the graph: along the horizontal axis, along the vertical axis, across pages, and overlayed. It is best to split horizontally by the the variable of most interest, for the mean/standard error and median/confidence marks will line up for easy comparison. Overlaying one quantile graph over another for comparison purposes has great appeal, but it will also cause crowding if the two quantile functions are very close. How to differentiate between the symbols from two or more separate sets of observations in an obvious way is an additional complication. Splitting vertically makes sense for confounding variables where tests for differences are less important.

Figure 2, "Score at Week 3", contains the same data as in Figure 1, but this time split by the two categorical variables Treatment and Sex. Note that there are about half as many males as females, as indicated by fewer points in the cells for males. Also, the means and standard errors, on the left side of each cell, indicate that treatment 1 is more effective. A higher score means worse in this variable, so that the further below the diagonal line the quantile plot lies, the better off the patients are. It is important to remember that the means, standard errors, and all other summary statistic symbols in the plot are not based on any particular model but only the data in each cell alone.

One could also plot rank-transformed data to graphically look at a non-parametric Kruskal-Wallis ANOVA. In this case, the diagonal line enhances the plot because it represents a theoretical distribution of rank transformed values. Here, ties are assigned the mean rank but the range of the plot runs from 0 to N, or 0 to 1 if the ranks are divided by N. When split, deviations from the overall distribution show as more points above or below the diagonal line. Figure 3, "Score at Week 3 (Ranked Values)", is again based on the same data using the ranks of the values within the whole sample rather than the values themselves. Note that the values at the top have been squeezed together and are no longer evenly spaced.

The plotting of rank transformed data is also useful for ordered categorical data, which includes dichotomous data. However, one should remember to use mean rank so that the points will not end up all at the top or the bottom of the cell, possibly merging with points from another cell. For ranked values the diagonal line should go through the centers of each level overall. In Figure 4, "Outcome (Ranked Values)", there is a single dichotomous outcome variable. Treating it as a numerical variable and creating quantile plots of the ranked values, we have one way of graphing categorical data in a 2 by 2 by 2 table. Note that the upper, right-hand cell has fewer observations and the rows of diamonds reflect the relative proportion of observations with each of the outcomes. The upper, right-hand cell has a lower rate of high outcome. But to reiterate, the means and standard errors are based only on each cell, not on any overall model.

Comparison to the Box Plot

Often one starts looking at data with a traditional box plot [Tukey], but some have been looking for improvements. One possibility is altering the shape of the box to show density [Benjamini] which requires some density estimation. One of the problems is the visual difference between the "box" and the "whiskers". What is the intuitive meaning of representing the middle half of the data with a two dimensional object and other subsets with one dimensional objects? Also, computer statistics packages can be inconsistent in their calculation of the length of the whiskers [Frigge, et al].

The box plot also does not readily reflect the actual number of observations, N. Some try to remedy this by letting the width of the box be proportional to the square root of N. However, with this alteration, different box plots are no longer comparable visually. In the quantile plot there is no need to make a mental transformation from width ($\sqrt{N}$), or whatever, to N. Sometimes there are confidence regions around the median with either "notches" in the box, or shaded blocks; but the notches or shaded blocks alter the visual weight of the primary features of the box plot.

The box plot is an abstract picture based on a handful statistics calculated from the data. There is a reduction of information in the transformation, which is fine if these statistics are the right statistics. However, if the distribution has certain peculiarities, those handful statistics may not reflect important features and instead present an inaccurate picture.

Of course, the combination of a box plot with a stem-and-leaf plot or histogram will give more information, but there are some drawbacks:

- The combination requires two graphs and uses more space and paper.
- The histogram implicitly requires a choice of division points which is a smoothing decision. Likewise, the stem-and-leaf plot also has implicit smoothing and often must round values to a convenient number of significant digits
- By continually varying, each quantile plot will be nearly unique. The endless variety of plots may hold an audience's attention longer because human beings tend to notice and be more curious about variety.

Figure 5A, "Box Plots", contains two groups of data. The first thing to notice is that group B has significantly higher values and is more spread out. Group A looks to have some outliers on both the high and low ends.

Figure 5B, "Quantile Plots of 5A", contains the same data but shows a different picture. Group B actually has what looks to be a bimodal distribution with the median falling nearly halfway between the two modes. This property did not show up in the box plot. Additionally, in group A, three suspected outliers on the high side actually turn out to be about 10% of group A. The box plot was using single asterisks to indicate what was actually more than one observation. As it turns out, after discussing this with the client, there was a systematic problem in our definition of this variable which lead to the suspicious distribution. The additional detail in the quantile plot helped identify and explain the problem much sooner.

Summary

The quantile plot is a less abstract presentation of an empirical distribution than the traditional box plot. It presents a picture closer to the statistician's own mental picture of the data and analyses. Because it displays each observation and not just an object created from certain statistics, it may be better for presentaton of data to non-statisticians. Finally, quantile plots can show features of the data that might be hidden by other methods, including problems resulting from bad data coding or calculation errors.

Bibliography

Benjamini, Y. (1988) "Opening the Box of a Boxplot", The American Statistician, 42, pp. 257-262

Chambers, J.M. et al (1983) *Graphical Methods for Data Analysis*, Pacific Grove, CA: Wadsworth and Brooks/Cole Publishing Co., p 11.

Frigge, M. et al (1989) "Some Implementations of the Boxplot", The American Statistician, 43, pp. 50-54

Parzen, E. (1979) "Non-Parametric Statistical Data Modeling," Journal of the American Statistical Association, 74, pp105-131.

Tufte, E.R. (1983) *The Visual Display of Quantitative Information*, Cheshire, CT: Graphics Press

Tukey, J. (1977) *Exploratory Data Analysis*, Reading, MA: Addison-Wesley

Yang, S. (1985) "A Smooth Nonparametric Estimator of a Quantile Function", Journal of the American Statistical Association, 392, pp. 1004-1011

Figure 1 – Score at Week 3

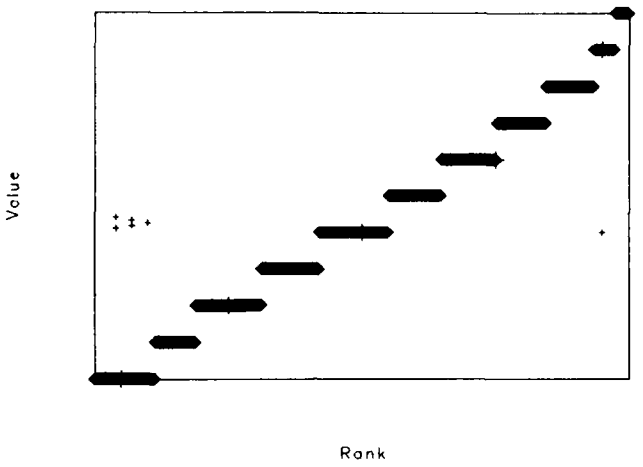


Figure 2 – Score at Week 3

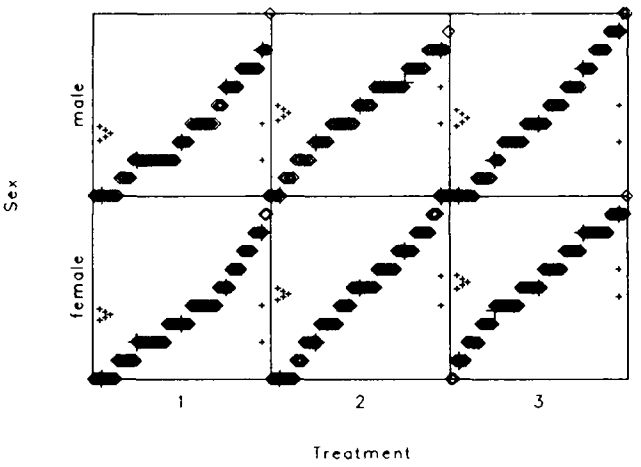


Figure 3 – Score at Week 3 (Ranked Values)

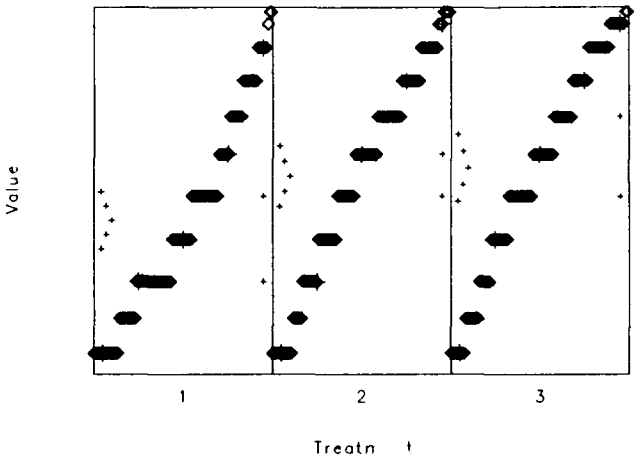


Figure 4 – Outcome (Ranked Values)

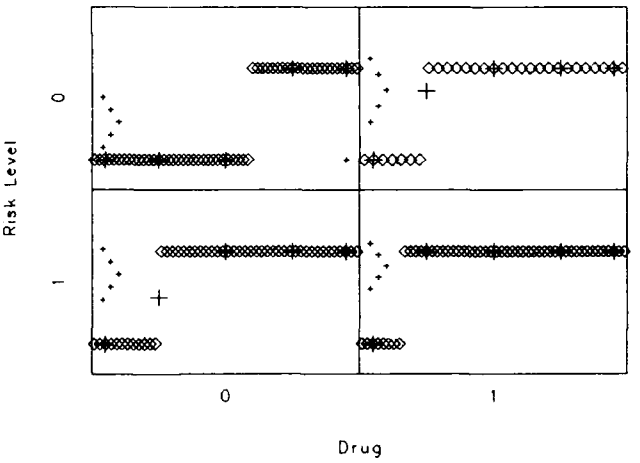


Figure 5A – Box Plots

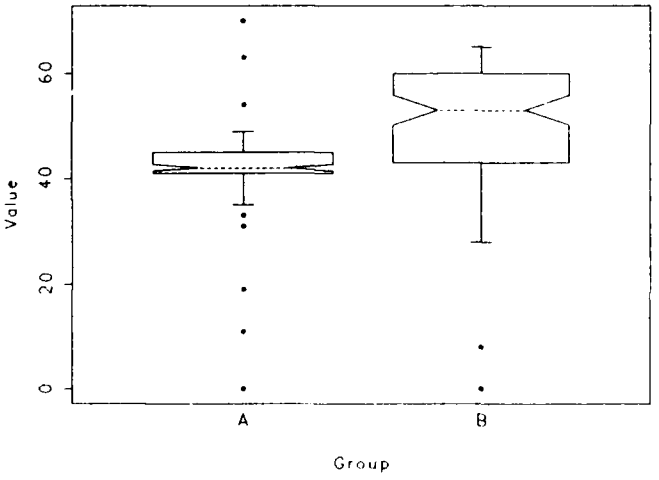
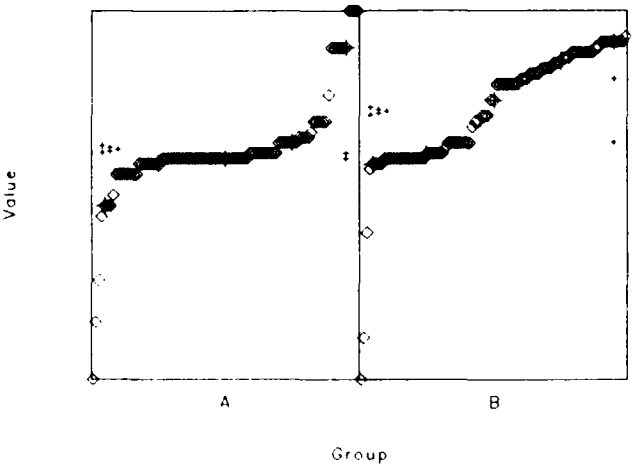


Figure 5B – Quantile Plots of 5A



Singular values of large matrices subject to Gaussian perturbation

Lorraine Denby and Colin Mallows

AT&T Bell Labs
Murray Hill, NJ 07974

Abstract

Extending the work of Wachter (1978, 1980) and many others, we study the configuration of the singular values (s.v.'s) of an a by b matrix of the form $X = M + \sigma Z$ where M is a constant matrix, and the elements of Z are i.i.d., standard Gaussian, in the limit as a and b increase in constant ratio. We put $N = a + b$ and suppose $a = \alpha N$, $b = \beta N$, with σ of order $1/\sqrt{N}$. Let the empirical distribution of the s.v.'s of X be G_N, and let the corresponding moment-generating-function (m.g.f) be $g_N(t)$. These are random quantities; their distributions depend only on σ and the empirical distribution F_N of the s.v.'s of M. We derive a differential equation that governs the evolution of $E(g_N)$ as σ increases. In the limit as $N \to \infty$ we can solve this equation and hence exhibit the limiting (non-random) g itself.

This study was motivated by some blood-pressure data collected by a new type of transducer. It suggests a novel way of adjusting large matrices to reduce the effect of additive contamination.

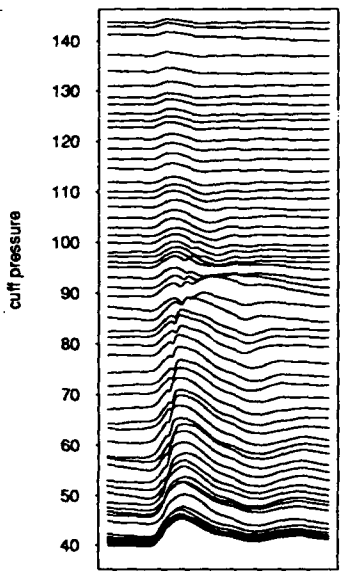


Figure 1

1. Introduction

In the standard technique for measuring blood pressure, a pressure cuff is applied to the upper arm, inflated to constrict the artery, and deflated while a technician listens (through a stethoscope) for the so-called Korotkoff signal. A novel form of transducer now allows the recording of a continuous trace of inaudible low-frequency auditory data, thus affording a first glimpse of the details of the process. Figure 1 shows such a record, segmented into individual heartbeats. Cuff pressure decreases down the figure.

An early attempt to analyze such data consisted of regarding Figure 1 as a display of the rows of a 70×373 matrix X. We performed a singular-value decomposition in the hope that an additive representation of the form

$$\hat{X}_{ij} = \sum_{k=1}^{m} c_k a_{ik} b_{kj}, \qquad (1)$$

with m not too large, would fit the data; here each row of the matrix $B = (b_{kj})$ is a prototypical component of a heartbeat trace, and the columns of $A = (a_{ik})$ show how these components enter and leave during the evolution of the traces. By convention, the rows of B and the columns of A are standardized to unit length; the magnitudes of the coefficients $\{c_k\}$ measure the importance of the components. It is a property of the singular-value decomposition of a matrix that the best (least-squares) representation of the form (1), using m terms, is obtained by taking the first m components of the singular-value decomposition

$$X = ACB^T$$

where $A^TA = B^TB = I_r$, C = diagonal, where r is the rank of X. It would be pleasant to find that a small value for m suffices to give a good fit to the data.

On performing the calculation, we found that a few of the singular values were quite large, while most were small. We were faced with the problem of deciding how many components to use.

2. An idealized problem.

As an idealization of this set-up, suppose we have observed an $a \times b$ matrix X with the structure

$$X = M + \sigma Z$$

where the elements of Z are independent standard Gaussian. How do the singular values of X depend on those of M? We formulate this as an asymptotic question. Suppose $a \le b$, and put $N = a + b$, $a = \alpha N$, $b = \beta N$. We study the asymptotic configuration of the singular values (s.v.'s) of an a x b matrix of the form $X = M + \frac{\sigma}{\sqrt{N}} Z$ where M is a constant matrix, and the elements of Z are i.i.d., standard Gaussian. Suppose X has singular values $x_1, \cdots, x_a$. It is convenient to work with a symmetrized form of the empirical distribution of these s.v.'s, namely

$$G_N^{(X)}(x) = \frac{1}{N}\left(\sum I(-x_i < x) + (b-a) I(x > 0) + \sum I(x_i < x)\right)$$

We also need the generating function (modified Stieltjes transform)

$$g_N^{(X)}(z) = \int_{-\infty}^{\infty} \frac{z}{1 - zx} dG_N(x)$$

$$= \sum_{k=0} z^{2k+1} X_{2k}$$

where X_{2k} is the $2k$-th moment of the (symmetrized) distribution $G_N^{(X)}$, so

$$X_{2k} = \frac{2}{N} \sum x_i^{2k}$$

Both $G_N^{(X)}$ and $g_N^{(X)}$ are random quantities; their distributions depend only on σ and the (symmetrized) empirical distribution $F_N^{(M)}$ of the s.v.'s of M. We define the moments M_{2k} and a generating function $f_N^{(M)}$ from $F_N^{(M)}$ in a similar fashion. Below we shall let $N \to \infty$, and shall assume that $F_N^{(M)}$ converges to a limiting distribution F that has a moment-generating function f (with moments μ_{2k}) that converges within some non-vanishing interval.

Wachter (1978) considered this problem, replacing the Gaussian assumption by one involving boundedness of moments; also he allowed the columns of Z to have different variances. However (in our notation) he assumed $M \to 0$ as $N \to \infty$, so that the effect of M was negligible in the limit. In the present work, the role of M is crucial. Our results seem to be new. We find, as did Wachter, that as $N \to \infty$ G_N converges to a non-random limit G (with generating function g, and moments γ_{2k}).

We derive a differential equation for $E(g_N^{(X)})$. We cannot solve this in general; however letting $N \to \infty$, we derive a formula for the limiting g as a function of f and σ. In principle this enables us to calculate the density corresponding to f once g is known; in practice (since N is finite) this is an ill-conditioned calculation and we need approximate methods.

3. A differential equation

Since we are assuming that Z is Gaussian, we can appeal to the fact that

$$M + \sigma Z = M + sZ_1 + tZ_2$$

where $\sigma^2 = s^2 + t^2$, and Z_1 and Z_2 are independent Gaussian. We can set up a differential equation for the expected generating function

$$\gamma_N(\sigma^2, z) = E(g_N^{(M+\sigma Z)}(z))$$

by retaining only the terms of order σ^2 in the expected moments. We find

$$\frac{\partial \gamma}{\partial \sigma^2} = -\gamma \frac{\partial \gamma}{\partial z} + (\alpha\beta - \frac{1}{4}) \frac{1}{z^3} \tag{2}$$

$$+ \frac{1}{4N}\left(\frac{\partial^2 \gamma}{\partial z^2} + \frac{1}{z}\frac{\partial \gamma}{\partial z} - \frac{\gamma}{z^2}\right)$$

and we need merely to solve this equation.

4. The case $N = \infty, \alpha = 1/2$.

From (2) we have

$$\frac{\partial \gamma}{\partial \sigma^2} = \frac{1}{2} \gamma \frac{\partial \gamma}{\partial x} \tag{3}$$

with the boundary condition

$$\gamma(0, x) = \gamma^{(M)}(x)$$

This relation provides a rapid way of computing the moments of G from those of F. The solution of (3) is

$$\gamma(\sigma^2, z) = \gamma^{(M+\sigma Z)}(z) = \gamma^{(M)}(y)$$

where

$$z = y + \sigma^2 \gamma^{(M)}(y)$$

When $M = 0$ we find

$$\gamma^{(0)}(z) = \gamma(0, z) = \frac{1}{2z}$$

$$\gamma^{(\sigma Z)}(z) = \gamma(\sigma^2, z) = \frac{1}{2y}$$

where

$$z = y + \sigma^2 \gamma^{(0)}(y) = y + \frac{\sigma^2}{2y}$$

so

$$\gamma(\sigma^2, z) = \frac{1}{2\sigma^2}\left(z - \sqrt{z^2 - 2\sigma^2}\right)$$

$$f^{(\sigma Z)}(x) = \frac{2}{\pi \sigma^2} \sqrt{2\sigma^2 - x^2} \qquad 0 \le x \le \sqrt{2\sigma^2}.$$

This is Wigner's "semicircle law", see Mehta (1967) and Figure 2.

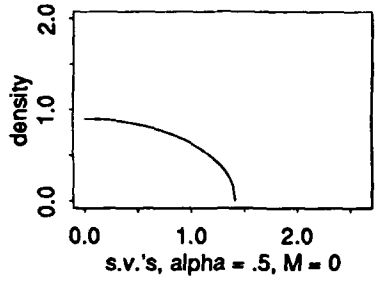


Figure 2

5. The case $N=\infty, \alpha \neq 1/2$.

Write $\delta=(\alpha-\beta)^2$. We find

$$z\gamma(\sigma^2,z)=y\gamma(0,y)+\sigma^2 Q \tag{4}$$

where

$$Q=\gamma^2(\sigma^2,z)-\frac{\delta}{4z^2}=\gamma^2(0,y)-\frac{\delta}{4y^2}$$

When $M=0$, take $\sigma=1$, and write γ for $\gamma(1,z)$. We find

$$z\gamma=\frac{1}{2}+\gamma^2-\frac{\delta}{4z^2}$$

whence

$$f(x)=\frac{1}{\pi\alpha x}\sqrt{(B^2-x^2)(x^2-A^2)} \tag{5}$$

where

$$A=\sqrt{\beta}-\sqrt{\alpha}, \quad B=\sqrt{\beta}+\sqrt{\alpha}.$$

See Figures 3,4 for the cases $\alpha=.3$, $\alpha=.1$.

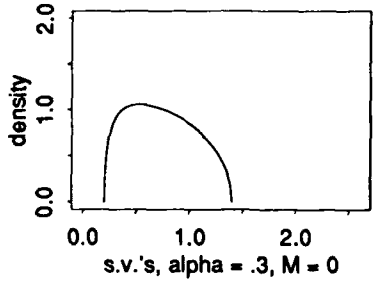


Figure 3

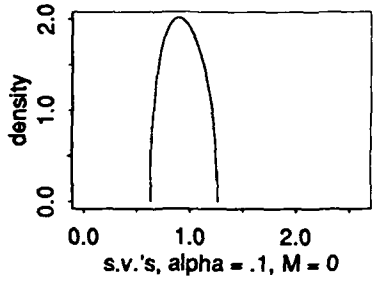


Figure 4

6. A special case.

Suppose $\alpha=1/2$, and that all the s.v.'s of M are equal to μ. Then $\mu_{2k}=2\mu^{2k}$, and

$$\gamma^{(M)}(x)=\frac{1}{2}\left(\frac{x}{1-x\mu}+\frac{x}{1+x\mu}\right)$$

Thus if $X=M+(\sigma/\sqrt{N})Z$ (remember $N=2a$) we have from (7)

$$\gamma^{(X)}(x)=\frac{y}{1-y^2\mu^2}$$

where

$$\frac{1}{x}=\frac{1}{y}+\frac{\sigma^2}{2}\frac{y}{1-y^2\mu^2}$$

This relation holds within the circle of convergence. To get F itself, we need to continue the definition outside this circle, taking care to use the correct branch. Then we apply the formula (see Wachter (1978))

$$g(\xi)=\frac{1}{\pi}\mathrm{Im}(\gamma(\frac{1}{\xi}))$$

In one case we can get an explicit result, namely when

$$\mu^2=\sigma^2/2$$

In this case

$$x=y(1-y^2\mu^2)$$

so that

$$\gamma^X(1/\xi)=y^2\xi$$

Thus we need only solve a cubic equation. Writing $y=(2/\mu\sqrt{3})\sin\theta$, we have $\sin 3\theta=3\sqrt{3}\,\mu/2\xi$. We get complex roots for $|\xi|<3\sqrt{3}\,\mu/2$. For $0<\xi<3\sqrt{3}\,\mu/2$ we put $\theta=\pi/6+i\psi$ and find

$$f^X(\xi)=\frac{1}{\pi}\frac{\xi}{\sqrt{3}\,\gamma^2}\sinh 2\psi \tag{6}$$

where

$$\cosh 3\psi=\frac{3\sqrt{3}\,\mu}{2\xi}$$

See Figure 5. Remember that this is the symmetrized distribution, so we need to multiply by 2 to get the limiting density of the s.v.'s of X.

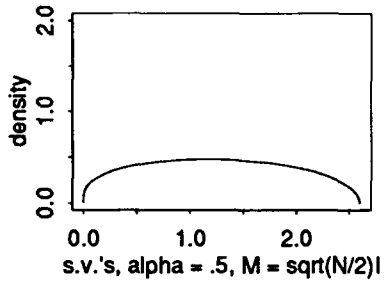


Figure 5

7. Statistical application.

Suppose we compute the s.v.'s of a large matrix, and observe that their empirical distribution is similar to (6) above. Then this supports the view that the matrix can be regarded as the sum of (i) a fixed matrix with all s.v.'s equal, and (ii) a matrix of independent random variables with equal variances. In more generality, if the s.v.'s of a large square matrix have an empirical distribution G, we would like to estimate an F such that the relation (4) is approximately satisfied. As yet we have no detailed suggestions as to how to do this.

For the 70×373 matrix that stimulated this investigation, we find that a q-q plot of the 70 realized singular values against quantiles of the distribution (5) (Figure 6) is very far from linear; the lowest 30 or so s.v.'s (Figure 7 shows 40) do conform roughly to this null prescription, with σ about 65. But this value for σ is much too large to be reasonable for these data; computing the the root-mean-square successive difference of the rows of the matrix, we get numbers averaging 23, with a maximum of 45. We conclude that for this approach to work, we will need an M with very few non-zero singular values. Evidently this approach is unsuited to these data.

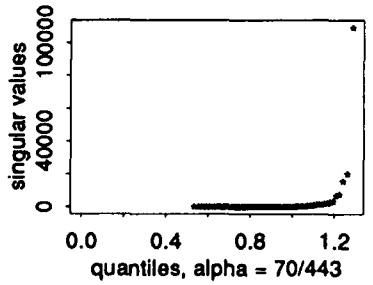


Figure 6

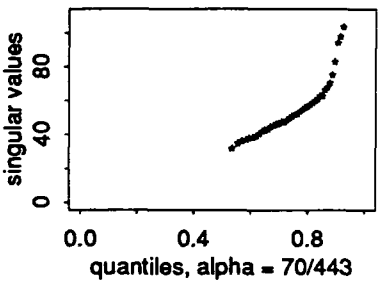


Figure 7

8. Final Comments

Clearly this work is incomplete. Among the things that need doing are:

(i) Extend the results to dispense with the assumptions of Gaussianity and identical distribution of the elements of Z.

(ii) Extend the results to dispense with the assumption that the moments of F are finite.

(iii) Develop an algorithm to find the density of G directly from the density of F.

(iv) Develop techniques to do (iii) approximately (in some appropriate sense) in the case N finite.

(v) Solve (2) in general.

(vi) Study the variability of $G_N^{(X)}$.

References

Mehta, M.L. (1967) *Random Matrices and the Theory of Energy Levels*. Academic Press, New York.

Wachter, K.W. (1978) The strong limits of random matrix spectra for sample matrices of independent elements. *Ann. Probability* **6**, 1-18.

Wachter, K.W. (1980) The limiting empirical measure of multiple discriminant ratios. *Ann. Statistics* **8**, 937-957.

Gaussian Windows: A Multivariate Exploratory Method

LOUIS A. JAECKEL*

Research Institute for Advanced Computer Science
NASA Ames Research Center
Moffett Field, CA 94035

Abstract

This paper presents a method for interactively exploring a large set of quantitative multivariate data, in order to estimate the shape of the underlying density function. It is assumed that the density function is more or less smooth. The local structure of the data in a given region may be examined by viewing the data through a *Gaussian window*, whose location and shape are chosen by the user. The method, which is applicable in any number of dimensions, can be used to find and describe simple structural features such as peaks, valleys, and saddle points in the density function, and also extended structures such as ridges and analogous structures in higher dimensions. A Gaussian window is defined by giving each data point a weight based on a multivariate Gaussian function. The weighted sample mean and sample covariance matrix are then computed, using the weights attached to the data points. These quantities are used to compute an estimate of the shape of the density function in the window region. The local structure of the data is described by a method similar to the method of principal components. Thus we can apply our geometrical intuition to the structural features we find in the data, in any number of dimensions. By taking many such local views of the data, we can form an idea of the structure of the data set. Since the computations involved are relatively simple, the method can be implemented on a small computer.

1 Introduction

Suppose that we are given a large set of quantitative multivariate data, say, N data points $\mathbf{x}_i$ in a p-dimensional space, and that we want to explore the structure of the data. That is, we want to find the shape of the underlying density function, by looking for concentrations of data points. We will assume that the density function is more or less smooth, but we will not make any more specific assumptions about its structure. To explore the data, we need a way to look at the local structure of the data in a limited region. So we will examine the data in a given region by viewing the data through a *Gaussian window*, whose location and shape are chosen by the user. We will describe the local structure of the data by a method similar to the method of principal components. By doing this we will be able to find and describe simple structural features in the data in any number of dimensions.

Some examples of the kinds of structures that we can find and describe are the following: A peak, or relative maximum, in the density function, which would appear as a cluster of data points; a valley, or relative minimum; and a saddle point, where the density function would be concave upward in some directions, and downward in others. We can also find extended structures such as a "ridge", or "bar", in the data. A "ridge" is an essentially one-dimensional structure, or concentration of data points, consisting of data points lying near a "center line" but scattered about it in all directions. Only a part of such an extended structure would be visible in a single window. In a case like this we will be able to tell that we are looking at a structure that extends beyond the window. We can then follow along it and map out its extent and shape. Similarly, we might find an essentially k-dimensional structure in a p-dimensional space, for any $k < p$.

By taking many local views of the data, that is, by exploring the data interactively, we can build up an idea of the structure of the data set. With some practice, we can apply our geometrical intuition to the features we find in the data, in any number of dimensions. Since the computations are relatively simple, the method can be implemented on a small computer.

The approach here is different from that in the many graphical methods that involve projecting the data onto a space of lower dimension. See for example Chambers et al. (1983) and Cleveland and McGill (1988). However, such graphical methods can be used in conjunction with the method described here.

The ideas outlined in this paper are treated more thoroughly in Jaeckel (1990).

*Work reported herein was supported in part by Cooperative Agreements NCC 2-408 and NCC 2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA).

2 The Gaussian window

To focus on a limited region in the space, we use a window. A *Gaussian window* is defined by choosing a center point α and a non-negative definite symmetric matrix V to describe its size and shape. Let

$$w(x) = e^{-\frac{1}{2}(x-\alpha)'V(x-\alpha)},$$

where x is a p-vector and "prime" means "transpose". The matrix V is analogous to the inverse of a covariance matrix. Each data point x_i is given the weight $w_i = w(x_i)$. Note that $w(\alpha) = 1$, that $w(x) \le 1$ for all x, and that w(x) decreases as x moves away from α. Thus we have defined a window with "fuzzy" boundaries. The function w(x) may be thought of as the relative transparency of the window at x.

We then compute the *weighted sample mean vector*,

$$\bar{x}_w = \frac{1}{\Sigma w_i} \Sigma w_i x_i ,$$

and the *weighted sample covariance matrix*,

$$S_w = \frac{1}{\Sigma w_i} \Sigma w_i (x_i - \bar{x}_w)(x_i - \bar{x}_w)' .$$

We also compute $(1/N)\Sigma w_i$.

These quantities are the simplest things to compute, especially in a high-dimensional space. They describe the overall shape of the weighted data in the "window region" (the region vaguely defined as the region where w(x) is "not small"). The estimated shape of the density function in the window region will be based on these quantities. Note that these quantities are overall statistics; any "fine structure" in the region is smeared out. To look for finer details, we would use smaller windows.

3 Example: a cluster

Suppose that in the region of a window, the density function has approximately a multivariate Gaussian shape:

$$f(x) = C \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)},$$

where μ, Σ, and C are all unknown parameters. That is, we have a single peak (or cluster of data points) in the window region. The vector μ is the center point of this part of the density. The symmetric matrix Σ is its covariance matrix. The constant C represents the "probability mass" of this part of the entire probability distribution.

The *windowed density function*, the effective density function of the data as viewed through the window, is w(x)f(x). That is, if we assign weight $w_i = w(x_i)$ to each data point x_i, and if we do computations with the weighted x_i, the results will be as if we were working with a sample from w(x)f(x).

Assume for simplicity that α, the window center, is **0**.

Let $B = \Sigma^{-1}$. It will be more convenient to work with B.

Let $A = B + V$. Then, by doing some algebra, we find that the windowed density function is

$$w(x)f(x) = K \frac{|A|^{1/2}}{(2\pi)^{p/2}} e^{-\frac{1}{2}(x - A^{-1}B\mu)'A(x - A^{-1}B\mu)} .$$

This is a multivariate Gaussian function with "windowed mean" $A^{-1}B\mu$ and "windowed covariance matrix" A^{-1}. It follows that the weighted sample mean $\bar{x}_w$ is an estimate of $A^{-1}B\mu$, and the weighted sample covariance matrix S_w is an estimate of A^{-1}. The constant K above is the integral of w(x)f(x) over the entire space. We will estimate it by $(1/N)\Sigma w_i$, the average of the weights.

We now "degauss" the view of the data as seen through the Gaussian window; that is, we remove the effect of the weights on the shape of the data in the window region. Since S_w is an estimate of A^{-1}, we can estimate A by S_w^{-1}, and we have

$$S_w^{-1} = \hat{A} = \hat{\Sigma} + V .$$

So we can estimate B by

$$\hat{B} = S_w^{-1} - V .$$

We can then estimate Σ by

$$\hat{\Sigma} = \hat{B}^{-1} = (S_w^{-1} - V)^{-1} ,$$

assuming that $S_w^{-1} - V$ is positive definite.

Since $\bar{x}_w$ is an estimate of $A^{-1}B\mu$, we can estimate μ by

$$\hat{\mu} = \hat{B}^{-1} S_w^{-1} \bar{x}_w .$$

And since $(1/N)\Sigma w_i$ is an estimate of K, we can also estimate the constant C. These estimated parameters give us an estimate of the shape of the density function in the window region. Note that all of the computations are simple matrix operations.

If we find a cluster in a window, we can describe its shape using the method of principal components. See Morrison (1990). To do this we find the eigenvalues and corresponding eigenvectors of $\hat{\Sigma}$. The estimated shape of the cluster is a p-dimensional ellipsoidal shape centered at $\hat{\mu}$. The principal axes of the ellipsoid are parallel to the eigenvectors. The estimated density function can be expressed as a product of p univariate Gaussian (normal) densities, each lying along a principal axis. The standard deviation of each of these densities is the square root of the corresponding eigenvalue (all of which are positive in this case). Thus we have a way of thinking about the shape of the cluster in any number of dimensions.

Note that we could do this analysis based on the matrix $\hat{B}$, which is the inverse of $\hat{\Sigma}$. These two matrices have the same eigenvectors, and the eigenvalues of $\hat{B}$ are the reciprocals of those of $\hat{\Sigma}$. It follows that a large positive eigenvalue of $\hat{B}$ indicates that the data points are tightly concentrated along the corresponding direction, while an eigenvalue near 0 indicates a structure that may extend beyond the window region. When we deal with more general structures, we will analyze their shape by looking at the eigenvalues and eigenvectors of $\hat{B}$.

The analysis above also applies if the shape of the density function in the window region is a valley or a saddle point. In these cases all or some of the eigenvalues of $\hat{B}$ will be negative. A negative eigenvalue indicates that, in the window region, the density function is concave upward along the direction of the corresponding eigenvector.

4 The general case

We now give a more general formulation which will include the examples above, and also extended structures such as a "ridge". We will assume that the density function in the window region can be approximated by

$$f(\mathbf{x}) = H\, e^{-\frac{1}{2}\mathbf{x}'B\mathbf{x} + \mathbf{r}'\mathbf{x}}.$$

The exponent is a general polynomial of degree two in the coordinates of the vector $\mathbf{x}$. (Any constant term is absorbed in H.) The constant H is the density at the window center (assumed to be at $\mathbf{0}$). The symmetric matrix B may or may not be positive definite, and it may or may not be non-singular. If B is singular, there is no center point μ for the function.

As before, the windowed density function $w(\mathbf{x})f(\mathbf{x})$ is a multivariate Gaussian function. We therefore compute $\bar{\mathbf{x}}_w$, S_w, and $(1/N)\Sigma w_i$ as before, and we estimate the parameters B, $\mathbf{r}$, and H based on these quantities. See Jaeckel (1990). Since in the general case $\hat{B}$ might be nearly singular, we will work directly with $\hat{B}$ instead of inverting it. We then find the eigenvalues and eigenvectors of $\hat{B}$, and we use these quantities to describe the shape of the estimated density function in the window region. The method is analogous to the method of principal components. The interpretation of the eigenvalues of $\hat{B}$ is the same as in the previous section. As in principal components analysis, we can express the estimated density function as a product of p functions of one variable each.

We can now handle the case of an extended structural feature, such as a "ridge" of data points, that passes through a window and extends beyond it. In this case $\hat{B}$ will have some eigenvalues very near 0; these eigenvalues tell us that the structure extends beyond the window. Since $\hat{B}$ is the estimated inverse covariance matrix, an eigenvalue near 0 indicates that the data in the window region appear to have an essentially "infinite" variance in the direction of the corresponding eigenvector. In the case of a ridge, which is an essentially one-dimensional concentration of points, $\hat{B}$ will have one eigenvalue very near 0, and the corresponding eigenvector will be parallel to the "center line", or crest, of the ridge.

Since a structure like this does not have a center point, as a cluster does, we will not try to estimate a center point here. Instead, we will estimate the location of the center line of the ridge. See Jaeckel (1990). We can also use the $p-1$ remaining eigenvalues and eigenvectors to estimate the shape of the cross-section of the ridge. In a p-dimensional space, a ridge would have a $(p-1)$-dimensional cross-section orthogonal to the center line.

If we find a structure like this, we can then move the window center to the nearest point on the center line and try another window. Then we can follow along the ridge by moving the window center along the estimated center line. By continuing in this way we can map out the extent and shape of the ridge. An essentially k-dimensional structure, or concentration of data points, can be treated in a similar way.

Since the method is interactive, it is flexible and open-ended. It can be used (in principle) in any number of dimensions. Few assumptions are made about the data. We can search for structural features by trying many different windows, and we can describe the features we find. Then we can put together what we have found into an overall description of the data. The method can be used in conjunction with other methods, such as graphical methods and automatic clustering algorithms. Note that with this method we can find structural features other than clusters. Since the computations are relatively simple, the method can easily be implemented on a small computer. Any standard algorithms for inverting a matrix and for finding the eigenvalues and eigenvectors of a symmetric matrix can be used.

Most importantly, we can apply our geometrical intuition to the features we find in the data, so that we can think about and describe the structure of a set of data in any number of dimensions.

References

Chambers, J., Cleveland, W., Kleiner, B., and Tukey, P. (1983), *Graphical Methods for Data Analysis*, Pacific Grove, CA: Wadsworth & Brooks/Cole.

Cleveland, W., and McGill, M. (eds.) (1988), *Dynamic Graphics for Statistics*, Pacific Grove, CA: Wadsworth & Brooks/Cole.

Jaeckel, L. A. (1990), Gaussian windows: a tool for exploring multivariate data. Technical Report 90.41, RIACS, Moffett Field, CA (Submitted for publication in *J. Amer. Statist. Assoc.*).

Morrison, D. F. (1990), *Multivariate Statistical Methods* (3rd ed.), New York: McGraw-Hill.

ANALYZING OF HIGH DIMENSIONAL 0–1 DATA SET, BOOLEAN FACTOR ANALYSIS

Lidia Rejtő

Department of Mathematical Sciences, University of Delaware
Newark, Delaware 19716

Definition of the Problem

Factor analysis is a frequently used statistical tool for representing a usually large number of observable variables with a smaller set of latent factors. In classical factor analysis, the observable variables are expressed as linear combinations of the factors. During this procedure neither the factors nor the scores are binary. Boolean factor analysis is a procedure for the representation of binary variables in terms of Boolean combinations of binary factors.

Suppose that X is a d dimensional random variable with binary coordinates and

$$X = \mathbf{A} \otimes Y,$$

where $\mathbf{A}$ is a fixed $(d \times l)$ matrix with binary coordinates and Y is an l dimensional random vector with binary coordinates with $l < d$. The $\otimes$ notation means that we are using Boolean operations which are reflected in the following tables:

$\oplus$	0	1
0	0	1
1	1	1

$\otimes$	0	1
0	0	0
1	0	1

In our model, $\mathbf{A}$ is unknown, Y is unknown and $l < d$ means that the data comes from a smaller dimensional space through the fixed matrix $\mathbf{A}$.

Furthermore it is supposed that there is a random error in the observations; instead of X we observe $\tilde{X} = X + \epsilon$ where ϵ is a d dimensional random vector with independent coordinates $\epsilon_1; \ldots; \epsilon_d$. The conditional distribution of ϵ_i given X_i is

ϵ_i	$P(\epsilon_i \mid X_i = 0)$	$P(\epsilon_i \mid X_i = 1)$
-1	0	p_1
0	$1 - p_0$	$1 - p_1$
1	p_0	0

The error probability depends on the actual value of X_i of the i–th coordinate of X. It is supposed that the error probabilities p_0 and p_1 are small.

The aim of the Boolean factor analysis is to recover $\mathbf{A}$ and Y with the help of the given data set.

The idea of Boolean factor analysis at first appeared at the BMDP package (see Dixon [1]) although their model is slightly different. The algorithm developed in [2] is entirely different in one step.

Finding the Boolean Scores and Loading Matrices

Suppose the data set is given in a matrix form; $\mathbf{D} = (d_{ij})$ is a $(d \times n)$ binary data matrix. The algorithm seeks a loading matrix $\mathbf{A}$ and a scores matrix $\mathbf{S}$ such that $\mathbf{B} = \mathbf{A} \otimes \mathbf{S}$ is "close" to $\mathbf{D}$, where $\mathbf{B}$ is called estimator or predictor of $\mathbf{D}$ Now we define a criteria for closeness.

Definitions: Positive discrepancy means that $d_{ij} = 0$ and $b_{ij} = 1$, i.e., the i–th variable of the j–th data is 0 which is predicted as 1. Negative discrepancy means that $d_{ij} = 1$ and $b_{ij} = 0$ i.e., that the i–th variable of the j–th data is 1 which is predicted as 0.

Suppose that the
cost of positive discrepancy is $c_p \geq 0$
and the
cost of negative discrepancy is $c_n \geq 0$.
The task is to find loading and score matrices for fixed unknown l which minimizes the overall cost function:

$$C = \|\mathbf{D} - \mathbf{A} \otimes \mathbf{S}\| =$$

$$c_p \sum_{i=1}^{d} \sum_{j=1}^{n} I(d_{ij} = 0,\ b_{ij} = 1) +$$

$$c_n \sum_{i=1}^{d} \sum_{j=1}^{n} I(d_{ij} = 1,\ b_{ij} = 0).$$

A two step algorithm developed to solve the above problem is in paper [2] which contains the details. That version of the Boolean factor analysis program was written for the DIscrete STatistical ANalysis (DISTAN) package sponsored by the Social Science Information Center of the Hungarian Academy of Sciences. This program has another version with more features. A brief description of the algorithm now follows.

Step One searches for a new vector of the loading matrix. That search is based upon the dependence between the variables. The method developed for this step is different then the one used in BMDP. This step is very important because at the beginning it is possible to incur only a small cost if the loading matrix is appropriately chosen. To explain it in more detail, in the first step we must give a d dimensional $0-1$ vector as the loading matrix and one dimensional scores for each case. Suppose that both costs $c_p = c_n = 1$. Considering the nature of the Boolean operations, we can initialize the algorithm with the loading vector having all its components equal to 1. Then we define the scores for each case as 1 if the case has more 1 then 0 or as 0 otherwise. The cost for each case, is the number of 0-s if the case has more 1 or the number of 1-s otherwise. Are there any loading vectors with a smaller cost? The answer is yes. The initial loading vector defined with the help of the pairwise dependence of the variables can be different from the one with all its coordinates equal to 1 and data analysis shows this has a lower cost. A data set of 796 patients was analyzed. The rigidity and strengthness of the muscles were measured in different parts of the body; there were 89 variables. Table 1 shows part of the output of the Boolean factor analysis program. Using a loading vector all of whose coordinates equal 1 produces a cost of 17572; if we use the random nature of the data set, with the help of the dependence structure of the variables the initial cost is lowered to 5808.

Step Two consists of defining and refining the scores and loadings. This is the so-called Boolean regression step similar to the one used in the BMDP 8M program. In this step for a given $(d \times k)$ loading matrix $\mathbf{A}$ and for a given case x_i the algorithm chooses a score which minimizes the cost of misprediction for that case examining all possible 2^k scores. Then the loading matrix $\mathbf{A}$ is modified in a similar fashion for the given scores matrix.

Example

The data for the example come from a study of muscles of 796 subjects with muscle disorders. The flexibility and strength of different muscles of the body were measured on a scale from 0 to 6. A 0 value means normal muscle function. A 6 means completely rigid and weak muscle. A value between 1–5 means different levels of flexibility or strength. 45 different muscles were tested. For the purpose of a Boolean factor analysis the data was coded by 0 and 1 in the following way: if the value of the variable was between 1–6, referring to abnormal muscle function, we code 1. In case of one muscle the value of both variable flexibility and strength was the same; either 0 or 6. This way we analyzed 89 variables for 796 subjects.

Table 1 shows an output of the Boolean factor analysis of the described data set. Using 9 factors the cost is only 1554. Because $c_p = c_n = 1$ it means that $\mathbf{B}$, the estimator of $\mathbf{D}$, of the coded data set, is different from $\mathbf{D}$ in 1554 places out of 70844. Thus the prediction error is only 2%. Table 2 shows the nonzero coordinates of the column vectors of the loading matrix.

The example shows that Boolean factor analysis can be applied successfully not only binary data set. The final prediction error is very impressive considering the fact that the new codes are producing larger error.

References

[1] Dixon, W.J. 1988: *BMDP Statistical Software Manual*, Vol. 2 789–800. University of California Press

[2] Rejtő, L. (1990) Boolean Factor Analysis. appears in *DISTAN for discrete statistical data analysis.* (ed. Rudas, T. Social Science Informatics Center of the Hungarian Academy of Sciences, Budapest, Hungary).

```
MAXIMAL DISCREPANCY=        70844
MAXIMAL COST =       70844

 ##  1 MODELLVECTORS  ##
COST = 5808.0000
DISCREPANCY=     5808 PDIS=     5090  NDIS=      718
PREDICTION ERROR=  0.08198   COST ERROR =  0.08198

 ##  2 MODELLVECTORS  ##
COST = 2920.0000
DISCREPANCY=     2920 PDIS=     1930  NDIS=      990
PREDICTION ERROR=  0.04122   COST ERROR =  0.04122

 ##  3 MODELLVECTORS  ##
COST = 2768.0000
DISCREPANCY=     2768 PDIS=     1780  NDIS=      988
PREDICTION ERROR=  0.03907   COST ERROR =  0.03907

 ##  4 MODELLVECTORS  ##
COST = 2708.0000
DISCREPANCY=     2708 PDIS=     1751  NDIS=      957
PREDICTION ERROR=  0.03822   COST ERROR =  0.03822

 ##  5 MODELLVECTORS  ##
COST = 2587.0000
DISCREPANCY=     2587 PDIS=     1664  NDIS=      923
PREDICTION ERROR=  0.03652   COST ERROR =  0.03652

 ##  6 MODELLVECTORS  ##
COST = 2096.0000
DISCREPANCY=     2096 PDIS=     1052  NDIS=     1044
PREDICTION ERROR=  0.02959   COST ERROR =  0.02959

 ##  7 MODELLVECTORS  ##
COST = 1814.0000
DISCREPANCY=     1814 PDIS=      861  NDIS=      953
PREDICTION ERROR=  0.02561   COST ERROR =  0.02561

 ##  8 MODELLVECTORS  ##
COST = 1667.0000
DISCREPANCY=     1667 PDIS=      813  NDIS=      854
PREDICTION ERROR=  0.02353   COST ERROR =  0.02353

 ##  9 MODELLVECTORS  ##
COST = 1554.0000
DISCREPANCY=     1554 PDIS=      783  NDIS=      771
PREDICTION ERROR=  0.02194   COST ERROR =  0.02194
```

Table 1

Number of Factors : 9

LOADING MATRIX

-** 1 ***
16 17 18 19 20 21 22 23 24 25 26 27 28 29 38 39 40 41 42 43
44 45

-** 2 ***
46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 76 79 81 82 84

-** 3 ***
1 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88

-** 4 ***
46 49 51 52 58 59 60 63 65 66 75 77 78 80 83 85 86 87 88 89

-** 5 ***
1 89

-** 6 ***
30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45

-** 7 ***
1 2 3 4 6 7 8 9 10 11 12 13 16 17 18 20 21 22 23 24
27 32 35 37 40 43 45

-** 8 ***
6 13 20 27 30 32 38

-** 9 ***
5 8 13 14 15 19 22 27 28 29 39

Table 2

General Similarity Measures of Location Models

Ruey-Pyng Lu
Department of Statistics
North Dakota State University
Fargo, ND 58105

Abstract

The location models, which can be used in discriminant problems when the data contain both categorical and continuous variables, requires separate continuous variables means to be fitted for each possible pattern of categorical responses. Several forms of similarity measure are reviewed. The problem of estimating similarity when the continuous variables of location models are multivariate normal distributions with equal covariance matrices across the discrete states has previously been studied. In this work, the assumption of equal covariance matrices is relaxed. The explicit form of general similarity measure between two location models is derived assuming general multivariate normal distributions. Estimation of parameters in this similarity measure is discussed.

1 Measures of distance and similarity measures

Consider two populations π_1 and π_2 and a vector-valued continuous random variable **X** defined over a space **R** such that $F_1(x)$ and $F_2(x)$ are the distribution functions of **X** in π_1 and π_2 while $f_1(x)$ and $f_2(x)$ are the corresponding density functions with respect to a suitable measure. For a discrete random variable **X**, $f_1(x)$ and $f_2(x)$ will be treated as the corresponding probability mass functions.

The following distance measures or similarity measures have been extensively studied:

(a) Hellinger distance (1907):

$$\rho_p(\pi_1,\pi_2) = \{ \int_{-\infty}^{\infty} | [f_1(x)]^{1/p} - [f_2(x)]^{1/p} |^p dx \}^{1/p}$$

(b) Bhattacharyya distance measure (1946):

$$\theta(\pi_1,\pi_2) = \cos^{-1} \rho(\pi_1,\pi_2)$$

where $$\rho(\pi_1,\pi_2) = \int_{-\infty}^{\infty} [f_1(x) f_2(x)]^{1/2} dx$$

(c) Jeffreys divergence measure (1946):

$$J(\pi_1,\pi_2) = \int_{-\infty}^{\infty} (f_2(x) - f_1(x)) \log [\frac{f_2(x)}{f_1(x)}] dx$$

(d) Kullback & Leibler's information measure (1951):

$$\Delta_1(\pi_1,\pi_2) = \int_{-\infty}^{\infty} f_1(x) \log [\frac{f_1(x)}{f_2(x)}] dx$$

$$\Delta_2(\pi_1,\pi_2) = \int_{-\infty}^{\infty} f_2(x) \log [\frac{f_2(x)}{f_1(x)}] dx$$

(e) Chernoff measure (1952):

$$\rho(\pi_1,\pi_2) = \int_{-\infty}^{\infty} [f_1(x)]^{\alpha} [f_2(x)]^{1-\alpha} dx , \qquad 0<\alpha<1$$

(f) Matusita's distance (1955):

$$\| F_1, F_2 \|_r = | \int_{-\infty}^{\infty} [(f_1(x))^{1/r} - (f_2(x))^{1/r}]^r dx |^{1/r}$$

This is essentially the same as Hellinger distance. If the affinity between F_1 and F_2 is

$$\rho(F_1,F_2) = \int_{-\infty}^{\infty} [f_1(x) f_2(x)]^{1/2} dx$$

then $$\| F_1, F_2 \|_2^2 = 2 (1 - \rho(F_1,F_2))$$

(g) Morisita's similarity measure (1959):

$$\lambda = \frac{2 \int_{-\infty}^{\infty} f_1(x) f_2(x) dx}{\int_{-\infty}^{\infty} f_1^2(x) dx + \int_{-\infty}^{\infty} f_2^2(x) dx}$$

(h) MacArthur-Levins similarity measure (1967):

$$\alpha_{ij} = \frac{\int_{-\infty}^{\infty} f_i(x) f_j(x) dx}{\int_{-\infty}^{\infty} f_i^2(x) dx} \qquad \text{for } i,j = 1,2,\ i \neq j .$$

(i) Sibson's information radius (1969):

$$\Delta(\pi_1,\pi_2) = \frac{1}{2}\int_{-\infty}^{\infty}\{f_1(x)\log[\frac{f_1(x)}{f(x)}] + f_2(x)\log[\frac{f_2(x)}{f(x)}]\}dx$$

where $f(x) = \frac{1}{2}(f_1(x) + f_2(x))$.

(j) Pianka's measure of overlap (1974):

$$\alpha = \sqrt{\alpha_{ij}\,\alpha_{ji}} \qquad \text{for } i,j=1,2,\ i\neq j.$$

$$= \frac{\int_{-\infty}^{\infty} f_i(x)\,f_j(x)\,dx}{[\int_{-\infty}^{\infty} f_i^2(x)\,dx]^{1/2}\,[\int_{-\infty}^{\infty} f_j^2(x)\,dx]^{1/2}}$$

(k) Good and Smith (1987) General measures of similarity:

$$I(r,s) = \int_{-\infty}^{\infty}[f_1(x)]^r\,[f_2(x)]^s\,dx$$

and two alternatives: $J(r,s) = 2\,\frac{I(r,s)}{I(2r,0) + I(0,2s)}$

and $G(r,s) = \frac{I(r,s)}{\sqrt{I(2r,0)I(0,2s)}}$

The parameters r and s are weighting parameters and are usually given the values $\frac{1}{2}$ or 1. Note: Some of the above measures will be special cases of these general similarity measures. For example, Bhattacharyya distance measure $\theta(\pi_1,\pi_2) = \cos^{-1} I(\frac{1}{2},\frac{1}{2})$, Chernoff measure is $\rho(\pi_1,\pi_2) = I(\alpha, 1-\alpha)$, Matusita's affinity measure is $\rho(F_1,F_2) = I(\frac{1}{2},\frac{1}{2})$, Morisita's similarity measure $\lambda = J(1,1)$, MacArthur-Levins similarity measure is $\frac{I(1,1)}{I(2,0)}$ or $\frac{I(1,1)}{I(0,2)}$, and Pianka's measure of overlap is $\alpha = G(1,1)$.

The above measures can be applied to the populations with discrete distributions and probability mas functions. In this case, summation over the possible states will be used instead of integration.

2 Location Models

Suppose that p continuous or quantitative variables $\mathbf{Y}^T = (Y_1, ..., Y_p)$ and q discrete or qualitative variables $\mathbf{X}^T = (X_1, ..., X_q)$ are measured on each individual, and that individuals are drawn from 2 populations π_1, π_2. The location model was introduced by Olkin and Tate (1961) to cope with the mixed variables, and this model has subsequently be applied to the two-sample case for tests of hypotheses, for discriminant analysis, for clustering, for classification, and for medical diagnostics.

The q discrete variables (may be binary or categorical) are assumed to define a multinomial vector **Z** containing d possible states, eg: for b binary variables and k three-state categorical variables, $d = 2^b3^k$.

Thus each distinct pattern of X define a multinomial cell uniquely. $\mathbf{X}^T = (X_1, ..., X_q)$ can be replaced by a random vector $\mathbf{Z}^T = (Z_1, ..., Z_d)$ and each Z_j takes the value one for a particular state of the original X's and zero elsewhere. The probability of observing state m in population p_i is assumed to be p_{im} (i=1,2; m=1,...,d). Then conditionally on Z falling in state m, the p continuous variables Y are assumed to follow a multivariate normal distribution with mean $\mu_i^{(m)}$ and dispersion matrix $\Sigma_i^{(m)}$ in population π_i (i=1,2; m=1,...,d). The only assumption embodied in this model is normality, and this is imposed in most parametric techniques.

The location model can be defined as

$$f(z,P_i \mid \pi_i) = \prod_{m=1}^{d} p_{im}^{z_{im}} \qquad \text{where } P_i^T = (p_{i1}, p_{i2},...,p_{id})$$

$$p_{im} = E(z_m \mid \pi_i), \qquad \sum_{m=1}^{d} z_m = 1, \qquad \sum_{m=1}^{d} p_{im} = 1$$

and $f_i^{(m)}(y) = f(y \mid \pi_i, z_m = 1, z_k = 0, m \neq k = 1,2,...,d)$

$$\sim N(\mu_i^{(m)}, \Sigma_i^{(m)})$$

The proposed model admits the following special cases of interest:

L_1: the conditional dispersion matrix is constant for all state in each population, that is $\Sigma_i^{(m)} = \Sigma_i$ (i=1,2; m=1,...,d): Homogeneous variance-covariance matrices across states within population.

L_2: the conditional dispersion matrix is constant for all state in each population, that is $\Sigma_i^{(m)} = \Sigma^{(m)}$ (i=1,2; m=1,...,d): Homogeneous variance-covariance matrices between populations with respect to states.

L_3: the conditional dispersion matrix is constant for all state in each population, that is $\Sigma_i^{(m)} = \Sigma$ (i=1,2; m=1,...,d). Homogeneous variance-covariance matrices across states and populations.

3 General similarity measures with mixed variables: Population case

Let us derive the general similarity measures under the most relaxed conditions between two multivariate normal populations - different means and different dispersion matrices.

$$I(r,s) = \frac{\exp[-\frac{rs}{2}(\mu_1-\mu_2)^T(s\Sigma_1+r\Sigma_2)^{-1}(\mu_1-\mu_2)]}{|\Sigma_1|^{(r-1)/2}|\Sigma_2|^{(s-1)/2}|s\Sigma_1+r\Sigma_2|^{1/2}(2p)^{p(r+s-1)/2}}$$

$$I(2r,0) = |2\pi\Sigma_1|^{(1-2r)/2}(2r)^{-p/2}$$

and $$I(0,2s) = |2\pi\Sigma_2|^{(1-2s)/2}(2s)^{-p/2}$$

$$J(r,r) = \frac{2\exp[(-r/2)(\mu_1-\mu_2)^T(r\Sigma_1+r\Sigma_2)^{-1}(\mu_1-\mu_2)]}{|\Sigma_1\Sigma_2|^{(r-1)/2}|\frac{1}{2}(\Sigma_1+\Sigma_2)|^{1/2}(|\Sigma_1|^{(1-2r)/2}+|\Sigma_2|^{(1-2r)/2})}$$

$$G(r,s) = \frac{\exp[-\frac{rs}{2}(\mu_1-\mu_2)^T(s\Sigma_1+r\Sigma_2)^{-1}(\mu_1-\mu_2)]}{(4rs)|\Sigma_1\Sigma_2|^{-1/4}|s\Sigma_1+r\Sigma_2|^{1/2}}$$

$$G(r,r) = \frac{\exp[-\frac{r}{2}(\mu_1-\mu_2)^T(\Sigma_1+\Sigma_2)^{-1}(\mu_1-\mu_2)]}{|\Sigma_1\Sigma_2|^{-1/4}|\frac{1}{2}(\Sigma_1+\Sigma_2)|^{1/2}}$$

For $r = s = \frac{1}{2}$, and when $\Sigma_1 = \Sigma_2 = \Sigma$,

$\rho = I(\frac{1}{2},\frac{1}{2}) = \exp[-\frac{1}{8}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2)]$ and

$\lambda = J(1,1) = \exp[-\frac{1}{4}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2)]$

$= \alpha_{12} = \alpha_{21} = \frac{I(1,1)}{I(2,0)} = \frac{I(1,1)}{I(0,2)} = G(1,1)$.

These are the exponential forms of certain functions of Mahalabonis generalized distance $(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2)$.

Krzanowski (1983) derived the following for $\Sigma_1 \neq \Sigma_2$,

$$\rho = 2|\Sigma_1|^{1/4}|\Sigma_2|^{-1/4}|I+\Sigma_1\Sigma_2^{-1}|^{-1/2} \cdot \exp[-\frac{1}{4}\sum_{j=1}^{p}\{\frac{(v_{1j}-v_{2j})^2}{(1+\lambda_j)}\}]$$

Here λ_j are the eigenvalues of $\Sigma_2\Sigma_1^{-1}$ and v_{1j}, v_{2j} $(j=1,...,p)$ are the coordinates of the population means in the transformed space.

Lu, Smith and Good (1989) derived

$$\rho = \frac{|\Sigma_1\Sigma_2|^{-1/4}}{|\frac{1}{2}(\Sigma_1+\Sigma_2)|^{1/2}}\exp[-\frac{1}{4}((\mu_1-\mu_2)^T(\Sigma_1+\Sigma_2)^{-1}(\mu_1-\mu_2)]$$

Now consider the mixed variable case:

The joint density of state m of the discrete variables and values of $y_1,...,y_p$ for the continuous variables is given by the product of the conditional and marginal densities as p_{im} $f_i^{(m)}(y)$.

$$I_L(r,s) = \sum_{m=1}^{d}\int_{-\infty}^{\infty}\{p_{1m}f_1^{(m)}(y)\}^r\{p_{2m}f_2^{(m)}(y)\}^s dy$$

$$= \sum_{m=1}^{d}[(p_{1m}^r p_{2m}^s)\int_{-\infty}^{\infty}\{f_1^{(m)}(y)\}^r\{f_2^{(m)}(y)\}^s dy]$$

$$= \sum_{m=1}^{d}\{p_{1m}^r p_{2m}^s I_m(r,s)\}$$

where $I_m(r,s)$ is the general measure of similarity $I(r,s)$ between $N(\mu_i^{(m)}, \Sigma_i^{(m)})$ and $N(\mu_i^{(m)}, \Sigma_i^{(m)})$ and can be evaluated as the above. Krzanowski (1983) discussed the case with $r = s = 1/2$ for k populations.

Moreover, two alternatives are

$$J_L(r,s) = 2\frac{I_L(r,s)}{I_L(2r,0)+I_L(0,2s)} \quad \text{and}$$

$$G_L(r,s) = \frac{I_L(r,s)}{\sqrt{I_L(2r,0)I_L(0,2s)}}$$

Most of the measures discussed in section 1 will be special cases of the above general similarity measures when they are applied to the location models.

4 General similarity measures with mixed variables: Sample case

In practice, the general similarity measures between two groups of sample data will be evaluated. First, we can adopt the procedures of Daudin (1986) or Krusinska (1989) to select the variables which will construct the location models; Daudin's procedure is based on Akaike's criterion whereas Krusinska's procedure is based on the multivariate discriminatory measure similar to the distance measure. To obtain the sample estimates of general similarity measures, the simplest way is to treat the data in either group as a sample from the corresponding population π_i and to replace

all parameter values by their sample estimates (maximum likelihood estimates in terms of conditional likelihood on each state), and this may be called PLUGALL. As the parameters r and s in the general similarity measures, researcher will select the appropriate values ($\frac{1}{2}$ or 1) to meet the forms of similarity measure in the applications. For sufficiently large samples, the sample estimates of general similarity measures can be obtained for models L_1, L_2 and L_3. However, it would generally make sense to pool across those categories which have relatively few observation; that is, L_3 is the common model may be encountered. The task remains is to evaluate the statistical properties of the estimators of various general similarity measures. The mathematical difficulties in deriving the properties of the estimators are formidable, and consequently we will evaluate the properties by the resampling methods - jackknife and bootstrap. These results will be discussed elsewhere.

References

Bhattacharyya, A. (1943), On a measure of divergence between two statistical populations defined by their probability distributions, *Bull. Calcutta Math. Soc.* 35, 99-109.

Chernoff, H. (1973) Some measures for discriminating between normal multivariate distributions with unequal covariance matrices, in "*Multivaraite Analysis III* " (P. R. Krishnaiah, Ed.), 337-344, Academic Press, New York.

Daudin, J. J. (1986) Selection of variables in mixed-variable discriminant analysis, Biometrics, 42, 473-481.

Good, I. J. and Smith, E. P. (1987) Some general measures of similarity and their values under multinormality, *J. Stat. Comp. Simul.* 28, 75-79.

Hellinger, E. (1904) "Die orthogonalinvarienten quadratischer formen von unendlich vielen variabelen" Gottingen dissertation, 84; reviewed in *Jahrbuch der Math.* 38 (1907), 153-156.

Jeffreys, H. (1948) *Theory of Probability*, 2nd ed. Clarendon Press, Oxford.

Krusinska, E. (1989) New procedure for selection of variables in location model for mixed variable discrimination, Biom. J., 81, 511-523.

Krzanowski, W. J. (1983) Distance between population using mixed continuous and categorical variables, *Biometrika*, 70, 235-243.

Kullback, S. and Leibler, R. (1951) On information and sufficiency. *Ann. Math. Statist.*, 22, 79-86.

Lu, R., Smith, E. P. and Good, I. J. (1989) Multivariate measures of similarity and niche overlap. *Theo. Pop. Biol.*, 35, 1-20.

MacArthur, R. and Levins, R. (1967) The limiting similarity, convergence, and divergence of coexisting species, Amer. Nat. 101, 377-385.

Matusita, K. (1955) Decision rules, based on the distance, for problems of fit, two samples, and estimation. Ann. of Math. Statist., 26, 631-640.

Matusita, K. (1966) A distance and related statistics in multivariate analysis, in "*Multivaraite Analysis I* " (P. R. Krishnaiah, Ed.), 187-200, Academic Press, New York.

Morisita, M. (1959) Measuring of interspecific association and similarity between communities, *Mem. Fac. Sci. Kyushu Univ. Ser. E.*, 65-80.

Olkin, I., and Tate, R. F. (1961) Multivariate correlation models with mixed discrete and continuous variables, *Ann. Math. Statist.*, 32, 448-465.

Pianka, E. R. (1974) Niche overlap and diffuse competition, *Proc. Natl. Acad. Sci. USA* 71, 2141-2145.

Sibson, R. (1969) Information radius. *Z. Wahr. verw. Geb.*, 14, 149-160.

The MD4* Algorithm: Randomizing Nonrandom Bits

Mark J. Kiemele Philip L. Mayfield
Department of Mathematical Sciences
United States Air Force Academy
USAF Academy, CO 80840-5701

Abstract

In theory, it is difficult to define a hash function which is capable of creating random data from nonrandom data. This paper addresses the randomization properties of an extremely fast, compact hash function. The MD4 message digest algorithm produces a 128-bit output or "message digest" from an arbitrarily-long input string of bits. The results of a variety of empirical tests which were conducted to detect possible statistical defects in the algorithm are presented. . .

This paper presents the results of a statistical analysis of the randomization properties of the MD4 Algorithm [7]. The MD4 message digest algorithm is a fast, compact hash function which maps an arbitrarily-long string of bits onto a 128-bit quantity. For a complete description of the algorithm, the reader is referred to Rivest [7]. The investigation of MD4 consisted of a series of six empirical tests in which a large number of 128-bit outputs was generated and then examined for randomness, or the lack thereof. The results of these tests are as follows.

The first test conducted was a byte parity test. The appropriate hypotheses for this Chi-Square test are presented as follows:

H_0: Odd/Even parity of bytes are equally likely.
H_1: Odd/Even parity of bytes are not equally likely.

From a total of one million iterated applications of MD4, each of the 16 byte positions of the one million outputs was examined for parity. The results of this test are shown in Table 1. It is apparent that the null hypothesis cannot be rejected, indicating that each byte is equally likely to be odd or even. In fact, the extremely high P-value (.9813) might lend statistical credence to the algorithm's "purposeful smashing of bytes."

Byte Position	Actual	Expected	Chi-Square Contribution
1	500979	500000	1.916882
2	499412	500000	.691488
3	499985	500000	.000450
4	500510	500000	.520200
5	500513	500000	.526338
6	499849	500000	.045602
7	499780	500000	.096800
8	499808	500000	.073728
9	500624	500000	.778752
10	499776	500000	.100352
11	499787	500000	.090738
12	499922	500000	.012168
13	500242	500000	.117128
14	499414	500000	.686792
15	499939	500000	.007442
16	500347	500000	.240818
		Total	5.905678
		P-value	.9813

Table 1. Byte Parity Test

* MD4 is the product of Ron Rivest, MIT Laboratory for Computer Science, 1990.

A second test conducted is a check for uniformity in the bivariate distribution of byte position versus byte value. The hypotheses tested are as follows:

H_0: Bivariate distribution of byte position vs byte value is uniform.
H_1: Bivariate distribution is not uniform.

Three million iterated applications of MD4 were performed, and the results of examining the decimal integer value of each byte in each of the three million outputs are shown in Table 2.

	Byte Position			
	1	2	. . . . .	16
0	11645	11591	. . . . .	11678
1	11722	11658	. . . . .	11780
V .	.	.		.
a .	.	.		.
l .	.	.		.
u .	.	.		.
e .	.	.		.
255	11832	11823	. . . . .	11483

Bivariate Frequency Distribution

E(X) = 11718.75 χ^2_0 = 3895.87
Min = 11354 df = 4095
Max = 12099 P = .54

Table 2. Uniformity of Byte Position vs Byte Value

The results indicate that the distribution is indeed uniform. One can also conclude independence between position and value. That is, given a particular byte position, the byte value is equally likely to be any of the 256 possible values. Similarly, given a particular value, it is equally likely to occur in any of the 16 byte positions.

A third frequency test was then conducted, this time at the bit level. The hypotheses for this test are expressed as follows:

H_0: The distribution of 1's across all 128 bit positions is uniform.
H_1: This distribution is not uniform.

Another three million outputs from MD4 were generated and each of the bit positions examined to determine the frequency of 1's in each position. Table 3 provides the results of this test. These results again indicate uniformity across the 128 bit positions, i.e., each bit is equally likely to be a 0 or 1.

Bit Position	Actual	Expected	Chi-Square Contribution
1	1,499,496	1,500,000	.169
2	1,500,769	1,500,000	.394
3	1,501,119	1,500,000	.835
4	1,500,256	1,500,000	.037
5	1,500,256	1,500,000	.044
.	.	.	.
.	.	.	.
126	1,499,975	1,500,000	.000
127	1,499,466	1,500,000	.190
128	1,500,624	1,500,000	.260

Min = 1,498,191 χ^2_0 = 70.657
Max = 1,502,403 df = 127
P = .85

Table 3. Frequency Test for Bit Positions

The fourth test, a gap test, examined another set of one million outputs from MD4. Each output was scanned for the number of 0's between successive 1's. For example, the string "10010110001" has gaps of 2, 1, 0, and 3, respectively. Table 4 shows the total number of observed gaps for gaps of size 24 or less. No gaps of size 25 or larger were encountered. The successive halving of the number of observed gaps for an incremental gap size of 1 is what we would expect to see if the probability of a 1 or 0 in each bit position is .5.

Gap Size	Observed #	Gap Size	Observed #
0	32,263,081	13	3,583
1	16,247,658	14	1,793
2	8,063,542	15	881
3	4,001,042	16	421
4	1,984,905	17	209
5	983,004	18	121
6	488,068	19	59
7	241,362	20	27
8	120,025	21	18
9	59,219	22	5
10	29,843	23	3
11	14,471	24	2
12	7,337		

Table 4. Gap Test

The fifth test conducted was one in which the difference (in absolute value) between the number of 1's and the number of 0's occurring in each of one million outputs was noted. The observed (actual) frequencies, as well as expected frequencies (under the assumption that a 0 or 1 in any position is equally likely), are shown in Table 5. These results clearly support the assumption.

Difference	Actual	Expected	Chi-Square
0	70331	70386	.043
2	138539	138606	.032
4	132566	132306	.511
6	122122	122433	.790
8	109900	109829	.046
10	94974	95504	2.941
12	80948	80496	2.538
14	65734	65757	.008
16	52053	52058	.000
18	39905	39935	.023
20	29959	29681	2.604
22	21426	21370	.147
24	14928	14903	.042
26	10117	10064	.219
28	6508	6581	.810
30	4205	4165	.384
32	2532	2551	.142
34	1480	1512	.677
36	870	866	.018
38	490	480	.208
40	277	257	1.556
42	115	133	2.436
44	67	67	.000
46	32	32	.000
48	12	15	.600
50	4	7	1.286
52	5	5.833	.119
54	0	2.436	2.436
56	1	.980	.000
$\chi^2_0 = 18.603$		df = 27	P = .884

Table 5. Differences Between # of 1's and # of 0's

A final test was conducted to examine the avalanche effect of MD4. A series of 30,000 comparisons was made, where each comparison compared two outputs of MD4. The two outputs compared were the outputs corresponding to two "almost identical" inputs to MD4. That is, if A is a 32,000-byte (256,000-bit) string and A' is also a 32,000-byte string which differs from A in only 1 bit position, we then compared MD4(A) and MD4(A'), each being a 128-bit string. We looked at the Hamming distance between MD4(A) and MD4(A'), i.e., the number of bit position changes that occurred between MD4(A) and MD4(A'). The hypotheses tested can be described as follows:

H_0: The distribution of Hamming distances is binomial with $n=128$ and $p=.5$.

H_1: This distribution is not binomial with $n=128$ and $p=.5$.

The frequency distribution of Hamming distances that occurred among the 30,000 comparisons is shown in the histogram of Figure 1.

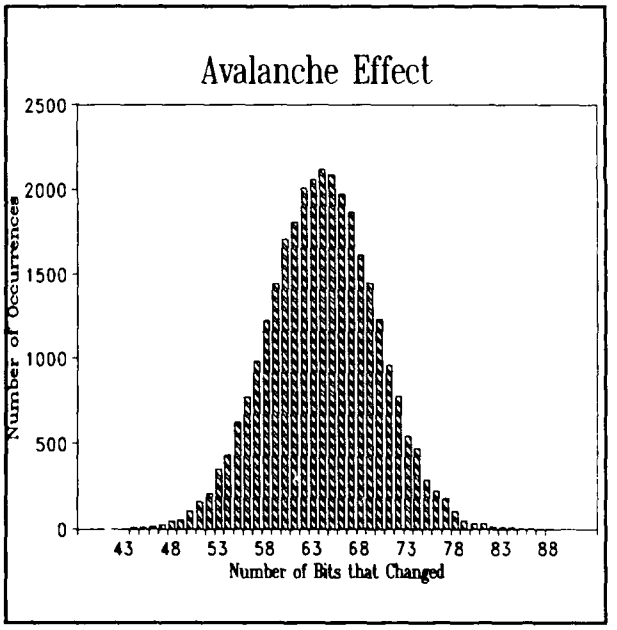


Figure 1. Avalanche Effect: Hamming Distance

This figure suggests that, on the average, about half (64) of the bits will change. While this figure gives us an indication of how many bits will change, Table 6 shows us that, of the bits that do change, each of the bit positions tends to contribute equally to the number of changes. Clearly, the avalanche effect demonstrated is one in which the outputs for two "almost identical" inputs appear to be as random as any other two randomly chosen 128-bit strings.

The results shown here indicate that MD4 is a byte smasher extraordinaire. These random properties of MD4, together with its speed and compactness, make it a

potentially valuable tool for a variety of applications, including virus detection and compressing large files prior to signing them with a public-key algorithm such as RSA.

Bit Position	Actual	Expected	Chi-Square
1	14,995	15,000	.002
2	14,997	15,000	.001
3	15,123	15,000	1.009
4	14,956	15,000	.129
5	14,915	15,000	.482
.	.	.	.
.	.	.	.
.	.	.	.
126	14,855	15,000	1.402
127	15,029	15,000	.056
128	15,092	15,000	.564

Min = 14,811
Max = 15,294

χ^2_0 = 65.148
df = 127
P = .89

Table 6. Avalanche Effect: How often each of the 128 bit positions changed in the 30,000 comparisons

References

1. Damgård, Ivan Bjerre. Design Principles for Hash Functions. ***Proceedings CRYPTO '89***, pp 395-405.

2. Denning, Dorothy E. R. ***Cryptography and Data Security***. Addison-Wesley, 1982.

3. Knuth, Donald E. ***The Art of Computer Programming (2nd ed), Vol 2/Seminumerical Algorithms***. Addison-Wesley, 1981.

4. Merkle, Ralph C. A Fast Software One-Way Hash Function. ***Journal of Cryptology***, Vol. 3, No. 1 (1990), pp 43-58.

5. Merkle, Ralph C. One-Way Hash Functions and DES. ***Proceedings CRYPTO '89***, pp 407-419.

6. Patterson, Wayne. ***Mathematical Cryptology for Computer Scientists and Mathematicians***. Rowman & Littlefield, 1987.

7. Rivest, Ronald L. The MD4 Message Digest Algorithm. ***Proceedings CRYPTO '90***, pp 281-291.

Massively Parallel Simulation and Optimization of Queueing Networks

Pirooz Vakili and Edward Lau
Department of Manufacturing Engineering
Boston University

April 1991

Abstract[1]

We simulate several variants of a class of queueing networks - corresponding to different system parameter values or operating policies - simultaneously. One clock mechanism is used to drive all the variants. This clock synchronizes the system trajectories such that the "same event" takes place at the "same time" at all systems. This synchronization is the basis of the massively parallel algorithms we develop. Implementation of the algorithms on the massively parallel Connection Machine and the implications of the approach for performance optimization is discussed.

1 Introduction

There is an inherent partial parallelism in networks of queues. Often each server operates as an independent entity as long as customers are present to be served. The effect of other servers is experienced through idle periods - where no customer is present - or blocked periods - where no space is available for a served customer (to illustrate we are considering a simple scenario). While the status of the servers (busy, idle, blocked) remains unchanged, they can be simulated independently and in parallel. Most parallel algorithms for queueing simulation use this partial parallelism for simulating one "large" network (see[4]).

In contrast, we consider the simulation of a "large" number of variants of a "nominal" network that differ, for example, in their routing schemes, buffer configurations, service or arrival rates, or the number of customers in the system. Obviously there is a total parallelism among the variants. More importantly, we simulate each variant as a network of autonomous servers (i.e. servers that determine the "departure times" of customers independently of the presence or absence of customers); the same autonomous servers can be shared between all the variants. We call this approach the Standard Clock (SC) technique [3, 6, 7] since a single simulation clock mechanism (that may be standardized) is defined which drives all the variants simultaneously. This clock synchronizes the system trajectories such that the "same event" takes place at the "same time" at all systems. The obtained synchronization is the basis of the algorithms we develop for the implementation on the massively parallel Connection Machine (CM). This approach is applicable to queueing networks that can be modeled as Generalized Semi-Markov Processes (GSMP) with bounded hazard rate event life times. For networks that can be modeled as continuous time uniformizable Markov chains, SC is based on the well known uniformization procedure.

An important feature of this approach is the concurrent evaluation of the performance of the network at very large numbers of parameter values or operating policies. We believe this feature opens up new possibilities for performance modeling and optimization. As a first step we consider a global random search for performance optimization of a queueing network.

Section 2 defines our model of a single queueing network; a parameterization of the model is considered in section 3; the Standard Clock algorithm and its massively parallel implementation is given in section 4, and in section 5 we consider solving a stochastic optimization problem via massively parallel simulation.

2 Model : systems driven by marked Poisson processes

Let $(\tau, \epsilon) = \{(\tau_n, \epsilon_n); n \geq 0\}$ be a marked Poisson process where $\{\tau_n; n \geq 0\}$ is the sequence of arrival instances of a Poisson process N, and $\{\epsilon_n; n \geq 0\}$ is an I.I.D. sequence of discrete random variables, independent of the

[1] The work in this paper was partially supported by the National Science Foundation under Grant DDM-8914277.

Poisson process N, such that $\epsilon_n \in E$, where E is a finite set called the set of events.

Let S, a denumerable set, be the set of "physical" states of the system. If upon the occurence of an event $e \in E$, the state of the system is $x \in S$, then the next state of the system $x' \in S$ is determined via a given state transition rule:

$$x' = f(x, e, W) \tag{1}$$

W is a random variable used to model probabilistic transitions.

Let $X(0)$ be the random variable of the initial state. The sequence of states $\{X(n); n \geq 1\}$ is defined recursively by $X(n) = f(X(n-1), \epsilon_n, W_n)$ and the process $X = \{X(t); t \geq 0\}$ is defined as follows:

$$X(t) = \sum_{n=0}^{\infty} X(n) I\{\tau_n \leq t < \tau_{n+1}\} \text{ for } t \geq 0 \tag{2}$$

This model is quite versatile: open and closed networks of queues with multiple classes of customers, Markovian routing, finite and infinite buffer spaces, and a variety of service disciplines can be modeled as such. Networks with exponential service times and inter-arrival times provide the most staightforward examples but networks with phase-type service times and inter-arrival times can be modeled as well by considering a more intricate state space.

To illustrate we consider a simple example:

Example 2.1 : Consider a tandem network of K exponential servers with rates $\mu_1, \ldots, \mu_K$ respectively. There are B_i buffers between server i and server $(i+1)$ $(i = 1, \ldots, K-1)$. We assume that there are no spaces at the servers. There is an infinite supply of parts at server 1 and infinite space for finished parts after server K. Server i begins processing a part only if the immediate down stream buffer, i.e. B_i, is not full (the so-called communication blocking). In this case:
$S = \{\underline{x} = (x_1, \ldots x_{K-1}); 0 \leq x_i \leq B_i\}$
$E = \{d_1, \ldots d_K\}$ (d_i = departure from server i)
τ = arrival instances of a Poisson process with rate $\Lambda = \mu_1 + \ldots \mu_K$.
$\text{Prob}(\epsilon_n = d_i) = \mu_i / \Lambda$.
Let m_i be a $(K-1)$ dimensional vector with ith entry equal to -1, $(i+1)$the entry equal to 1, and all other entries equal to 0 $(1 \leq i < K-1)$. Let m_{K-1} a vector with $(K-1)$th entry equal to -1 and all other entries equal to 0, then

$$f(\underline{x}, d_i, W) = \begin{cases} \underline{x} + m_i & \text{if } x_i > 0,\ x_{i+1} < B_{i+1} \\ \underline{x} & \text{otherwise} \end{cases}$$

(for d_1 only $x_1 < B_1$, and for d_K only $x_{K-1} > 0$ is required.)

3 A parametric family of systems driven by the same marked Poisson process

To consider several variants of a "nominal network" we parameterize the system with respect to a parameter of interest. The parameterization may be with respect to the number of buffers, buffer configurations, routing proportions, number of customers, control policies, service and inter-arrival rates or any combinations of the above. The parameterization of a model of the system can be accomplished through the state transition function f while leaving the marked Poisson process (τ, ϵ) unchanged.

Example 3.1 : Consider the parameterization of example 2.1 through buffer configurations such that $\sum_{i=1}^{K-1} B_i = C$.
Let $B = \{(B_1, \ldots, B_{K-1}); B_i \geq 1, \sum_{i=1}^{K-1} B_i = C\}$. For each $b \in B$ let S_b be the state space corresponding to configuration b, and let E, τ, and ϵ be as defined in example 2.1. The state transition rules for each configuration b are defined by

$$f^b(\underline{x}^b, d_i, W) = \begin{cases} \underline{x}^b + m_i & \text{if } x_i^b > 0,\ x_{i+1}^b < B_{i+1}^b \\ \underline{x} & \text{otherwise} \end{cases}$$

(for d_1 only $x_1^b < B_1^b$, and for d_K only $x_{K-1}^b > 0$ is required.)
Note that the same (τ, ϵ) (model of the simulation clock mechanism) is used for all $b \in B$. The next section describes algorithms for using one clock mechanism to drive many systems simultaneously.

4 Standard Clock Algorithm and Massively Parallel Implementation

Assume M variants of a "nominal network" corresponding to M distinct parameter values or operating policies are given. Assume further that the nominal system can be modeled by the model described in section 2 and that the variants are parameterized through the state transition rules f as described in section 3. Let (τ, ϵ) be the common marked Poisson process and $f^1, \ldots, f^M$ the state transition rules associated with variants $1, \ldots, M$, respectively. The simulation algorithm consists of two parts: algorithm A that simulates the clock mechanism (generates samples of marked Poisson process (τ, ϵ)) and algorithm B that describes the simultaneous updating of the system states upon occurrence of events.

Let $E = \{e_1, \ldots, e_K\}$ be the set of events. We use the Alias method to generate samples of ϵ_n. To use this method it is necessary to initially generate two K dimensional vectors R and A. We refer the reader to [2] for the algorithm to generate these vectors and assume

here that R and A are generated. Then:

Algorithm A: determining τ_{n+1} and ϵ_{n+1}

1. generate t_{n+1}, a sample of an exponential r.v. with rate 1.
   Set $\tau_{n+1} = \tau_n + t_{n+1}/\Lambda$.
2. generate u_{n+1}, a sample of a uniform$(0,1)$ r.v. U_{n+1} let $i = [Ku_{n+1}] + 1$ ($[x]$ denotes the integer part of x).
3. generate v_{n+1}, a sample of a uniform$(0,1)$ r.v. V_{n+1} if $v_{n+1} \leq R[i]$ then $\epsilon_{n+1} = e_i$
   else $\epsilon_{n+1} = A[i]$.

This clock mechanism is simple and very efficient. In fact, except for the generation of vectors R and A, that can be accomplished in $O(K)$ and is performed only once at the beginning of the simulation, the execution of the clock mechanism is essentially independent of K, the number of events in the system.

Let X^j be the state of the variant j at time τ_n $(j = 1, \ldots, M)$. Then:

Algorithm B: updating the states of the systems at τ_{n+1}
(Assume that $\epsilon_{n+1} = e_i$)

1. generate w_{n+1}, a sample of a uniform$(0,1)$ r.v. W_{n+1}.
2. For $j = 1, \ldots, M$
   set $X^j(n+1) = f^j(X^j(n), e_i, w_{n+1})$

Massively parallel implementation

The Connection Machine (CM) that we have used as the platform for the massively parallel implementation of the SC algorithm is a SIMD (Single Instruction Multiple Data) computer. It consists of a large number of small processors (32000 in our case) each with its associated memory. All the processors operate under the direction of a serial computer, called the front end. The front end acts as a central control mechanism that directs all processors as to the next instruction to be executed. All processors then execute the same instruction; hence the name, Single Instruction Multiple Data (SIMD) systems (for more extensive description of parallel and distributed systems see [1],[4]). In massively parallel systems the synchronization of the computational tasks is a crucial element of the parallel implementation. The SC algorithm is particularly well suited for such implementation.

We simulate algorithm A (i.e. the clock mechanism) at the front end computer: at each tick of this clock the time and type of the "next" event is generated. Algorithm B is implemented in a distributed fashion at the CM: each processor of the CM simulates a version of the system with a distinct parameter value. The event type and time generated by the clock is broadcast to all processors which in turn execute the instruction corresponding to the event type. This execution is done according to the parameter value at the processor. To illustrate we consider the parallel implementation of the model of example 3.1 at a finite number (M) of parameter values:

Example 4.1 : The implementation of the clock mechanism (Algorithm A) at the front end is trivial. To implement Algorithm B we define parallel variables $\bar{x}_1, \ldots, \bar{x}_{K-1}$ to represent the states of the systems at all variants: $\bar{x}_i$ is an M dimensional parallel variable whose every components is kept at a distinct processor. The value kept at processor j is the number of customers at buffer i at the configuration associated with processor j. Similarly we define parallel variables $\bar{B}_1, \ldots, \bar{B}_{K-1}$ (the jth component of $\bar{B}_i$, kept at processor j, is the number of buffers between i and $(i+1)$th servers at configuration j). Assume that the event reported by the front end (the clock mechanism) is d_i (for simplicity assume $1 < i < K-1$). To update the states of the systems we proceed as follows:
Define a logical parallel variable Δ as :

$$\begin{cases} \Delta = 1 & \text{if } \bar{x}_i > 0,\ \bar{x}_{i+1} < \bar{B}_{i+1} \\ \Delta = 0 & \text{otherwise} \end{cases}$$

and execute the following code (the code is executed in parallel on CM)
set $\Delta = (\bar{x}_i > 0 \text{ and } \bar{x}_{i+1} < \bar{B}_{i+1})$
$\bar{x}_i = \bar{x}_i - \Delta$
$\bar{x}_{i+1} = \bar{x}_{i+1} + \Delta$
The component of Δ at processor j takes value 1 if $x_i^j > 0$ and $x_{i+1}^j < B_{i+1}^j$ are both satisfied; otherwise it takes value 0 (these conditions are checked in parallel at each processor based on local information at the processor). The next two steps represent the movement of a part for all processors where $\Delta = 1$ and no action for those where $\Delta = 0$.

5 Performance optimization

Such massively parallel implementations dramatically increase our ability to generate data points (performance estimates) for analysing and optimizing queueing network performances. An immediate and important question to be answered is: what type of optimization algorithms are most appropriate in this context.

As a first step we have considered a random global search approach on the parameter space. Due to a lack of space and to the preliminary nature of our investigation our discussion below will be informal.

Consider the following optimization problem:

$$\text{Max}\{J(\theta); \theta \in \Theta\} \quad (3)$$

where $J(\theta) = \lim_{T\to\infty} \frac{1}{T} g(X(\theta, \omega, T)$, ($\omega \in \Omega$ represents all the underlying randomness in the system). $J(\theta)$ is, for example, some average steady state performance of the network at parameter value θ. To address this problem we proceed as follows:

Let $\theta_1, \ldots, \theta_M$ be M parameter values in Θ chosen randomly according to some distribution on Θ. We run $X(\theta_1, \omega, .), \ldots, X(\theta_M, \omega, .)$ in parallel and evaluate $g(X(\theta_1, \omega, T_k)), \ldots, g(X(\theta_M, \omega, T_k))$ at different epochs T_k ($k = 1, 2, \ldots$). At each epoch the parameter values are ranked in descending order of $g(X(\theta_j, \omega, T_k))$. We choose the best L parameter values at each T_k (those with highest value of g); when this population "stabilizes" (i.e. when there is a small migration in and out of the population, or changes in ranking within the population) the simulation is stopped.

The following considerations has been the basis of our approach: our objective is to find near optimal solutions to (3). For "large" values of M, and "reasonable" performance functions $J(\theta)$, the top L parameter values at the termination of the simulation are expected to be "near-optimal" with "high probability", i.e. produce performance measures that are close to Max $J(\theta)$. Furthermore, in the context of networks of queues they are expected to reveal some of the "desirable" properties of near-optimal variants. A concurrent comparison of sample performances of all variants is possible because in the SC simulation, all variants live in the same simulated world. This approach is identical to some of the coupling methods of sample paths of stochastic processes - by defining them on the same probability space - to establish stochastic monotonicity [e.g. see 5].

Example 5.1 : Consider the system of example 3.1 with the following modification: there are 11 servers in the system and server i is Erlang(r_i, μ). Consider the problem of optimal allocation of 20 buffers between the servers in order to maximize throughput. In our example $\mu = 1$ and $(r_1, \ldots, r_{11}) = (1, 2, 5, 4, 4, 2, 2, 3, 2, 5, 2)$. 4000 variants of the system (numbered 1 through 4000) were randomly selected and simulated in parallel on CM. At $T_1 = \tau_{5000}, T_2 = \tau_{10000}$, and $T_3 = \tau_{15000}$ the parameters were ranked (the simulation was performed in about 20 sec). By T_3, the top 20 ranked variants had "stabilized" (by this time in the "best" configuration 243 parts were produced). The ranked configurations were also observed at $T_4 = \tau_{100000}$, $T_5 = \tau_{150000}$ to check for possible long term change in ranking. The Table below shows the rank of 5 top configurations at T_1, T_3, T_4, T_5.

config.	R at T_1	R at T_2	R at T_3	R at $T4$
570	1	1	1	1
3321	19	4	2	2
907	6	3	3	3
1756	15	6	4	4
1277	3	2	5	5

They correspond to the following allocation of buffers (we have also included the number of parts produced at these configurations by T_5:

$b_{570} = (1, 2, 3, 3, 2, 1, 2, 2, 2, 2)$, Parts produced = 2449.
$b_{3331} = (1, 2, 3, 2, 2, 1, 2, 2, 3, 2)$ Parts produced = 2397.
$b_{907} = (1, 3, 3, 2, 2, 1, 2, 2, 2, 2)$ Parts produced = 2385.
$b_{1756} = (2, 2, 3, 2, 2, 1, 2, 2, 2, 2)$ Parts produced = 2378.
$b_{1277} = (1, 2, 3, 2, 2, 3, 1, 2, 2, 2)$ Parts produced = 2362.

References

[1] Bertsekas, D.P., Tsitsiklis, J. N., *Parallel and Distributed Computation: Numerical Methods*, Prentice hall, 1989.

[2] Bratley, P. B., Fox, B., and Schrage, L., *A Guide to Simulation* , Springer-Verlag, 1983.

[3] Ho, Y. C., Li, S., and Vakili, P., "On the Efficient Generation of Discrete Event Sample Paths under Different System Parameter Values" *Mathematic and Computers In Simulation* , 30, pp. 347-370, 1988.

[4] Righter, R. and Walrand, J.C., " Distributed Simulation of Discrete Event Systems", *Proceedings of the IEEE*, Vol. 77, No. 1, pp. 99-113,1989.

[5] Shanthikumar, J. G. , and Yao, D.D. "Monotonicity and Concavity Properties in Cyclic Queueing Networks with finite Buffers", *Queueing Networks with Blocking* H. Perros and T. Altiok, eds. Elsevier Science pp.325-344, 1989.

[6] Vakili, P. "Using a Standard Clock Technique for Efficent Simulation", to appear in *Operations Research Letters*. 1992.

[7] Vakili, P. "Massively Parallel and Distributed Simulation of a Class of Discrete Event Dynamic Systems: A Different Perspective", Manuscript. 1991.

VERSION 3 OF GPSS/SAS COMPILER

Gretchen K. Jones, National Center for Health Statistics
Michael A. Greene, The American University

Abstract

This paper describes Version 3 of a GPSS compiler. GPSS is a discrete event simulation language used to model queuing problems. The complier was written in the SAS language (version 6.06), which was chosen for three reasons: (1) it has character string handling and other functions required for a compiler, (2) the SAS language has a full range of mathematical and statistical functions that are used to extend the GPSS syntax and (3) the statistical procedures in the SAS system are available to preprocess data for the simulation or to postprocess simulation output.

The current version of the compiler implements much of the GPSS functionality and contains the usual devices in a simulation language including a clock mechanism, an event scheduler, a source of random numbers following a large number of probability distributions and data structures to represent queues and other required quantities.

I. Introduction

A. Simulation Language

A simulation language is a computer language which facilitates the programming of models for discrete-event simulations. It is useful for solving queueing problems because it has constructs which represent all the aspects of the queuing situation. It is possible but tedious to program a simulation problem in a high level language such as Fortran. A simulation language automatically handles many tasks such as maintaining a simulated clock, scheduling events and causing them to occur in the proper time-ordered sequence. In addition, most simulation languages automatically collect data describing the model's simulated behavior and print out summaries of these data. Thus much of the underlying logic of the simulation of the queuing problem is built into the simulation language.

We describe also in the paper how the compiler performs typical functions such as storage allocation, symbol table maintenance, cross referencing, garbage collection and error messaging. Applications for this compiler and some thoughts on using the SAS language as the development are also discussed.

A GPSS program consists of a sequence of statements, called blocks, which correspond to the boxes in the flow diagram of a queing model. The GPSS/SAS compiler translates a GPSS program into a SAS program using the SAS language first. Entities called transactions (for example, representing customers) move through these blocks. At any simulated instant, there may be many transactions in different parts of the flow diagram. Transactions can model the movement of customers through a facility. Usually a transaction is on one of two lists, the current events chain (CEC), or the future events chain (FEC). Transactions on the CEC are moving or ready to move through the blocks in the program. They can be held up if a block refuses entry or can be delayed. Transactions on the FEC will move later when the simulated clock reaches their block departure time, at which time they will be transfered to the CEC to continue progress. At any instant of simulated time, GPSS tries to move all each current (CEC) transaction as far as possible through the block diagram. Every transaction has a priority, which can be changed as it goes through the program. The CEC is in order of highest to lowest priority, causing transactions of high priority to move before those of lower priority at any given simulated time. Transactions also have parameters which may be used to carry data.

B. Why SAS?

SAS was chosen as the language in which to write the compiler for the following reasons: (1) the completeness and flexibility of SAS as a programming language (2) the capability for outputs to be analyzed through the immediate access to SAS's high quality statistics and graphics procedures, and (3), SAS has good random number generators, built-in mathematical functions and character string-handling functions useful in parsing program coding. The disadvantages to using SAS are that there are no multidimensional arrays and the execution speed is relatively slow.

C. Background on Previous Versions

The original version of the GPSS/SAS compiler is described in 'A GPSS-like Language in SAS for Discrete Event Simulation' (Proceedings of SUGI, 1988). At that time, the program consisted of a single SAS data step. The GPSS language statements which were implemented were GENERATE,

ASSIGN, TRANSFER, ENTER, ADVANCE, LEAVE, AND TERMINATE.

The second version of the program is described in 'How to Stop a Simulation' (Proceedings of SUGI, 1990). At this point, the compiler was separated into five data steps, in order for the work to be modularized (see figure 1). The main data steps were LEXAN, PASS2 and RUNSTEP. These data steps call TABLES (which contains the symbol table),and ERRMSGS (which contains the error messages). LEXAN is a lexical analyzer those main job is to translate free format mixed case input to fixed format uppercase, and also to compress spaces. The output from LEXAN is passed to PASS2, where most of the compiling and code generating takes place. Output from PASS2 goes into RUNSTEP, where the simulation execution occurs. The operation of the simulated clock, the scheduling of events, and the movement of transactions from block to block is all a part of RUNSTEP.

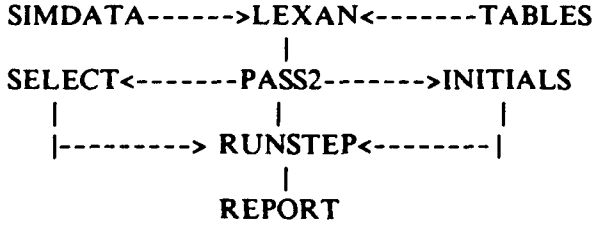
In addition to the features implemented in Version 1, two new statements were implemented in Version 2: REGS (regenerative start), and REGE (regenerative end), blocks which cause counting of the number of transactions and waiting times . These features were meant to be used to collect queue statistics. This then permits stopping the simulation after a completion of enough events to allow interval estimation of parameters with appropriate precision.

We have now completed Version 3 of the compiler. Version 3 has the same structure as Version 2, but implements a much larger subset of GPSS including MATRIX handling, parameters, GATES and LOGIC, TEST, etc.

II. The structure of our Compiler.

Version 3 consists of three main data steps working with 5 files. Figure 1 below shows the way they work together.

Figure 1
Compiler Phases



A. LEXAN, a lexical analyzer. Changes the free format of the program, SIMDATA, to fixed format, lowercase to upper case, does space compression, puts entities into labels, and reads in symbol TABLES. It passes the analyzed GPSS program to PASS2.

B. PASS2 does entity translation, symbol table maintenance, storage allocation, macro variable creation for array dimensioning going into the INITIALS data set, syntax analysis, compile time error messaging, translation of GPSS random number calls to SAS random number subroutine calls, and creation of the dynamic half of RUNSTEP, the file SELECT. It passes the compiled program to RUNSTEP.

C. RUNSTEP does the actual execution of the compiled code from SELECT and INITIALS. Dynamic storage allocation is done from INITIALS. Also done are garbage collection from parameter arrays, run-time error messaging, simulation event tracing and output in the form of REPORT.

III. Examples and sample output from the compiler.

The text below describes how GPSS operands are translated into SAS statements by the compiler. The usual form of a GPSS language statement is:

LAB OP-FLD AUX OPRNDS

where LAB refers to LABEL, OP-FLD refers to OPERAND-FIELD, AUX refers to AUXILIARY, and OPRNDS refers to OPERANDS. LABELs identify either the statement or the entity such as a STORAGE or a MATRIX. In Figure 2 below, the first line has the label "MIKE" which is the name of the matrix to be dimensioned. Labels are sometimes optional. OPERATION FIELDs define the purpose of the GPSS statement. Line 1 has the operation "MATR(IX)" which causes dimensioning. Line 5 has operation "GENE" or "GENERAT" which causes production of a transaction. Auxiliaries are adjuncts to operations which further define the operation. Line 8 has an auxiliary, to TEST on LESS THAN. Operands (up to 8) are found to the right of operations (or auxiliaries if present).

Figure 2

```
MIKE    MATR  3 2
        INIT  M$MIKE(3,2) 1
        INIT  X$JON 2
XEROX   STOR  2
LABEL1  GENE  RANUNI(X$JON) 2 15 . . 21
        MSAV  MIKE 3 2 M$MIKE(3,2)+1
        ASSI  1 M$MIKE(3,2)
        TEST L M$MIKE(3,2) 12 SKIP
        ENTE  XEROX
        ADVA  5*P1
        LEAV  XEROX
SKIP    TERM  1
        STAR  4
        END
```

Figure 2 represents nearly original GPSS source code which has been processed by LEXAN.

The program is changed by PASS2. PASS2 performs GPSS entity translation, symbol table maintenance, storage allocation, macro variable creation for array dimensioning, syntax analysis, compile-time error messaging, translation of GPSS random number calls to SAS random number subroutine calls, and creation of the dynamic portions of RUNSTEP. In that part of the program, GPSS labels are assigned to SAS variables and initialized, storage for various GPSS entities is created, GPSS operands are translated to SAS expressions and pointers of various types are set up. Then RUNSTEP (Section III. C. above) needs only two pieces of information, (1) the type of operation field being executed, and (2) the values of the operands at the time the statement is being executed. The operation field code is passed through the PASS2 SAS dataset, while the operands values are obtained in one of the dynamic portions, the SELECT File.

The SELECT file evaluates the operands and the final values are set to be _T1, _T2, ...etc up to _T8. Figure 3 shows the SELECT file for statement number 5. (LABEL1 GENE ...) First a temporary value is set to the first savevalue, X{001}. Then a call to RANUNI is made with X$JON as the seed, the random number being put in _T101. Then X$JON is set back to the new seed, _T1 is set to _T101, the first operand. The third operand is set to 15, and the sixth to 2. What is occurring is that PASS2 is translating GPSS code into SAS code which then gets appended to the end of the RUNSTEP data step. In this manner any valid SAS statement can be used as GPSS operands, representing a substantial extension to the language.

Figure 3

```
When (005) DO;
 TEMP011 = X{001};
 CALL RANUNI(TEMP011,_T101);
 X{001} = TEMP011;
 _T1 = _T101;
 _T2 = .;
 _T3 = 15;
 _T4 = .;
 _T5 = .;
 _T6 = 2;
END;
```

The main data structures in RUNSTEP are those associated with transactions TA{*}, NEXT{*}, _PARMTX{*}(the parameter array), and block arrays (BLKTYPE{*}, BLKAUX{*}, BLKCNT{*} and BLKMISC{*}). The transaction array is dimensioned beforehand to the best guess at the maximum number of active transactions * 10. The block arrays are dynamically dimensioned to the block counts in INITIALS (see figure 4). The blockarrays contain information which is specific to the blocks. Filling the block arrays is the last step in compilation and is done in the beginning of RUNSTEP. BLKTYPE{*} gets a number representing the operation field of the block. BLKAUX{*} gets the auxiliary operand. BLKMISC{*} is used for miscellaneous operations on blocks such as the value of the logical evaluation of the test block, etc. BLKCNT{*} is the block count or the number of transactions which have passed through the block.

TA{*} is a linked list with the pointers in NEXT{*}. TA{*} contains most of the relevant information about the transaction including its block departure time (BDT), number, current block occupied by the transaction, transaction status (active,blocked or terminated),maximum number of parameters, pointer to starting place in the parameter matrix(_PARMTX{*}) and priority). BDT represents the time that the transaction may be moved from its current block. Transactions are linked by BDT and priority, that is NEXT{i} points to the transaction with the same (and lower priority) BDT or next larger BDT. This allows scanning the transaction array from the beginning in order to find the next transaction to be moved. Transactions are inserted in the Ta{*} array when created in the

GENERATE block and removed when they move through the TERMINATE block. The position of the transaction may be modified by traversing through an ADVANCE (which causes revision of the BDT) or a PRIORITY block (changing the priority).

Before beginning the simulation, at the point where each GENERATE block is symbolized, its first transaction is made, space allocated in the _PARMTX{*} array, its block departure time computed, and it is installed in the TA{*} linked list. The first transaction is then taken off the top of the linked list and the simulation clock is set to its BDT. Then the following pattern ensues:

1. The transaction is moved as far as it can be moved. It is then destroyed or put back into the linked list.

2. The next transaction is identified.

3. The simulation clock is updated if required to the block departure time for the next transaction.

4. If the termination counter is zero, the simulation stops, otherwise return to step 1.

Figure 4

INITIALS

```
ARRAY M    {*}  _M1 - _M7;
ARRAY OFF  {*}  _OFF1- _OFF2;
ARRAY NC   {*}  _NC1- _NC2;
ARRAY X    {*}  _X1 - _X2;
ARRAY BLOKTYPE {*}  _BT1 - _BT15;
ARRAY BLOKAUX  {*} $2 _BX1 - _BX15;
ARRAY BLOKCTS  {*}  _BCTS1 - _BCTS15;
ARRAY BLOKCNT  {*}  _BCNT1 - _BCNT15;
ARRAY BLOKMISC {*}  _BMIS1 - _BMIS15;
ARRAY STORCAP  {*}  _STC1 - _STC2;
ARRAY STORUSE  {*}  _STU1 - _STU2;

RETAIN
    _STC1   2    _STU1   0
    _NC1    2    _OFF1   1
    NULL   0000
    LABEL1 0006
    SKIP   0013
    ;
```

Conclusion:

We think the third version has represented a substantial extension over other versions. We plan to test this using practical applications in the near future.

REFERENCES

Jerry Banks, John S. Carson,II and John Ngo Sy (1989), Getting Started with GPSS/H. Wolverine Software Corporation, Annandale, VA.

Paul Bratley, Bennett L. Fox and Linus E. Schrage (1987), A Guide to Simulation, 2nd edition. Springer Verlag, New York.

Kenneth A. Dunning (1981), Getting Started in GPSS. Engineering Press, Inc. San Jose, CA.

Michael A. Greene and Gretchen K. Jones (1990), "How to Stop a Simulation," Proceedings of the SAS Users Group International, SAS Institute, Cary, NC.

Michael A. Greene and Gretchen K. Jones (1991), "Enhancements to the GPSS/SAS Compiler," Proceedings of the SAS Users Group International, SAS Institute, Cary, NC.

James O. Henriksen and Robert C. Crain (1989), GPSS/H Reference Manual, 3rd edition. Wolverine Software Corporation, Annandale, VA.

IBM (1971) General Purpose Simulation System V User's Manual, 2nd edition. IBM, N. Y.

Gretchen K. Jones and Michael A. Greene (1988), "A GPSS-like Language in the SAS System for Discrete Event Simulation," Proceedings of the SAS Users Group International, SAS Institute, Cary, N. C.

Gretchen K. Jones and Michael A. Greene (1989), "A Prototype Implementation of GPSS in SAS," Simulation, January.

SAS Institute Inc. (1990), SAS Language: Reference, Version 6, First Edition. SAS Institute Inc., Cary, NC.

Thomas G. Schriber (1974), Simulation Using GPSS. John Wiley and Sons, New York

Applying Bootstrap Methods to Simulation Output Analysis

Charles B. Rea, Wei-Kei Shiue and Chong-wei Xu
Southern Illinois University at Edwardsville

Abstract

Confidence intervals obtained by bootstrap methods and normal approximation are compared, based on output data from terminating and steady-state simulations. Bootstrap intervals are equal or better than normal approximation intervals in actual probability coverages. Furthermore, bootstrap methods capture the skewness in the distribution of outputs and, therefore, are more desirable than normal approximation.

1 Introduction

Computer simulation is a method for studying a system or process which is far too complex to easily derive analytic results for performance measure of interest. Usually several simulation runs are conducted and the resulting output data are employed to make inference about performance measure; for instance, the average delay in the queueing system. Here we assume proper steps have been taken so that the outputs from either terminating or steady- state simulation are independently and identically distributed. Law [4] gives precise definition of the two types of simulations. In this article the regenerative method is considered for the case of steady-state simulation. Central limit theorem (normal approximation) is the most common technique for constructing confidence interval for performance measure. This is because it is easy to use and, when the size of replications is large, it yields very accurate results. However, it does not capture the asymmetric nature of underlying distribution of the output data. Since the distribution of data is rarely known, we are dealing with nonparametric situation where bootstrap method [3] proves to be useful in that it takes into account of asymmetry involved and is as easy to implement as normal approximation. A brief description of bootstrap methods and related confidence intervals for a mean are given in section 2. Section 3 contains confidence intervals obtained by the two methods for M/M/1 queue and reliability model, which are pertinent to terminating simulation. M/M/1 queue is studied again using regenerative method in section 4. Comparisons are made of Jackknife and bootstrap confidence intervals for the steady-state average delay in the system. Section 5 includes some conclusions.

2 Bootstrap Methods

Bootstrap method is a resampling scheme. It uses a given set of independently identically distributed observations $Y = \{X_1, \ldots, X_n\}$ from an unknown distribution F to construct an empirical distribution $\hat{F}$. Random samples $Y_1^*, \ldots, Y_B^*$ are then taken from $\hat{F}$. This is the same as sampling from $\{X_1, \ldots, X_n\}$ with replacement. Suppose θ is the parameter of interest and $\hat{\theta}$ is an estimate of θ. The bootstrap estimates $\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*$ can be calculated from $Y_1^*, \ldots, Y_B^*$, which are used to assess the accuracy of $\hat{\theta}$ or to form bootstrap distribution $\hat{G}$, defined by $\hat{G}(s) = Pr[\hat{\theta}^* \leq s]$. Using αth and $(1-\alpha)$th percentiles of $\hat{G}$ as endpoints of interval will yield a $(1-2\alpha)$ 100% confidence interval for θ. This is the simplest of bootstrap methods for constructing confidence intervals and is called percentile method (P).

Improvements on the percentile method have been proposed, noticeably the bias corrected percentile method (BC) and bias corrected percentile acceleration method (BCa). Edgeworth expansion technique can be employed to get asymptotic expressions for the endpoints of the BCa, BC and percentile intervals for the case of estimating the population mean, μ [2].

$$\theta_{(BCa)}[\alpha] = \bar{x} + \frac{s}{\sqrt{n}}\{t(\alpha) + a[2t^2(\alpha) + 1]\} \quad (1)$$

$$\theta_{(BC)}[\alpha] = \bar{x} + \frac{s}{\sqrt{n}}\{t(\alpha) + a[t^2(\alpha) + 1]\} \quad (2)$$

$$\theta_{(P)}[\alpha] = \bar{x} + \frac{s}{\sqrt{n}}\{t(\alpha) + a\} \quad (3)$$

where $\bar{x}$ and s are the mean and standard deviation of the data, $t(\alpha)$ is the αth percentile of t distribution with $(n-1)$ degrees of freedom and

$$a = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^3}{6[\sum_{i=1}^{n}(x_i - \bar{x})^2]^{1.5}}$$

Thus, $(\theta_{(P)}[\alpha], \theta_{(P)}[1-\alpha])$ gives a $(1-2\alpha)$ 100% confidence interval for μ by the percentile method.

3 Confidence Intervals for Terminating Simulation

In order to compare bootstrap methods with normal approximation, we study the following systems.

- Model 1 — M/M/1 queueing system with utilization factor $\rho = 0.9$ ([5], p.289). Assuming that the number of customers in the queue at time 0 is zero, the performance measure of interest is the expected average delay in the queue for the first 25 customers entering the system, which is 2.124.
- Model 2 — Reliability model ([5], p.289) consisting of three components, each of which has a lifetime following Weibull distribution with shape parameter 0.5 and scale parameter 1.0. The model is structured in such a way that the system will function as long as component 1 works and either component 2 or 3 works. The performance measure of interest is the mean lifetime of the system, which can be shown to be 0.778.

The $(1-2\alpha)$ 100% confidence interval for the measure of each system, based on the central limit theorem, is

$$\bar{x} \pm t(1-\alpha)\frac{s}{\sqrt{n}} \qquad (4)$$

and the corresponding bootstrap confidence interval are given by equation (1), (2) and (3).

500 simulation runs are conducted for each model and, for each run, replication sizes n = 5, 10, 20 and 40 are considered. The true confidence level is 90%. The actual coverage probabilities along with 90% confidence interval of the true coverages are summarized in table 1 and 2.

Table 1 - Estimated Coverage Results for Model 1 (Terminating M/M/1 Queue)

Sample Size	5	10
Normal App.	.844 ± .027	.868 ± .025
Bootstrap(P)	.842 ± .027	.874 ± .025
Bootstrap(BC)	.838 ± .027	.880 ± .024
Bootstrap(BCa)	.840 ± .027	.882 ± .024

Sample Size	20	40
Normal App.	.880 ± .024	.882 ± .024
Bootstrap(P)	.880 ± .024	.886 ± .023
Bootstrap(BC)	.876 ± .024	.894 ± .023
Bootstrap(BCa)	.880 ± .024	.894 ± .023

Table 2 - Estimated Coverage Results for Model 2 (Weibull Model)

Sample Size	5	10
Normal App.	.700 ± .034	.758 ± .032
Bootstrap (P)	.710 ± .033	.762 ± .031
Bootstrap (BC)	.738 ± .032	.790 ± .030
Bootstrap (BCa)	.740 ± .032	.790 ± .030

Sample Size	20	40
Normal App.	.816 ± .029	.840 ± .027
Bootstrap (P)	.820 ± .028	.838 ± .027
Bootstrap (BC)	.836 ± .027	.842 ± .027
Bootstrap (BCa)	.780 ± .030	.842 ± .028

The distributions involved are quite skewed as indicated by the sample skewness, which are 1.755 and 5.35 for model 1 and 2, respectively. However, equation (4) always provides symmetric interval that is of course unrealistic. The asymmetry of a confidence interval for mean can be described by the asymmetry coefficient, defined by $\frac{UB-\bar{x}}{\bar{x}-LB}$, where UB and LB are upper and lower confidence bounds respectively. Table 3 and 4 contain the values of coefficient for each model. It is apparent that all bootstrap intervals capture this asymmetry.

Table 3 - Asymmetry Results for Model 1 (Terminating M/M/1 Queue)

Sample Size	5	10	20	40
Normal App.	1.000	1.000	1.000	1.000
Bootstrap (P)	1.046	1.061	1.056	1.049
Bootstrap (BC)	1.282	1.299	1.243	1.201
Bootstrap (BCa)	1.580	1.598	1.468	1.378

Table 4 - Asymmetry Results for Model 2 (Weibull Model)

Sample Size	5	10	20	40
Normal App.	1.000	1.000	1.000	1.000
Bootstrap (P)	1.075	1.100	1.109	1.101
Bootstrap (BC)	1.504	1.525	1.517	1.452
Bootstrap (BCa)	2.155	2.165	2.122	1.944

More evidence for supporting the asymmetric correctness of bootstrap confidence intervals can be found by studying a system of which the performance measure can be derived analytically.

- Model 3 — Estimation of mean service time for M/M/1 queueing system when the actual service times follow exponential distribution with mean 1.

The $(1-2\alpha)$ 100% level exact confidence interval for the mean service time can be calculated by

$$\left[\frac{2n\bar{x}}{\chi^2_{2n}(1-\alpha)}, \frac{2n\bar{x}}{\chi^2_{2n}(\alpha)}\right]$$

where $\chi^2_m(\alpha)$ is the αth percentile of chi-square distribution with m degrees of freedom.

Table 5 contains the average endpoints of normal approximation and bootstrap confidence intervals based on 500 simulation runs. The endpoints of exact intervals are also included. Exact intervals are asymmetric. Bootstrap intervals converge to them as $n \to \infty$, with BCa intervals being most correct. For this model the probability coverages of bootstrap methods are better than normal approximation. Sample skewness of the data is 2.091 and asymmetry results are given in table 6.

Table 5 - Comparison of approximate confidence intervals for model 3 versus exact confidence interval

Sample Size	5	10
Normal App.	0.174, 1.836	0.463, 1.550
Bootstrap (P)	0.193, 1.855	0.481, 1.567
Bootstrap (BC)	0.279, 1.941	0.539, 1.625
Bootstrap (BCa)	0.366, 2.027	0.579, 1.683
Exact	0.546, 2.538	0.637, 1.843

Sample Size	20	40
Normal App.	0.625, 1.360	0.740, 1.259
Bootstrap (P)	0.636, 1.372	0.747, 1.266
Bootstrap (BC)	0.671, 1.407	0.766, 1.285
Bootstrap (BCa)	0.705, 1.441	0.785, 1.304
Exact	0.717, 1.509	0.785, 1.325

Table 6 - Asymmetry Results for Model 3

Sample Size	5	10	20	40
Normal App.	1.000	1.000	1.000	1.000
Bootstrap (P)	1.047	1.066	1.065	1.053
Bootstrap (BC)	1.291	1.322	1.287	1.220
Bootstrap (BCa)	1.600	1.650	1.562	1.418

4 Confidence Intervals for Regenerative Simulation

In this section an example of steady-state simulation is considered.

- Model 4 — M/M/1 queueing system ([5], p.300) with utilization factor $\rho = 0.8$. The regenerative method developed by Crane and Iglehart [1] generates Y_i and X_i for each regenerative cycle, where Y_i represents the total delay in the queue of all customers served in the ith cycle and X_i represents the total number of customers served in the ith cycle. The performance measure of interest is the steady-state average delay in the queue given by $R = \frac{E(Y)}{E(X)} = 3.2$.

An estimator for R is $\hat{R} = \frac{\bar{Y}}{\bar{X}}$, but $\hat{R}$ is not unbiased. Jackknife technique can be employed to reduce bias and to construct confidence interval for R as follows.

1. For each i, compute z_i, where
$$z_i = n\frac{\bar{Y}}{\bar{X}} - (n-1)\frac{\sum_{j=1, j\neq i}^{n} Y_j}{\sum_{j=1, j\neq i}^{n} X_j}$$
2. A $(1-2\alpha)$ 100% Jackknife confidence interval is given by
$$\bar{z} \pm t(1-\alpha)\frac{\hat{\sigma_z}}{\sqrt{n}}$$
where $\bar{z}$ and $\hat{\sigma_z}$ are the mean and standard deviation of z_i's.

Note that as estimator for R, $\bar{z}$ has much less bias than $\hat{R}$. There is no closed form expression for bootstrap confidence interval for the ratio of two means problem. However, a crude bootstrap procedure can be applied directly to (Y_i, X_i) to obtain confidence interval.

Let $\mathbf{d} = \{e_1, e_2, \ldots, e_n\}$, with $e_i = \{Y_i, X_i\}$, $i = 1, 2, \ldots, n$.

1. Draw independent bootstrap samples, $\mathbf{d_1^*}, \ldots, \mathbf{d_A^*}$ by sampling from $\{e_1, e_2, \ldots, e_n\}$ with replacement;
2. Calculate from $\mathbf{d_1^*}, \ldots, \mathbf{d_A^*}$, the statistics
$\bar{R^*} = \frac{R_1^* + \ldots + R_A^*}{A}$ and $\hat{S^2} = \frac{\sum_{i=1}^{A}(R_i^* - \bar{R^*})^2}{A}$
where $R_i^* = \bar{y_i^*}/\bar{x_i^*}$ is calculated from $\mathbf{d_i^*}$;
3. Draw a bootstrap sample, say, $\mathbf{d^*}$ and compute $\bar{y^*}/\bar{x^*}$; Regard $\mathbf{d^*}$ as original data, repeat step 1 and 2 to obtain $\hat{S^*}^2$ and $Q^* = \frac{(\bar{y^*}/\bar{x^*}) - (\bar{y}/\bar{x})}{\hat{S^*}}$;

4. Repeat step 3 B times, and obtain, say, $Q_1^*, \ldots, Q_B^*$;
5. Sort $Q_1^*, \ldots, Q_B^*$ and construct bootstrap distribution $\hat{G}^*$;
6. Let z_α^* and $z_{1-\alpha}^*$ be respectively αth and $(1-\alpha)$th percentiles of $\hat{G}^*$, then $\theta[\alpha] = \hat{R} - z_\alpha^* \hat{S}$ and $\theta[1-\alpha] = \hat{R} - z_{1-\alpha}^* \hat{S}$ are the endpoints of a level $(1-2\alpha)$ 100% confidence interval for R.

The rationale behind the crude bootstrap is using bootstrap distribution $\hat{G}^*$ to approximate the distribution of $\hat{R}$ and approximation is enhanced by considering standardized $\hat{R}$. The precision of bootstrap intervals depend on A and B, which are 80 and 1000 respectively in this study. In practice, A in the range of 25 to 100 will give reasonable results. There is little gain in precision past $A = 100$. Guideline for determining B value is stated in [2] (p.181). Table 7 contains the coverage results of the two methods based on observations from 200 experiments. The true confidence level is 90%. Jackknife confidence intervals are easier to compute, but crude bootstrap intervals provide much improvement in coverage probability. An interactive program implementing the bootstrap algorithm mentioned above is available from the authors.

Table 7 – Estimated Coverage Results for Model 4 (Steady-state M/M/1 Queue)

Sample Size	64	128
Jackknife	.630 ±.056	.700 ±.053
Crude Bootstrap	.735 ±.051	.800 ±.046

Sample Size	256	512
Jackknife	.770 ±.049	.775 ±.049
Crude Bootstrap	.825 ±.044	.872 ±.039

5 Conclusion

The purpose of this article is to illustrate the usefulness of bootstrap methods in constructing confidence intervals for performance measures in simulations. For one mean problem, normal approximation and bootstrap methods are equal in actual probability coverages and computations involved. However, only bootstrap methods can capture the skewness in the underlying distribution. Either BC method or BCa method can be recommended in place of the normal approximation. Computations in crude bootstrap procedure are intensive but manageable. Compared to Jackknife method, crude bootstrap appears to be the only method, which produces reasonable probability coverages for the ratio of means in the regenerative process.

References

[1] Crane, M.A. and Iglehart, D.L. Simulating stable stochastic systems iii : regenerative processes and discrete event simulations. *Operations Research*, 23:33–45, 1975.

[2] Efron, B. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82:171–185, 1987.

[3] Efron, B. The jackknife, the bootstrap, and other resampling plans. *CBMS-NSF Regional Confidence Series in Applied Mathematics*, 38:5–11, 1982.

[4] Law, A.M. Statistical analysis of the output data from terminating simulations. *Naval Research Logistics Quarterly*, 27:131–143, 1980.

[5] Law, A.M. and Kelton, W.D. *Simulation Modeling and Analysis*. McGraw-Hill, 1982.

Relational Databases: A Tutorial for Statisticians

JOE R. HILL
EDS Research, 5951 Jefferson St. NE, Albuquerque, NM 87109

Abstract

This tutorial links relational database concepts to probability concepts. For example, the fundamental relational database concepts of an attribute (column heading), a relation scheme (unpopulated table), and a relation (populated table) correspond respectively to the probability concepts of a random variable, a random vector, and a multivariate probability distribution. The relational select and project operators correspond respectively to finding a conditional and marginal distribution. Functional dependencies, multivalued dependencies, and join dependencies correspond respectively to variable transformations, conditional independencies, and more general factorizations of distributions. These connections indicate that statisticians may know more about relational databases than they realize. Beyond these pedagogical benefits, these connections between relational databases and statistics provide a bridge, both directions of which have proven to be useful for developing new theory.

1 Introduction

This tutorial will cover:

- Relational database concepts and probability parallels (Section 2).
- An introduction to database normalization theory (Section 3).
- Parallel theorems for consistent databases and consistent sets of marginal distributions (Section 4).
- Finding closures of sets of multivalued dependencies and sets of conditional independencies (Section 5).
- Eliminating intersection anomalies in sets of conditional independencies and sets of multivalued dependencies (Section 6).
- Concluding remarks (Section 7).

This tutorial will not cover

- Anything about particular relational database management systems.
- Network, hierarchical, or object-oriented database models.
- Distributed databases.

Basic references for relational databases include Codd (1970), Date (1986), Maier (1983), and Ullman (1982). More advanced references include Fagin (1977), Fagin, Mendelzon & Ullman (1982), Beeri, Fagin, Maier & Yannakakis (1983), and Beeri & Kifer (1986a, b, 1987). Connections to probability theory are mentioned in Pearl (1988), Geiger & Pearl (1988, 1990), Geiger, Paz & Pearl (1991), Lauritzen & Spiegelhalter (1988), and Thoma (1989).

2 Database Concepts and Probability Parallels

This section defines the basic database concepts and the parallel probability concepts. The definitions are given in parallel because familiarity with the probability concepts might help the reader understand the essential ideas underlying the database concepts. Also, as sections 4, 5, and 6 show, there are parallel problems and results in the two fields.

A relation scheme (table skeleton) R is a set of attributes (column headings). A relation (table) over relation scheme R is an indicator function for a set of tuples (rows), written $r[R]$: $r[R](t) = 1$ if the tuple t is in the relation; $r[R](t) = 0$ if t is not in the relation. When storing or writing out a relation, it is common to list only those tuples that are in the relation (i.e. that have $r[R](t) = 1$).

The parallel concepts in probability theory are a random vector and a probability distribution. A random vector V is a set of random variables. A distribution for the random vector V is a probability function, written $p[V]$. The distribution of V evaluated at v is written $p[V](v)$.

AIRLINE EXAMPLE (Maier, 1983). Relation *schedule* contains scheduling information for an airline. Relation *schedule* is defined over the relation scheme with attributes FLT, FROM, TO, DEP, and ARR. The first tuple in *schedule*, t_1, maps FLT into 84, FROM into O'Hare, and DEP into 3:00pm. The projection of t_1 onto {FROM, TO} is t_1[FROM, TO] = (O'Hare, JFK).

schedule

FLT	FROM	TO	DEP	ARR
84	O'Hare	JFK	3:00pm	5:55pm
109	JFK	Los Angeles	9:40pm	2:42am
117	Atlanta	Boston	10:05pm	12:43am
213	JFK	Boston	11:43am	12:45pm
214	Boston	JFK	2:20pm	3:12pm

The basic operators on relations are a projection of a relation onto a subset of its attributes, a selection from a relation of the tuples having a specific value for a subset of its attributes, and a join of two relations. These operators correspond to a marginal distribution, a conditional distribution, and a product of two functions.

The projection of the relation $r[R]$ onto $X \subseteq R$, written $r[X]$ or $\pi_X(r[R])$, is the indicator function: $r[X](x) = 1$ if there is a tuple t such that $r[R](t) = 1$ and $t[X] = x$; $r[X](x) = 0$ otherwise.

The marginal distribution of $X \subseteq V$ based on $p[V]$, written $p[X]$, is found by summing $p[V]$ over the variables not in X; that is, letting $Y = V - X$,

$$p[X](x) = \sum_y p[XY](x, y).$$

AIRLINE EXAMPLE. The following tables show the projections of *schedule* onto {DEP, ARR} and onto FROM.

$\pi_{\text{DEP, ARR}}(schedule)$

DEP	ARR
3:00pm	5:55pm
9:40pm	2:42am
10:05pm	12:43am
11:43am	12:45pm
2:20pm	3:12pm

$\pi_{\text{FROM}}(schedule)$

FROM
O'Hare
JFK
Atlanta
Boston

The selection from the relation $r[R]$ of the tuples with $X = x$, $X \subseteq R$, written $r[R \mid X = x]$ or $\sigma_{X=x}(r[R])$, is the indicator function: $r[R \mid X = x](t) = 1$ if $t[X] = x$; $r[R \mid X = x](t) = 0$ otherwise. The Y-projection of the $X = x$ selection from $r[R]$ is written $r[Y \mid X = x]$.

The conditional distribution of V given $X = x$, $X \subseteq V$, based on $p[V]$, written $p[V \mid X = x]$, is the probability function:

$$p[V \mid X = x](v) = p[V](v)/p[X](x)$$

if $p[X](x) > 0$ and $v[X] = x$; $p[V \mid X = x](v) = 0$ otherwise. The Y-margin of the $X = x$ conditional is written $p[Y \mid X = x]$.

AIRLINE EXAMPLE. The following table shows the data for flights from JFK.

$\sigma_{\text{FROM=JFK}}(schedule)$

FLT	FROM	TO	DEP	ARR
109	JFK	Los Angeles	9:40pm	2:52am
213	JFK	Boston	11:43am	12:45pm

Let $r_1[R_1]$ and $r_2[R_2]$ be relations over relation schemes R_1 and R_2. Let $X = R_1 - R_2$, $Y = R_1 \cap R_2$, $Z = R_2 - R_1$. The join of $r_1[R_1]$ and $r_2[R_2]$ is the relation over $R_1 \cup R_2 = XYZ$ (XYZ is shorthand for $X \cup Y \cup Z$) defined by

$$(r_1 \bowtie r_2)[XYZ](x, y, z) = r_1[XY](x, y)\ r_2[YZ](y, z).$$

Let $h_1[V_1]$ and $h_2[V_2]$ be functions over variable sets V_1 and V_2. Let $X = V_1 - V_2$, $Y = V_1 \cap V_2$, $Z = V_2 - V_1$. The product of $h_1[V_1]$ and $h_2[V_2]$ is the function over $V_1 \cup V_2 = XYZ$ defined by

$$(h_1 \otimes h_2)[XYZ](x, y, z) = h_1[XY](x, y)\ h_2[YZ](y, z).$$

AIRLINE EXAMPLE. Relation *usable* contains the equipment requirements for each flight. Relation *certified* contains the equipment qualifications for each pilot. Suppose we want to know the pilots that can fly each of the flights. To find the answer to this query, we first form *options* = *usable* $\bowtie$ *certified*. Then we project *options* onto FLT and PILOT, providing the answer to the original query.

usable

FLT	EQPMT
83	727
83	747
84	727
84	747
109	707

certified

PILOT	EQPMT
Simmons	707
Simmons	727
Barth	747
Hill	727
Hill	747

options = *usable* ⋈ *certified*

FLT	EQPMT	PILOT
83	727	Simmons
83	727	Hill
83	747	Barth
83	747	Hill
84	727	Simmons
84	727	Hill
84	747	Barth
84	747	Hill
109	707	Simmons

$\pi_{\text{FLT, PILOT}}(options)$

FLT	PILOT
83	Simmons
83	Hill
83	Barth
84	Simmons
84	Hill
84	Barth
109	Simmons

Table 1 summarizes the basic database/probability parallels covered to this point.

Database design concepts involve putting constraints on the data that can populate a table. There are three basic kinds of constraints: a functional dependency, a multivalued dependency, and a join dependency. These correspond to three constraints on probablity distributions: *transformation constraints*, *conditional independencies*, and general factorization constraints.

A relation $r[R]$ satisfies the functional dependency FD: $X \rightarrow Y$ if for each X-value x with $r[X](x) = 1$, there is a unique Y-value y_x such that $r[Y \mid X = x](y) = 1$ if $y = y_x$ and $r[Y \mid X = x](y) = 0$ otherwise.

A distribution $p[V]$ satisfies the transformation constraint TC: $X \rightarrow Y$ if for each X-value x with $p[X](x) > 0$, there is a unique Y-value y_x such that $p[Y \mid X = x](y) = 1$ if $y = y_x$ and $p[Y \mid X = x](y) = 0$ otherwise.

AIRLINE EXAMPLE. The relation *schedule* satisfies the FD FLT → {FROM, TO, DEP, ARR}. The FLT-value of a tuple uniquely determines the rest of the tuple. The relation *schedule* does not satisfy the FD FROM → TO because t_2[FROM] = t_4[FROM] = JFK, but t_2[TO] = Los Angeles ≠ Boston = t_4[TO].

A random vector V satisfies a constraint if all distributions for V must satisfy the constraint. Likewise, a relation scheme R satisfies a data dependency if all relations over R must satisfy the dependency.

Table 1: Basic database and probability parallels.

DATABASE CONCEPT	PROBABILITY CONCEPT
Relation scheme (table skeleton) R, a set of attributes (column names)	Random vector V, a set of random variables
Relation (table) over R, $r[R]$, an indicator function for a set of tuples (rows)	Distribution for V, $p[V]$, a probability function
Projection of $r[R]$ onto $X \subseteq R$, $\pi_X(r[R])$, or $r[X]$	Marginal distribution of $X \subseteq V$, $p[X]$
Selection $\sigma_{X=x}(r[R])$, or $r[R \mid X = x]$, $X \subseteq R$	Conditional distribution $p[V \mid X = x]$, $X \subseteq V$
Join of 2 relations $r_1[R_1] \bowtie r_2[R_2]$	Product of 2 functions $h_1[V_1] \otimes h_2[V_2]$

AIRLINE EXAMPLE. The functional dependency FLT → {FROM, TO, DEP, ARR} remains true over time. As a result, FLT is a candidate key for the relation *schedule*.

A relation $r[R]$ satisfies the multivalued dependency MVD: $Z \twoheadrightarrow X \mid Y$ if

$$r[XYZ](x_1, y_1, z)\ r[XYZ](x_2, y_2, z)$$
$$= r[XYZ](x_1, y_2, z)\ r[XYZ](x_2, y_1, z).$$

Similarly, a distribution $p[V]$ satisfies the conditional independency CI: $X \perp\!\!\!\perp Y \mid Z$ if

$$p[XYZ](x_1, y_1, z)\ p[XYZ](x_2, y_2, z)$$
$$= p[XYZ](x_1, y_2, z)\ p[XYZ](x_2, y_1, z).$$

Multivalued dependencies are equivalent to binary join dependencies. That is, a relation satisfies an MVD iff

it can be recovered as the join of two relations defined on "smaller" relation schemes. In symbols, a relation $r[R]$ satisfies MVD: $Z \twoheadrightarrow X \mid Y$ iff there exist relations $r_1[XZ]$ and $r_2[YZ]$ such that $r = r_1 \bowtie r_2$: that is,

$$r[XYZ](x, y, z) = r_1[XZ](x, z)\ r_2[YZ](y, z).$$

If such r_j's exist, then $r[R]$ is said to satisfy the binary join dependency BJD: $\bowtie \{XZ, YZ\}$. Also, if such r_j's exist, then they can be taken to be $r_1[XZ] = r[XZ]$ and $r_2[YZ] = r[YZ]$.

Similarly, conditional independencies are equivalent to binary factorization constraints. That is, a probability distribution satisfies a CI iff it can be recovered as the product of two functions defined on "smaller" random vectors. In symbols, a distribution $p[V]$ satisfies CI: $X \perp\!\!\!\perp Y \mid Z$ iff there exist nonnegative functions $h_1[XZ]$ and $h_2[YZ]$ such that $p = h_1 \otimes h_2$: that is,

$$p[XYZ](x, y, z) = h_1[XZ](x, z)\ h_2[YZ](y, z).$$

If such h_j's exist, then $p[V]$ is said to satisfy the binary factorization constraint BFC: $\otimes\{XZ, YZ\}$.

AIRLINE EXAMPLE. Relation *service* satisfies the MVD: FLT$\twoheadrightarrow$DAY OF WEEK | PLANE TYPE because *service* = *servday* $\bowtie$ *servtype* where *servday* = *service*[FLT, DAY OF WEEK] and *servtype* = *service*[FLT, PLANE TYPE]. Relation *service*2, which has the same two projections as *service*, does not satisfy this MVD because it lacks the tuple (106, Thursday, 1011).

service

FLT	DAY OF WEEK	PLANE TYPE
106	Monday	747
106	Thursday	747
106	Monday	1011
106	Thursday	1011
204	Wednesday	707
204	Wednesday	727

A relation $r[R]$ satisfies the join dependency JD: $\bowtie \mathcal{R}$, $\mathcal{R} = \{R_1, ..., R_k\}$, $R_j \subseteq R$, if there exist relations $r_1[R_1]$, ..., $r_k[R_k]$ such that

$$r[R] = r_1[R_1] \bowtie \cdots \bowtie r_k[R_k].$$

If such r_j's exist, then they can be taken to be $r_j[R_j] = r[R_j], j = 1, ..., k$. The set of relation schemes $\mathcal{R}$ is a set of subsets of R; in other words, $\mathcal{R}$ is a hypergraph over R.

servday

$\pi_{\text{FLT, DAY OF WEEK}}(service)$

FLT	DAY OF WEEK
106	Monday
106	Thursday
204	Wednesday

servtype

$= \pi_{\text{FLT, PLANE TYPE}}(service)$

FLT	PLANE TYPE
106	747
106	1011
204	707
204	727

*service*2

FLT	DAY OF WEEK	PLANE TYPE
106	Monday	747
106	Thursday	747
106	Monday	1011
204	Wednesday	707
204	Wednesday	727

A distribution $p[V]$ satisfies the factorization constraint FC: $\otimes\mathcal{V}$, $\mathcal{V} = \{V_1, ..., V_k\}$, $V_j \subseteq V$, if there exist nonnegative functions $h_1[V_1], ..., h_k[V_k]$ such that

$$p[V] = h_1[V_1] \otimes \cdots \otimes h_k[V_k].$$

The set of margins $\mathcal{V}$ is a hypergraph over V. Factorization constraints generalize loglinear models which must be strictly positive.

EXAMPLE. The relation $r[ABC]$ satisfies the JD: $\bowtie \{AB, BC, AC\}$ but does not satisfy any nontrivial MVD.

$r[ABC]$			$r[AB]$		$r[BC]$		$r[AC]$	
A	B	C	A	B	B	C	A	C
1	1	1	1	1	1	1	1	1
1	2	2	1	2	2	2	1	2
2	3	3	2	3	3	3	2	3
3	3	4	3	3	3	4	3	4
4	4	5	4	4	4	5	4	5
5	5	5	5	5	5	5	5	5

Finally, databases correspond to sets of marginal distributions.

A database scheme over attribute set R is a set of relation schemes with attributes from R: $\mathcal{R} = \{R_1, ..., R_k\}$, $R_j \subseteq R$. The database scheme $\mathcal{R}$ is a hypergraph over R. A database over database scheme $\mathcal{R}$ is a set of relations over the relation schemes in $\mathcal{R}$: $\mathbf{r}[\mathcal{R}] = \{r_1[R_1], ..., r_k[R_k]\}$.

A set of margins of random vector V is a set of random vectors with variables from V: $\mathcal{V} = \{V_1, ..., V_k\}$, $V_j \subseteq V$. The set of margins $\mathcal{V}$ is a hypergraph over V. A set of marginals over set of margins $\mathcal{V}$ is a set of distributions for the margins in $\mathcal{V}$: $\mathbf{p}[\mathcal{V}] = \{p_1[V_1], ..., p_k[V_k]\}$.

Table 2 summarizes this second collection of parallels. There are many more parallels between database theory and probability. Sections 4, 5, and 6 discuss, very briefly, three parallel problems and solutions.

Table 2: Further database and probability parallels.

DATABASE CONCEPT	PROBABILITY CONCEPT
Functional dependency $X \rightarrow Y$ $X, Y \subseteq R$	Variable transformation $X \rightarrow Y$ $X, Y \subseteq V$
Multivalued dependency $Z \twoheadrightarrow X \mid Y$ $X, Y, Z \subseteq R$	Conditional independency $X \perp\!\!\!\perp Y \mid Z$ $X, Y, Z \subseteq V$
Join dependency $\bowtie \mathcal{R}$ $\mathcal{R} = \{R_1, ..., R_k\}$, $R_j \subseteq R$ MVDs are binary JDs $\bowtie \{XZ, YZ\}$	Factorization constraint $\otimes \mathcal{V}$ $\mathcal{V} = \{V_1, ..., V_k\}$, $V_j \subseteq V$ CIs are binary FCs $\otimes \{XZ, YZ\}$
Database scheme over R $\mathcal{R} = \{R_1, ..., R_k\}$, $R_j \subseteq R$	Set of margins of V $\mathcal{V} = \{V_1, ..., V_k\}$, $V_j \subseteq V$
Database over $\mathcal{R}$ $\mathbf{r}[\mathcal{R}] = \{r_1[R_1], ..., r_k[R_k]\}$	Set of marginals on $\mathcal{V}$ $\mathbf{p}[\mathcal{V}] = \{p_1[V_1], ..., p_k[V_k]\}$

3 A Brief Introduction to Normalization Theory

Here is a very tiny bit of normalization theory, an important standard topic in database theory with no useful parallels in probability theory. The basic reason for normalizing a database is to automatically eliminate possible inconsistencies that might otherwise arise.

A set of attributes K is a candidate key of R if $K \rightarrow R$. One of the candidate keys of relation R is designated the primary key and the other attributes are called non-keys.

A set of attributes Y is fully dependent on another set of attributes X if $X \rightarrow Y$ and there is no $Z \subset X$ such that $Z \rightarrow Y$. If there is such a Z then Y is partially dependent on X.

A set of attributes Z is transitively dependent on X if there is a Y such that $X \rightarrow Y$ and $Y \rightarrow Z$.

The normal forms are:

- First Normal Form (1NF): A relation is in 1NF if all the values in its tuples are atomic. There are no repeating groups.
- Second Normal Form (2NF): A relation is in 2NF if it is in 1NF and every non-key is fully dependent on the primary key. A relation in 2NF has no partial dependencies.
- Third Normal Form (3NF): A relation is in 3NF if it is in 2NF and no non-key is transitively dependent on the primary key. A relation in 3NF has no partial or transitive dependencies. All the non-keys in a 3NF relation are mutually independent (i.e. no nonkey is functionally dependent on another nonkey).
- Boyce/Codd Normal Form (BCNF): A relation is in BCNF if every FD is a consequence of the candidate keys. Date: "Each field must represent a fact about the key, the whole key, and nothing but the key."
- Fourth Normal Form (4NF): A relation is in 4NF if every MVD is a consequence of the candidate keys. All dependencies (MVDs and FDs) of a 4NF relation are FDs from a candidate key to another attribute. A relation is in 4NF if it is in BCNF and all its MVDs are FDs.
- Fifth Normal Form (5NF): A relation is in 5NF if every JD is a consequence of the candidate keys. 5NF is also called project/join normal form.

There are rules for converting database schemes that do not satisfy normal forms into ones that do. The interested reader should consult Maier (1983) or Ullman (1982), for example.

4 Parallel Theorems for Consistent Databases and Consistent Sets of Marginal Distributions

It was noted earlier that database schemes and sets of margins are hypergraphs. There are strong connections between relational databases and graph theory and between probability theory and graph theory. Often, properties of databases and properties of probability distributions are determined by the underlying graphical structure. This section gives an example of the kind of parallel results that arise because of these connections to graph theory.

A database $\mathbf{r}[\mathcal{R}]$ is pairwise consistent if $r_i[R_i \cap R_j] = r_j[R_i \cap R_j]$. A database $\mathbf{r}[\mathcal{R}]$ is globally consistent if there exists a single relation $r[R]$ such that $r_j[R_j] = r[R_j]$; if such an $r[R]$ exists, then it can be taken to be $r[R] = r_1[R_1] \bowtie \cdots \bowtie r_k[R_k]$.

A set of marginals $\mathbf{p}[\mathcal{V}]$ is pairwise consistent if $p_i[V_i \cap V_j] = p_j[V_i \cap V_j]$. A set of marginals $\mathbf{p}[\mathcal{V}]$ is globally consistent (or extendable) if there exists a single distribution $p[V]$ such that $p_j[V_j] = p[V_j]$.

Consider the following two examples.

EXAMPLE 1 (Vorob'ev, 1962). Let $\mathcal{V} = \{AB, BC, AC\}$ be a set of margins of the random vector ABC. Let $\mathbf{p} = \{p_1, p_2, p_3\}$ be the set of marginals over $\mathcal{V}$ defined by

$$p_1[AB](0,0) = p_1[AB](1,1) = 1/2,$$

$$p_2[BC](1,0) = p_2[BC](0,1) = 1/2,$$

and

$$p_3[AC](0,0) = p_3[AC](1,1) = 1/2.$$

There is no distribution $p[ABC]$ such that $p[AB] = p_1[AB]$, $p[BC] = p_2[BC]$, and $p[AC] = p_3[AC]$. Such a $p[ABC]$ would have $p[ABC](0,0,0) = 0$ because $p_2[BC](0,0) = 0$, and $p[ABC](0,0,1) = 0$ because $p_3[AC](0,1) = 0$, so $p[AB](0,0) = 0$, contradicting $p_1[AB] = 1/2$. This same example can be given as a database example with p's replaced by r's and $1/2$'s replaced by 1's.

EXAMPLE 2. Let $\mathcal{R} = \{ABD, BCD, BCE\}$ be a database scheme over $ABCDE$. For every pairwise consistent database $\mathbf{r} = \{r_1, r_2, r_3\}$ over $\mathcal{R}$, there is a single relation $r[ABCDE]$ such that $r[ABD] = r_1[ABD]$, $r[BCD] = r_2[BCD]$, and $r[BCE] = r_3[BCE]$. The parallel statement holds for probability distributions.

The difference between these examples is that the hypergraph in Example 2 is acyclic but the one in Example 1 is not acyclic. There are many ways to define an acyclic hypergraph. The following definition, referred to as the running intersection property, does not require definitions for any other concepts. A hypergraph $\mathcal{H}$ is acyclic if its elements can be ordered so that for each $i = 2, \ldots, k$, there is a $j < i$ with

$$H_i \cap (H_1 \cup \cdots \cup H_{i-1}) \subseteq H_j.$$

The two results can now be stated.

Vorob'ev (1962) proved that every pairwise consistent set of marginals over a set of margins $\mathcal{V}$ is extendable if and only if the hypergraph $\mathcal{V}$ is acyclic (see also Lauritzen, Speed & Vijayan, 1984).

Beeri, Fagin, Maier & Yannakakis (1983) proved the parallel result for relational databases: that is, every pairwise consistent database over a database scheme $\mathcal{R}$ is globally consistent if and only if the hypergraph $\mathcal{R}$ is acyclic.

5 Closures of Sets of MVDs and Sets of CIs

Let $\mathbf{M}$ be a set of MVDs over R. The closure $\mathbf{M}^*$ of $\mathbf{M}$ is the set of MVDs implied by the MVDs in $\mathbf{M}$, that is, if a relation satisfies the MVDs in $\mathbf{M}$, then it also satisfies the MVDs in $\mathbf{M}^*$.

The closure $\mathbf{M}^*$ of $\mathbf{M}$ can be found as follows. Let $\Sigma_{\mathbf{M}}(X) = \{Y \subseteq R - X : X \twoheadrightarrow Y \in \mathbf{M}^*\}$. The dependency basis of X, $\mathrm{DEP}_{\mathbf{M}}(X)$, is the partition of $R - X$ such that $Y \in \Sigma_{\mathbf{M}}(X)$ iff Y is a union of sets in $\mathrm{DEP}_{\mathbf{M}}(X)$. $\mathrm{DEP}_{\mathbf{M}}(X)$ can be found using the following algorithm:

(0) Start with partition $\mathcal{P} = \{V - X\}$.

(1) If $Y \in \mathcal{P}$ and there is an MVD: $Z \twoheadrightarrow W$ in $\mathbf{M}$ such that $Y \cap Z = \emptyset$, then replace Y by the 2 sets $Y \cap W$ and $Y - W$.

(2) Repeat (1) until it no longer changes $\mathcal{P}$.

The final partition is $\mathrm{DEP}_{\mathbf{M}}(X)$.

EXAMPLE. Let $\mathbf{M} = \{BC \twoheadrightarrow AD \mid E,\ BD \twoheadrightarrow A \mid CE\}$. To find $\mathrm{DEP}_{\mathbf{M}}(BCD)$, (0) let $\mathcal{P} = \{AE\}$, (1) $AE \in \mathcal{P}$, $BC \twoheadrightarrow AD \mid E \in \mathbf{M}$, and $BC \cap AE = \emptyset$, so replacing AE by $AE \cap AD = A$ and $AE \cap E = E$ gives $\mathrm{DEP}_{\mathbf{M}}(BCD) = \{A, E\}$.

Geiger & Pearl (1988, 1990) and Geiger, Paz & Pearl (1991) proved that the same algorithm can be used to find the closure of a set of conditional independencies with respect to arbitrary (i.e. not necessarily strictly positive) distributions. They also derived a graph-based approach for finding the closure with respect to strictly positive distributions.

6 Eliminating Intersection Anomalies

The two CIs $X \perp\!\!\!\perp Y \mid Z$ and $X \perp\!\!\!\perp Z \mid Y$ imply the third $X \perp\!\!\!\perp YZ$ for strictly positive distributions. The same is not true for arbitrary distributions. For example, the distribution $p[XYZ](0,0,0) = p[XYZ](1,1,1) = 1/2$, $p[XYZ](x,y,z) = 0$ otherwise, satisfies the first two of these CIs, but does not satisfy the third. The set of CIs $\{X \perp\!\!\!\perp Y \mid Z,\ X \perp\!\!\!\perp Z \mid Y\}$ is said to have an intersection anomaly.

After reviewing several statistical arguments that were flawed because they ignored intersection anomalies, Dawid (1979) showed that it is possible to fix up this anomaly by adding a variable W such that W is functionally determined by each of Y and Z individually (i.e. $Y \rightarrow W$, $Z \rightarrow W$) and $X \perp\!\!\!\perp YZ \mid W$. The variable W represents the information that Y and Z have in common.

Beeri and Kifer (1986a, b, 1987) and others have written extensively about the same issue for sets of MVDs. Their solution, which has implications for database design, is the same as Dawid's. They only apply the method to sets of MVDs that do not have split left hand sides, so after eliminating intersection anomalies they have a conflict-free set of MVDs which is equivalent to a single (acyclic) JD.

7 Concluding Remarks

This tutorial reviewed basic parallels between database theory and probability theory. It discussed three parallel problems and corresponding solutions in the two areas. It mentioned some of the connections to graph theory which provide another bridge between results in database theory and those in probability theory. For example, acyclic databases and decomposable models (distributions that satisfy acyclic factorization constraints) have many desirable properties (Beeri, Fagin, Maier & Yannakakis, 1983; Darroch, Lauritzen & Speed, 1980).

One particularly interesting connection concerns the positivity condition of the Gibbs-Markov equivalence theorem. It is possible to relax the positivity condition using concepts from relational database theory. Results on this topic and others will be given in future papers.

Acknowledgements

I thank Mathis Thoma and Dan Geiger for many insightful discussions on the connections between database theory, probability theory, and graph theory. I also thank Jon Kettenring for encouraging me to give the tutorial.

References

BEERI, C., FAGIN, R., MAIER, D. & YANNAKAKIS, M. (1983). On the desirability of acyclic database schemes. *Journal of the Association for Computing Machines* **30**, 479-513.

BEERI, C. & KIFER, M. (1986a). An integrated approach to logical database design of relational database schemes. *ACM Transactions on Database Systems* **11**, 134-158.

BEERI, C. & KIFER, M. (1986b). Elimination of intersection anomalies from database schemes. *Journal of the Association for Computing Machinery* **33**, 423-450.

BEERI, C. & KIFER, M. (1987). A theory of intersection anomalies in relational database schemes. *Journal of the Association for Computing Machinery* **34**, 544-577

CODD, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM* **13**, 377-387.

DARROCH, J. N., LAURITZEN, S. L. & SPEED, T. P. (1980). Markov fileds and loglinear models for contingency tables. *Annals of Statistics* **8**, 522-539.

DATE, C. J. (1986). *An Introduction to Database Systems, Volume I, Fourth Edition*. Addison-Wesley, Reading, Massachusetts.

DAWID, A. P. (1979). Some misleading arguments involving conditional independence. *Journal of the Royal Statistical Society, Series B* **41**, 249-252.

FAGIN, R. (1977). Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems* **2**, 262-278.

FAGIN, R., MENDELZON, A. O. & ULLMAN, J. F. (1982). A simplified universal relation assumption and its properties. *ACM Transactions on Database Systems* **7**, 343-360.

GEIGER, D., PAZ, A. & PEARL, J. (1988). Axioms and algorithms for inferences involving probabilistic independence. *Information and Computation* **1**, 128-141.

GEIGER, D. & PEARL, J. (1988). Logical and algorithmic properties of conditinal independence and qualitative independence. UCLA, Cognitive Systems Laboratory, Technical Report R-97.

GEIGER, D. & PEARL, J. (1990). Logical and algorithmic properties of independence and their applications to Bayesian networks. *Annals of Mathematics and Artificial Intelligence* **2**, 165-178.

LAURITZEN, S. L., SPEED, T. P. & VIJAYAN, K. (1984). Decomposable graphs and hypergraphs. *Journal of the Australian Mathematical Society, Series A* **36**, 12-29.

LAURITZEN, S. L. & SPIEGELHALTER, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society, Series B* **50**, 157-224.

MAIER, D. (1983). *The Theory of Relational Databases*. Computer Science Press, Rockville, Maryland.

PEARL, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California.

THOMA, M. (1989).*Factorization of Belief Functions*. Harvard Ph.D. thesis, Department of Statistics.

ULLMAN, J. D. (1982). *Principles of Database Systems, Second Edition*. Computer Science Press, Rockville, Maryland.

VOROB'EV, N. N. (1962). Consistent families of measures and their extensions. *Theory of Probability and its Applications* **7**, 147-163.

Mixing Parameter Regression Applied to Groundwater Contaminant Flow

Rose Ray
Failure Analysis Associates
2225 East Bayshore Road
P.O. Box 1470
Palo Alto, California 94303

Michael E. Tarter
Michael D. Lock
Department of Biomedical and
Environmental Health Sciences
University of California
Berkeley, California 94720

Abstract

A new form of regression is applied to the problem of modeling the flow of water and contaminants through soil. In a fashion analogous to nested ANOVA, the new method parametrizes global distributional structure separately from local structure. A blind study is conducted to assess the precision of mixing parameter estimation as a function of depth. It is shown that accurate estimates of the regression relationship can be obtained from a sample of size n=1000 for mixing parameters and all other component parameters, with the exception of the standard deviation of small components which have large variances.

It is shown that the hydraulic conductivity, transport, or infiltration of water borne contaminants through the vadose zone can be effectively modeled and simulated by the mixing parameter regression methods.

Research Supported by National Institute of Environmental Health Sciences Grant 1 RO1 ES 053479-01. The authors would like to thank C. Mellin for many useful comments and for guiding the many stages of this project to completion.

KEY WORDS: Hydraulic conductivity; Mixture decomposition; Nonparametric estimation; Nonparametric regression; Switching regression; Vadose zone infiltration.

1. Introduction

This paper concerns the description of soil characteristics by means of a new type of regression. The value of this form of regression stems from its capacity to separate global from local variability through the use of interactive graphical analysis.

As discussed by Wagenet (1986, p. 340) :

> It appears that a stochastic, rather than a deterministic, model approach should be considered when modeling water and chemical movement in the unsaturated zone. This will represent no small change in our conceptualization of basic principles of pesticide modeling. The resulting models will almost certainly not represent basic processes in fundamental mechanistic terms, but will instead will represent the soil-water-pesticide system in statistical terms.

In line with Wagenet's assertion we propose to simultaneously describe global and local variability by means of mixing parameter regression.

Soil can be considered to be both a mixture in the chemical and in the statistical sense. However, unlike the uniformity inherent in the molecules of compounds the substances such as clays, sands, pebbles and cobbles which are described below do not have uniform characteristics. Rather than uniformity, there is a degree of variability in hydraulic conductivity, density and pore size, as well as in many other characteristics of any given substance, which is comparable to the obvious variability between substances.

For the stochastic and other models with which water-borne contaminant flows are modeled, it is of great value to parametrize the between-substance variability separately from the within substance variability. In the two-substance case, consider mixture model (1) of the conditional probability density $f(y \mid x)$ of flow variate value $Y=y$ at a given value x of key variate X:

$$P(x)\, f_1\{[y-\mu_1(x)]/\sigma_1(x)\}/\sigma_1(x) + \{1-P(x)\}f_2\{[y-\mu_2(x)]/\sigma_2(x)\}/\sigma_2(x) \qquad (1)$$

where f_1 and f_2 are probability densities which are symmetric about zero , regression functions $\mu_1(x)$, $\sigma_1(x)$, $\mu_2(x)$ and $\sigma_2(x)$ describe the local substance-specific variation with value x of variate X (below we will specifically refer to x as a depth) and finally, and most importantly, mixing parameter regression function P(x) expresses the relationship between pure global variation of the Y variate and the value of the key variable X. Local variation within contiguous and homogeneous soil subregions, pockets, is described by the functions f_1 and f_2, where these functions will be assumed to be functionally independent of x. In realistic applications, there will of course be both more than two classifications of soil types and X will be vector rather scalar-valued. However, both for purposes of illustration, and because the methodology illustrated below is at the cutting edge of what is now computationally feasible, only the two component scalar case will be discussed.

Previous statistical literature which discusses mixture model regression focuses upon the relationship between the two means , $\mu_1(x)$ and $\mu_2(x)$, and x. Quandt (1958,1972), Kieffer (1978), and Quandt and Ramsey (1978) refer to this model as "switching regression". The distinction between switching regression and mixing parameter regression is central to the theme of this paper. A switching regression curve describes the overall distribution, and hence one form of variation of the distribution in its entirety. On the other hand, the mixing parameter regression function P(x) describes pure global variation. In the case of soil constituents such as sand or cobbles it quantifies the variation of a constituent in its entirety, independent of variation within the constituent itself. For example, it can be used to indicate how flow is affected by the change from the proportion of cobbles found at one depth $X=x_1$, to the proportion found at a second depth $X=x_2$. If the parameters of the cobble-specific density also change with depth, this change will affect the overall model through parameters other than P(x), specifically, $\mu_1(x)$, $\sigma_1(x)$. (Below we will use cobbles in examples of the new mixture methodology to emphasize that the material whose properties are being studied cannot always be brought to the surface and examined directly, but instead, must often be examined in situ.)

For purposes of illustration, suppose f_1 describes local variation within deposits of cobbles and, and f_2 describes local variation within deposits of sand. (Below, we will refer to the former as high density and the later as low density pockets.) It is extremely convenient to separate the estimation of the function P(x) from the estimation of the parameters which form part of f_1 's and f_2 's arguments. The stochastic models used to simulate flow processes can be systematically constructed when these three functions are considered separately. In addition, in a fashion analogous to nested analysis of variance (Fraser, 1958, pp.141-150), ANOVA, this formulation can facilitate studies of the relative importance of local (in nested ANOVA, within) versus global (in nested ANOVA, between) variation of soil characteristics for the prediction of water and contaminant flow through soil (Ray and Turk 1991).

2. Soil Configurations

Three basic types of soil configuration are shown in Figures 1a, b and c. Figure 1a depicts a distribution where small pockets of high density soil are uniformly distributed. (In terms of mixture model (1), the proportion of high density soil at depth x is

parametrized by P(x).) Because of this uniformity, the soil configuration shown in Figure 1a can be analyzed by switching regression methods.

Figure 1b shows a mixture which is similar to that shown in Figure 1a, where large pockets of high density soil are uniformly distributed. Figure 1c shows a configuration where the proportion of high density soil increases with depth variate x. Note that within each pocket, in other words, locally, hydraulic conductivity, infiltration, or movement might reasonably be assumed to have the same distribution. However, globally, the pocket-type (in other words the type of mixture subpopulation) might vary as a function of depth x. (Of course, it is also possible that other parameters besides P(x) are non-constant functions of x.)

Even though depth, as depicted in Figures 1abc, is usually treated as a vertical coordinate, below, the letter x will be used to represent a specific depth. This choice of representation was required by the convention that x represent the independent variate within a regression relationship. Consequently, in the scatter diagram and estimated density displays in this paper, the depth variate will vary horizontally, along the x-axis.

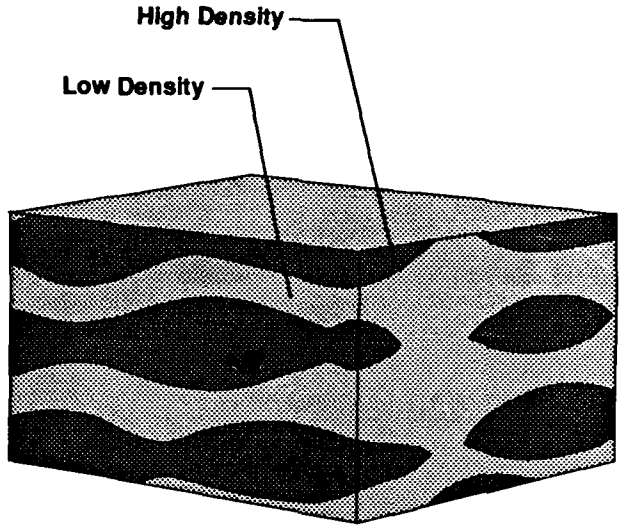


Figure 1b Mixture of two soil types: large pockets of high density soil.

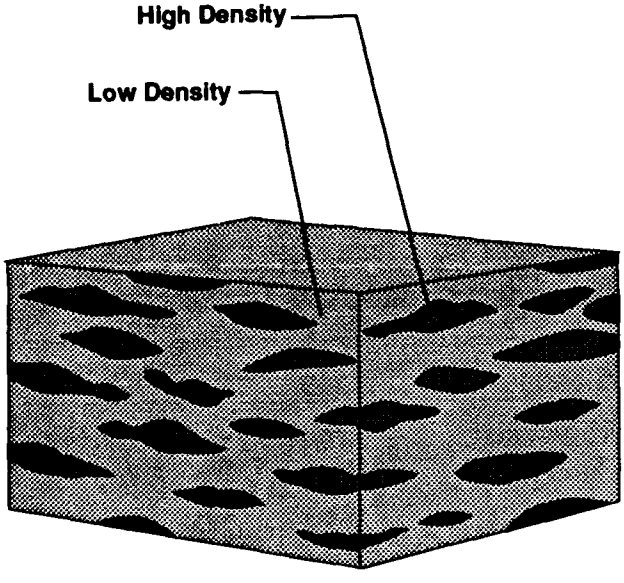


Figure 1a. Mixture of two soil types: small pockets of high density soil.

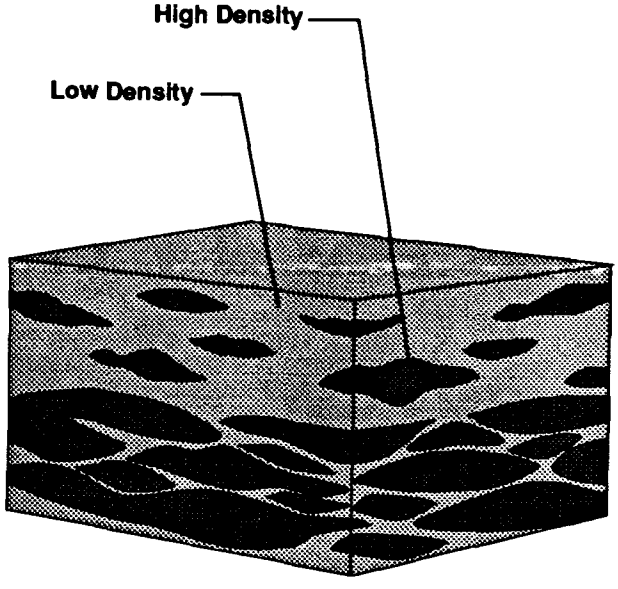


Figure 1c. Mixture of two soil types: high density soil increases with depth.

3. Theory

The methodological approach used to estimate the function P(x) is based on the distinction between mixing parameter and other forms of regression. For example, in the bivariate case what Quandt (1958,1972) refers to as switching parameter regression involves the decomposition of a mixture of two bivariate normal densities, where the constant value P(x)=p, specified how much one density contributes to the overall mixture. (For example, in Section 4.3 of Quandt and Ramsey, 1978, the mixing parameter, which these authors call "λ" is not considered to be functionally related to the of the random variate, which these authors represent by "ε.") The decomposition of bivariate normals is discussed by Tarter and Silvers (1975) and Titterington, Smith and Makov (1985) pp. 142-145. The two variate special case for which P(x)=p, and therefore P(x) is a horizontal line, is the only situation where mixing parameter regression is equivalent to the decomposition of bivariate normal densities.

In the two dimensional normal special case, mixing parameter regression is parametrized in terms of the conditional normal density and not in terms of the bivariate normal. Because it is a shorter and less technical term, below we will refer to estimated conditional densities as "slices." At a given point x, P(x) determines the proportion of a slice attributable to one component of a given two-component mixture. For example, in certain applications where x is a depth measurement, P(x) measures the proportion of one of the following list of soil components that is present within a specific soil layer: Clayey soils, sandy textured soils, soils with large pores (cobbles, large rocks, void root channels, worm holes etc..) Helling and Gish (1986). While the standard bivariate normal mixture model considered by Tarter and Silvers (1975) describes a situation where the means of the individual subpopulations of soil components change with depth, it does not describe a situation where the proportions of soil components vary with depth.

Because the conditional estimation or slicing process is central to mixing parameter regression, the crucial step of P(x) determination is the estimation of density slices. Once a slice is estimated at a depth x, this slice is then separated into its constituent components. This is accomplished by using the univariate procedures described by Kronmal (1964), by means of which a density can be estimated using a kernel transform which can reduce the overlap of mixture components. As modified by Tarter (1979a) Section 4 and described by Titterington, Smith and Makov (1985) pp. 138-140, this procedure will below be called the λ method. The following example describes these steps:

Figure 2 is a scatter diagram constructed from 1000 data points simulated as part of the study described in the next section. The dark vertical line shown in this figure is the intersection of the plane through the frequency axis, upon which Figures 3-5 are drawn, and the plane within which scatter diagram points are depicted.

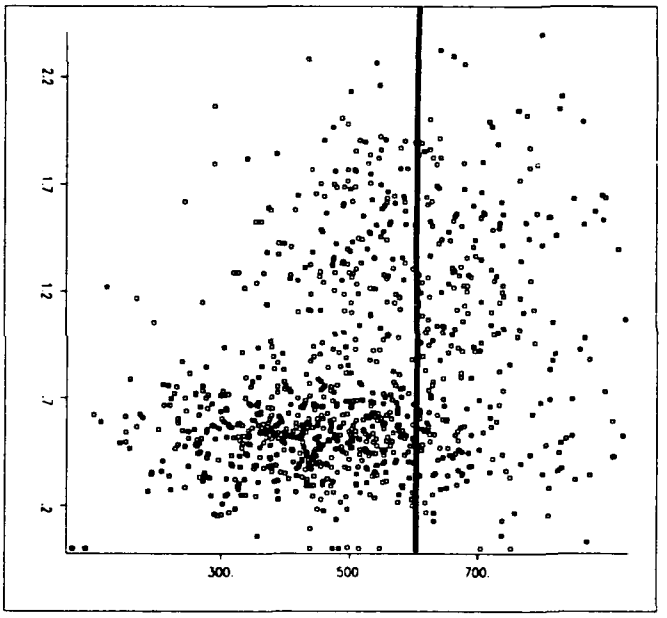


Figure 2. Scatter diagram of a 1000 point sample from the simulation experiment described in Section 3.

The first step in the process of mixing parameter regression is the estimation of the joint density of the dependent and independent variate. Methodology described in Tarter and Lock (1991) was used for this purpose. After the bivariate density is constructed, equations (2.22) and (2.23) of Tarter (1979b) were used to estimate the conditionals at a sequence of independent variate values. The slice taken at point x=600 inches through the line shown in Figure 2 is shown in Figure 3.

The spurious bumps shown at the right side of Figure 3 are due to the use of a curve estimation procedure based on fixed kernel methodology. As discussed by Tarter and Lock (1991) Section 3, methods of Breiman,

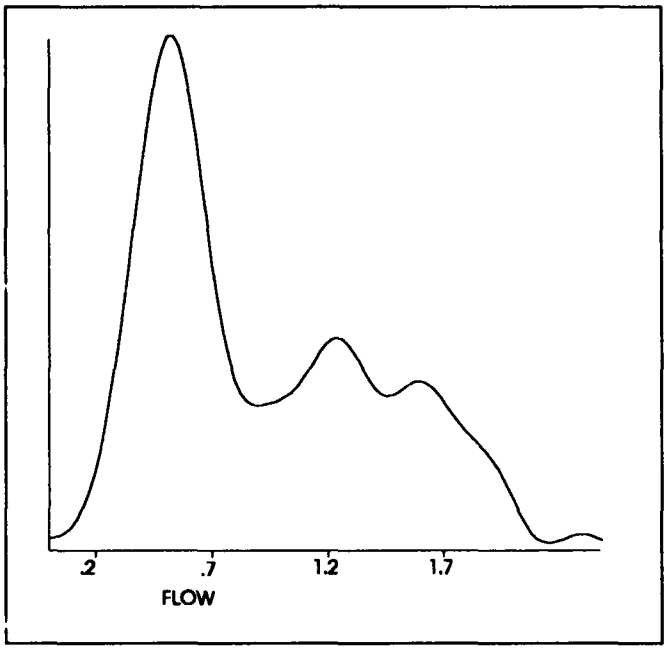


Figure 3. Estimated conditional density of the y variate in Figure 2 given the value x=600, i.e. along the solid line shown in Figure 2.

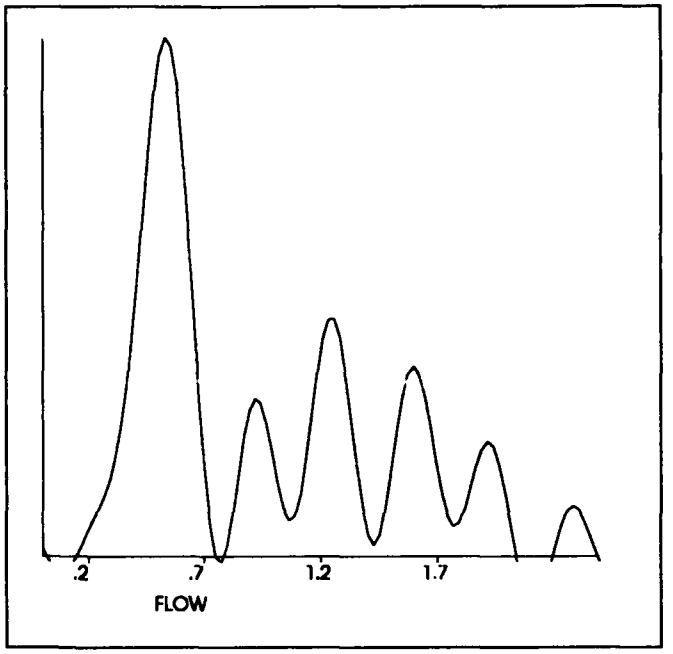


Figure 4. Estimated conditional density shown in Figure 3 with standard deviations of subcomponents reduced.

Meisel and Purcell (1977) and Tarter Silvers (1975) Expression (2.13) can be used as the basis of variable kernel techniques. However, because the transformation procedures (which we presently use to accomplish the same goals towards which variable kernel approaches are directed) would have complicated this paper, we used basic fixed kernel procedures to conduct the study described below.

Figure 4 depicts the application of the Kronmal (1964) and Titterington, Smith and Makov (1985) procedure to the curve shown in Figure 3. This involved the sample-size-controlled modification of the Fourier transform of the estimated density to obtain a curve estimate where the standard deviations of all mixture components are all reduced by a user selected constant λ.

Once the mixture-component-specific variances are reduced sufficiently so that the resulting curve has no component overlap, one or another of the components can be excised and mixing parameter P(x) can be estimated for the slice at X=x. Finally, the post-excision curve can have its component standard deviation increased by $-\lambda$ (which brings it back to the neutral setting). The effect of this step is shown by the solid curve in Figure 5. Once a component of the slice at x is isolated, the component-specific mean and variance, $\mu_1(x)$ and $\sigma_1(x)$, can be estimated. The dashed curve in Figure 5 was fit to the mixture component using the $\mu_1(x)$ and $\sigma_1(x)$ values estimated from the data shown in Figure 2. For the sample being illustrated, the only parameter which could not be accurately estimated is the standard deviation of the smaller density component, $\sigma_2(x)$.

4. Blind study

The graphical selection of the point at which components are sufficiently drawn apart for the separation step to be instituted is interactive. Consequently, the authors designed a simple single blind trial to assess the performance of the mixing parameter regression estimation procedure. One author devised and implemented simulation procedures and, independently, a second author performed the steps of the interactive parameter estimation process without any knowledge of the parameters selected for the simulated samples. Twenty different samples, where each sample corresponded to a reasonable choice of the curve P(x) as well as comparable regression functions associated with other parameters, were generated. (As a point of reference, it is the special case where P(x) is a

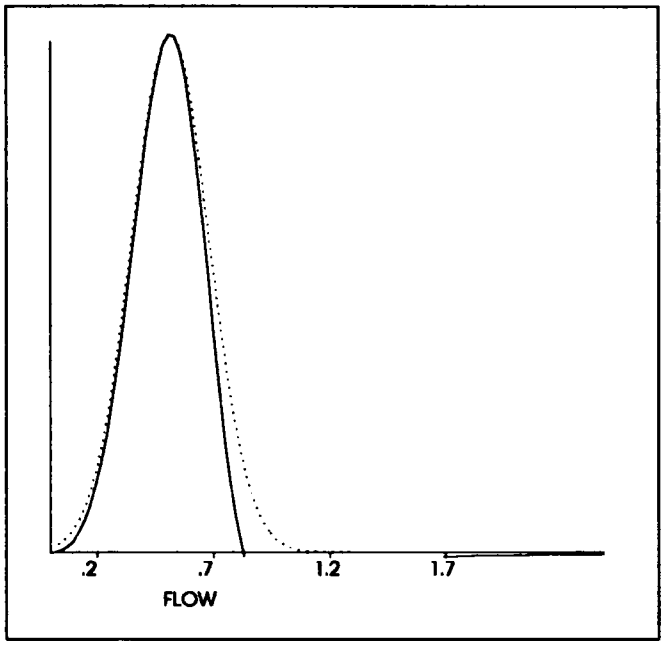


Figure 5. Leftmost component isolated from the conditional density shown in Figure 3. The dashed line is a normal curve fit to the component.

horizontal line and where the component mean functions $\mu_1(x)$ and $\mu_2(x)$ are linear, that corresponds to switching regression.)

Two forms of simulation modeling were used in the blind study. In the first set of simulations, the mixing parameter regression model was used directly. Here, models in which the mixing parameter P(x) varied as a linear function of x were used to simulate soils in which the proportion of high density soil increased directly with soil depth (as illustrated by Figure 1c). Models in which P(x) had many modes or bumps (technically, where the derivative P'(x) had many roots) were used to simulate soils of the sort that had a uniform dispersion of small pockets or lenses of high density soil. A second set of simulation experiments in which soil configurations were modeled stochastically by randomly assigning the location and size of high density soil pockets was also conducted.

Because switching regression and most mixture decomposition methods are applicable to normal data, normal random deviates were used to describe variation about parameter values. The interactive process of estimating all five regression curves P(x), $\mu_1(x)$, $\sigma_1(x)$, $\mu_2(x)$ and $\sigma_2(x)$ takes approximately twenty minutes using an IBM Personal System/2 Model 70 386 with a sample of one thousand points. The consistent performance of the estimation method in determining model parameters demonstrated the feasibility of using the mixing parameter regression model with the PC-based computational hardware which is available today. The following is a representative selection of trials, which correspond respectively to the three soil pocket configurations described in Section 2.

5. Discussion and Trials

Cressie (1988) described local and global components of environmental variation using the stochastic model $Y(x) = \mu(x) + W(x) + \varepsilon(x)$, where $\mu(.)$ is the deterministic mean structure by which large scale variation is modeled, W(.) is a zero mean intrinsically stationary process used to represent small scale variation and $\varepsilon(.)$ is a zero mean white noise process independent of W used to represent measurement error.

Universal kriging and *intrinsic random function of order k* methods have been used to separate large scale from local environmental variation. Cressie (1986) compared these methods and proposed the *median polish* method for the estimation of large scale variation or drift. These methods are based on the removal of large scale variation in order to locally predict the value of Y at a point (or small region) of contiguous x values. The estimated variogram, an estimate of the functional $E[Y_{t+h}(x) - Y_t(x)]^2 / 2$, is the resulting descriptive estimator of local variability. While the variogram targets local variability, the estimates obtained using mixing parameter regression yield separate local and global variability statistics.

Because of the above distinction it is of interest to study the behavior of the variogram estimator obtained from the examples shown in Figures 1a and 1b (which illustrate the situation where there is no systematic mean value drift) and the Figure 1c example.

Figure 6 displays the robust variogram estimates (Cressie and Hawkins, 1980) for the three examples shown in Figures 1a, 1b and 1c. The mixing parameter regression estimators for these three examples are shown in Figure 7. It is notable that the three examples are easily distinguished from one another by the mixing parameter method but, to all extents and purposes, yield indistinguishable

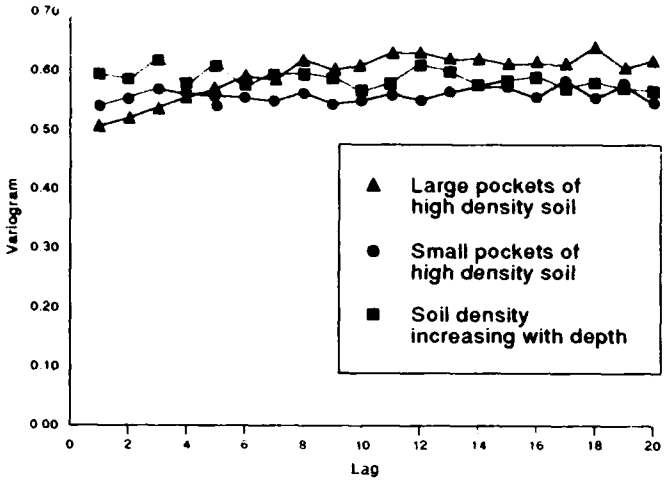


Figure 6. Variogram estimates of data simulated to have the characteristics described in Figures 1a, b and c.

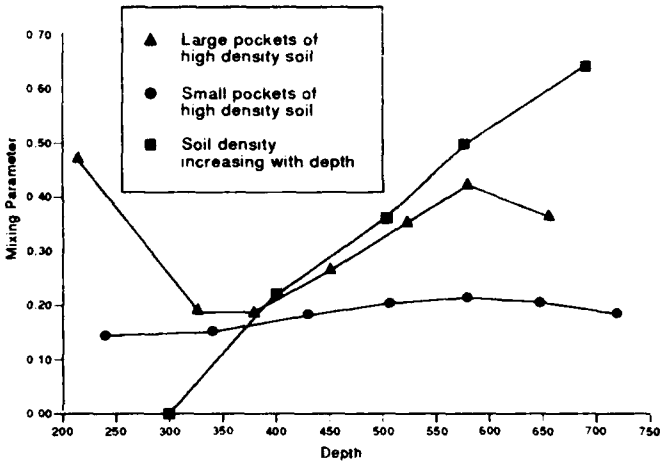


Figure 7. Regression of the mixing parameter versus depth for the three types of data described in Figures 1a, b and c.

variogram estimators.

An example of an actual mixing parameter regression curve is shown in Figure 8 as is the population curve which mixing parameter regression estimates. These curves correspond to the soil configuration depicted in Figure 1c. Although the shape (linear) and the slope can be estimated with great precision from a sample of n=1000 points Figure 8 also shows a bias which is characteristic of the computational and statistical approaches which are currently available. Bias in the

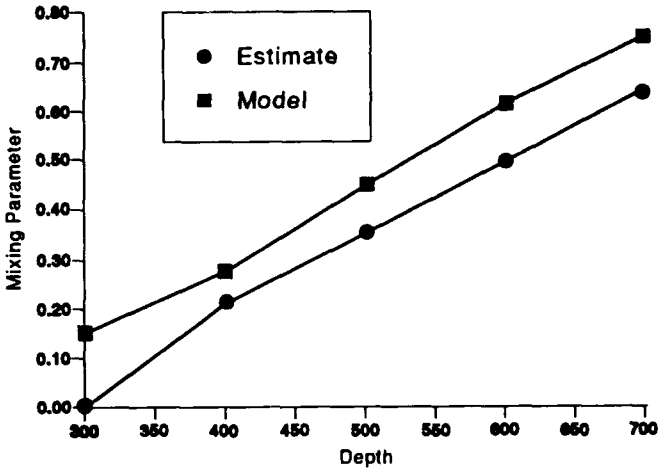


Figure 8. Comparison of estimated and true regression relationship between the mixing parameter and depth.

mixing parameter regression process tends to be the result of the following two closely related causes: (1) the process which first separates distributional components and then, in effect, snaps them back into their non-variance-reduced form does not eliminate all overlap effects. (2) The curve estimation methodology upon which the component reduction and isolation methodology is based is in no way tuned to perform well for mixture parameter regression applications. In particular, as discussed by Tarter and Lock (1991), with very few exceptions, there is a tendency for both kernel and series approaches to inflate the variance of estimated densities.

Use of an everywhere non-negative fixed bandwidth kernel must inflate this variance by a constant approximately equal to the variance of the kernel. (Expression 10 of Tarter and Raman (1971) indicates that such an estimate is the convolution of the kernel and a density whose variance is identical to n/(n-1) times the sample variance.) Hence, we are presently experimenting with the equivalent of variable (Brieman, L., Meisel, W. and Purcell, E., 1977) and somewhere-negative kernel methods which will yield mixing parameter regression estimators which have reduced bias properties. It is hoped that when appropriate methods are found it will be possible to obtain accurate mixing parameter regression curves with sample sizes considerably smaller than the n=1000 sized samples used in the above experiments.

6. References

Brieman, L., Meisel, W. and Purcell, E. (1977), Variable kernel estimates of multivariate densities, *Technometrics*, 19: 135-44.

Cressie, N. and Hawkins, D.M. (1980), Robust estimation of the variogram: I, *Mathematical Geology*, 12: 115-125.

Cressie, N. (1986), Kriging nonstationary data, *Journal of the American Statistical Association*, 81: 625-634.

Cressie, N. (1988), Spatial prediction and ordinary kriging, *Mathematical Geology*, 20: 405-421.

Fraser, D.A.S. (1958), *Statistics: An Introduction*, New York: Wiley.

Helling, C.S. and Gish, T.J. (1986), Soil characteristics affecting pesticide movement into ground water, in *Evaluation of Pesticides in Ground Wate*, (Garner, W.Y., Honeycutt, R.C. and Nigg, H.N., eds.), Washington DC: American Chemical Society, 14-38.

Kiefer, N. (1978), Discrete parameter regression: Efficient estimation of a switching regression model, *Econometrika*, 46: 427-434.

Kronmal, R.A. (1964), *The Estimation of Probability Densities*, Unpublished doctoral dissertation, Division of Biostatistics, University of California, Los Angeles.

Quandt, R.E. (1958), The estimation of the parameters of a linear regression system obeying two separate regimes, *Journal of the American Statistical Association*, 51: 873-880.

Quandt, R.E. (1972), New approach to estimating switching regressions, *Journal of the American Statistical Association*, 67: 306-310.

Quandt, R.E. and Ramsey, J.B. (1978), Estimating mixtures of normal distributions and switching regressions, *Journal of the American Statistical Association*, 73: 730-751.

Ray, R. and Turk, L. (1991), Uncertainty analysis for flow of water through unsaturated soil, *Amer. Soc. Mech. Eng. Winter Conference.*

Tarter, M.E., Lock, M.D. (1991), Model-free curve estimation: mutuality and disparity of approaches, *Journal of Official Statistics*, 17: 58-73.

Tarter, M.E. (1979a), Biocomputational methodology - an adjunct to theory and applications, *Biometrics*, 35: 9-24.

Tarter, M.E. (1979b), Trigonometric maximum likelihood estimation and application to the analysis of incomplete information, *Journal of the American Statistical Association*, 74: 132-139.

Tarter, M.E., and Raman, S. (1971), A systematic approach to graphical methods in Biometry, in *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability. Volume IV: Biology and Health*, (Neyman, J. ed.) Berkeley: University of California Press, 199-222.

Tarter, M.E., Silvers, A. (1975), Implementation of bivariate Gaussian mixture decomposition, *Journal of the American Statistical Association*, 70: 47-55.

Titterington, D. M., Smith, A.F.M., and Makov, U.E. (1985), *Statistical Analysis of Finite Mixture Distributions*, Chichester, England: Wiley.

Wagenet, R. (1986), Principles of modeling pesticide movement in the unsaturated zone in *Evaluation of Pesticides in Ground Water*, (Garner, W.Y., Honeycutt, R.C. and Nigg, H.N., eds.), Washington DC: American Chemical Society.

On Optimal Stopping Rules in Software Reliability

Mark C. K. Yang

Department of Statistics, University of Florida

Gainesville, Florida 32605

Anne Chao

Institute of Statistics, National Tsing Hua University

Hsin-Chu, Taiwan, R.O.C. 30043

1. INTRODUCTION

Software testing, or debugging, is one of the most important components in software development. It has been estimated that in many projects, the time accounted for debugging can be around 50% of the total development effort. There is an obvious question in the debugging process, that is when to stop. One naive answer is, of course, the process continues until there are no bugs (errors) in the program. However, this is a very difficult goal to achieve. For most commercial software, the release requirement is usually not 100% error free, but an acceptable error rate. Again, to determine when a debugging process has reached this stage is difficult. The best bet is often an *estimate* of the future error rate. However, the accuracy of the estimate may not be very high, depending on the estimation formula, and more seriously, on the assumptions that the formula is based upon. Let there be m faults in the software and their failure rates be

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m > 0. \qquad (1.1)$$

Then there are assumptions on equal failure rate for all the faults(eg. [2, 13, 14]), and unequal failure rates (eg. [3, 8]). Or a model based on failure time during testing instead failure rates of faults. Among them are the basic execution time and logarithmic Poisson execution time models(see Musa et al [10, 11]). However, these models are usually very difficult to verify.

In this paper, we try to find the optimal stopping rules based on testing cost and fault penalty after the software is release under very little on the model assumption. The debugging procedure is to test N cases during each testing stage, then to debug the software according to the testing result and make a decision whether to stop testing or not. Here we do not restrict to the case N = 1 as in the usual sequential analysis, because for large programs, the testing and debugging may not be done simultaneously. Debugging is performed only when a large number of programs has been tested. Optimal stopping rules in software testing has been notices in some recent literature. For example, Ross[14] considered a stopping rule based on an estimate of the future failure rate. It differs from ours in cost criterion. We feel that to know when to stop, one must know the relative costs between testing and penalty due to future failure. Dalal and Mallows ([6], [7]) considered loss function that can be equated to costs, but their rules depend on some prior assumption of the λ's. Rasmussen and Starr[13], Nayak[12], and Goudie[9] has also considered loss function, but their loss is basically a function of the remaining number of bugs instead of the future failure rate. It seems that the main concern of the software reliability is on the future failure rate rather than the number of bugs. Of course, the two are equivalent if equal failure rates are assumed for each bug. We feel that this assumption is probably not realistic.

2. Theory

In this section, only the most reasonable cases are presented. Possible generalizations to more complicated situations are given in §4. The assumption on λ is (1.1) with m and all the λ values unknown.

Let c_1 denote the cost of testing one case, c_2 be the cost (penalty) when an error is encountered in the released software, and M be the expected cases to be run by the consumers.
To develop theory, let I(x) be the indicator function

$$I(x) = \begin{cases} 1 & \text{if x is true} \\ 0 & \text{if x is false,} \end{cases}$$

$X_i(n)$ = number of times that the ith bug is encountered at the end of the nth test period,

and the remaining failure rate

$$U(n) = \sum_{i=1}^{m} \lambda_i I(X_i(n)=0).$$

Then if the program is released after the nth testing period, the cost is

$$C(n) = c_2 M \cdot U(n) + c_1 nN. \qquad (2.1)$$

Our purpose is to find the optimal stopping rule ψ such that

$$EC(\psi) \leq EC(\tau),$$

for any stopping rule τ. Here E denotes the expectation. Note that a stopping rule is a decision that depends only on the sampling information from the past, not the future. To put it in the usual notation, a stopping rule τ is a random variable such that the event $(\tau=n) \in \mathcal{F}_n$, where $\mathcal{F}_n$ is the sigma field generated by all the previous samples up to n. To find the optimal stopping rule, we first assume that all the λ's are known and use Theorem 3.3 in Chow, Robbins, and Siegmund[5]. In order to follow the theorem more easily, we let the payoff function $g(n)=-C(n)$ and the equivalent optimal stopping rule now is to find ψ such that

$$Eg(\psi) \geq Eg(\tau), \qquad (2.2)$$

for any stopping rule τ. Without loss of generality, we may let $c_1=1$ and $c_2M=c$. Hence,

$$g(n) = -cU(n) - nN.$$

Using our notation, we restate Theorem 3.3 of [5]. If the set

$$A_n = \{ E(g(n+1) \mid \mathcal{F}_n) \leq g(n) \}$$

is monotonically increasing with respect to n and

$$\liminf_n \int_{\psi>n} g^+(n)dP = 0,$$

then (2.2) is true for all τ satisfying

$$\liminf_n \int_{\tau>n} g^-(n)dP = 0,$$

where ψ is

the first $n \geq 1$ such that $g(n) \geq E(g(n+1) \mid \mathcal{F}_n)$.

It can be shown that the optimal stopping rule ψ is to stop at

$$\text{the first } n \geq 1 \text{ such that } \sum_{i=1}^{m} \lambda_i [1-(1-\lambda_i)^N] \cdot I(X_i(n)=0) \leq N/c. \qquad (2.3)$$

For small λ's, a good approximation for (2.3) becomes

the first $n \geq 1$ such that

$$\sum_{i=1}^{m} \lambda^2_i I(X_i(n)=0) \leq 1/c. \qquad (2.4)$$

Since the λ's are unknown, the ψ defined in (2.3) cannot be put into practice, but it tells us that if there is a good estimate of the left side of the inequality in (2.3) or (2.4), we may be close to the optimal stopping rule. Moreover, if any stopping rule that can almost reach the optimal value $EC(\psi)$ obtained from ψ when the λ's are known, it must be nearly optimal. From the simulation study to be presented in the next section, many nearly optimal situations are identified.

Let $\theta = E \sum_{i=1}^{m} \lambda^2_i I(X_i(n)=0)$. Then it can be shown that

$$\theta = \sum_{i=1}^{m} \lambda^2_i (1-\lambda_i)^{nN}$$

It is known in the literature (eg. [2], [4], [15]) that

$$E \sum_{i=1}^{m} I(X_i(n)=2) = \frac{nN(nN-1)}{2} \sum_{i=1}^{m} \lambda^2_i (1-\lambda_i)^{nN-2}.$$

By ignoring the small difference between $(1-\lambda_i)^{nN-2}$ and $(1-\lambda_i)^{nN}$, we can estimate θ by

$$\frac{2}{nN(nN-1)}\sum_{i=1}^{m} I(X_i(n)=2) \equiv \frac{2}{nN(nN-1)} B_n,$$

where B_n is the number of doubletons, i.e., the number of bugs that have been encountered exactly twice up to stage n. Thus, a reasonable adaptive rule $\hat{\psi}$ for the recapture debugging procedure is to stop at

$$\text{the first n such that } \frac{2}{nN(nN)} B_n \le 1/c. \tag{2.5}$$

The expected cost of this rule is denoted by $EC(\hat{\psi})$. When the standard debugging procedure is used, the number of doubletons up to stage n has to be estimated, because all the previous bugs before stage n have been removed. Let s_n and b_n denote the number of singletons and doubletons discovered at test period n. They are observable.

Let S_n denotes the number of singletons encountered up to stage n, S_n and B_n can be estimate recursively by

$$\hat{B}_n = \hat{B}_{n-1}\left(1-\frac{2}{N(n-1)}\right)^N + \hat{S}_{n-1}\left(1-\frac{1}{N(n-1)}\right)^{N-1}/(n-1) + b_n \tag{2.6a}$$

$$\hat{S}_n = \hat{S}_{n-1}\left(1-\frac{1}{N(n-1)}\right)^N + s_n. \tag{2.6b}$$

The two formulae are derived from the maximum likelihood principle. Thus, for the standard debugging procedure, the stopping rule $\tilde{\psi}$ is to stop at

$$\text{the first n such that } \frac{2}{nN(nN-1)} \hat{B}_n \le 1/c. \tag{2.7}$$

Again, we denote the cost under this rule by $EC(\tilde{\psi})$. Analytic study on the performance of (2.5) and (2.7) seems to be very difficult. Simulations are used to evaluate their performances.

3. SIMULATION STUDY

The stopping rules; optimal (2.3), approximation (2.4), adapted to recapture debugging procedure (2.5). and adapted to the standard debugging procedure (2.7) are compared. In standard software development, the failure rate should not be very high at the testing stage. Three values, 0.10, 0.05, and 0.01 are assigned for the total failure rate $T_\lambda \equiv \sum_{i=1}^{m}\lambda_i$, with m=100. Four configurations for λ are used;

a) rapidly decreasing λ (exponential rate): $\lambda_i=K/2^i$, i=1,2,...,m;
b) moderately decreasing λ (Zipf's Law): $\lambda_i=K/i$, i=1,2,...,m;
c) slowly decreasing λ (constant): $\lambda_i=K$, for all i=1, . . .,m;
d) random λ (following [14]): $\lambda_i=K\ U_i$, U_i is a random number, i=1,...,m,

where K is the normalization constant so that

$$\sum_{i=1}^{m}\lambda_i=T_\lambda.$$

Since T_λ is small, small N will make the test very ineffective. We choose N=100, 300, 500. Since N=100 is sometimes too small for the case T_λ=0.01, we hold our decision if there is no doubletons before the 5th period. Similarly, we hold our decision for N=300 and 500 if there no doubletons in the first period. The value $c=c_2M/c_1$ can vary considerably due to different real situation. We feel that c_1=1, c_2=100, and $M=10^4$ is a reasonable middle ground. Thus, three values, $c=10^5$, 10^6, and 10^7 are used. One hundred simulations were done for each combination of c, N, and T_λ.

The following results have been observed.

1) The first thing that surprises us is that there is little difference between (2.5) and (2.7). After detailed check into the stopping process, we found that this was due to large variation in B_n, the number of doubletons. Singletons are more stable in the sample. Thus, the estimated $\hat{B}_n$ from singletons can be as effective as the doubletons.

2) It is actually unfair to compare (2.5) and (2.7) with (2.3), because in (2.3) all the λ's have to be known. There is a tremendous prior information difference between (2.3) and the two adaptive stopping rules. However, the simulations show that in most situations, especially for cases b, c, and d, the adaptive methods perform extremely well. It is unlikely that in these situations any other stopping rule can beat them without any prior information on λ. At least we can say that they are nearly optimal.

3) From the expected cost point of view, the initial total failure rate T_λ has less influence than the sizes of λ.

4) Case a shows the biggest discrepancy in

costs between (2.5), (2.7), and (2.3). The reason is easy to see. If we have $\Sigma \lambda_i^2 < 1/c$ in the beginning, then no testing is the best strategy when the λ's are known. But under the real situation that the λ's are unknown, it will take a considerable number of testing samples to discover this fact. Thus, the adaptive methods cost considerably more than the optimal rule (2.3). Another situation, such as the rapidly decreasing λ case, is that although $\Sigma \lambda_i^2 < 1/c$ is not true for all the λ's in the beginning, the λ's are dominated by a few large ones and once they are removed, $\Sigma \lambda_i^2 < 1/c$ is satisfied by the rest of the λ's. Since the large λ's can be discovered pretty easily, they can be removed in the very beginning. The debugging process can then be stopped because the λ's are known. The adaptive methods again have to identify this fact at considerable cost.

5) The final costs vary little due to the test size N.

4. CONCLUDING REMARKS

1) When the λ's are equal, (2.3) is equivalent to (2.1) in Rasmussen and Starr[13]. We also repeated the simulation for the cases they considered. Our results confirm their results.

2) In the present study, the testing size N is assumed to be the same in all the testing stages. An interesting question would be what happens if we vary N. One thing we noticed is that the proof of the optimality of (2.3) is no longer valid. It seems to be a significant contribution if the tester can choose the optimal sample size at each stage.

3) From the derivation of (2.3), we can extend the result to a more general cost function i.e., let the cost for doing x tests be f(x). Then if $\Delta f(x) \equiv f(x+1) - f(x)$ is a nondecreasing function, then Theorem 3.3 of [5] holds and the optimal stopping rule becomes to stop at

$$\text{the first } n \geq 1 \text{ such that } \sum_{i=1}^{m} \lambda_i[1-(1-\lambda_i)^N]$$
$$I(X_i(n)=0) \leq [f((n+1)N) - f(nN)]/c.$$

The assumption of $\Delta f(x)$ being nondecreasing is reasonable when delay in releasing the software is considered as a cost.

REFERENCES

[1] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood, "Evaluation of competing software reliability prediction," *IEEE Trans. Software Eng.*, vol. SE-12, no. 9, pp. 950-966, Sept. 198

[2] P. K. Banerjee and B. K. Sinha, "Optimal and adaptive strategies in discovering new species," *Sequential Analysis*, Vol. 4, pp. 111-122, 1985.

[3] D. B. Brown, S. Maghsoodloo, and W. H. Deason, "A cost model for determining the optimal number of software testing cases," *IEEE Trans. Software Eng.*, Vol. Se-15, no. 2, pp. 218-221, Feb. 1989.

[4] A. Chao, "On estimating the discovering of a new species," *Annals of Statistics*, Vol. 9, pp. 1339-1342, 1981, Correction, 10, p. 1331, 1982.

[5] Y. S. Chow, H. Robbins, and D. Siegmund, *Great Expectation: The theory of optimal stopping*, Houghton Miffon Co., 1971.

[6] S. R. Dalal and C. L. Mallows, " When should one stop testing software, " *J. Am. Stat. Assoc.*, Vol. 83, 872-879,1988.

[7] S. R. Dalal and C. L. Mallows, "Some graphical aids for deciding when to stop testing software," *IEEE J. on Selected Area in Commun.*, 8, 169-175, 1990.

[8] E. H. Forman and N. D. Singpurwalla, "An empirical stopping rule for debugging and testing computer software," *J. Am. Stat. Assoc.*, Vol. 72, pp. 750-757, Dec. 1977.

[9] I. B. J. Goudie, "A likelihood-based stopping rule for recapture debugging," *Biometrika*, Vol. 77, 1, pp. 203-206, 1990.

[10] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability, Measurement, Prediction, Application*, McGraw-Hill, New York 1987.

[11] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability, Professional Edition*, McGraw-Hill, New York 1990.

[12] T. K. Nayak, "Estimating population size by recapture sampling," *Biometrika*, Vol. 75, no. 1, pp. 113-20, 1988.

[13] S. L. Rasmussen and N. Starr, "Optimal and adaptive stopping in the searching for new species," *J. Am. Stat. Assoc.*, Vol. 74, pp. 661-667, Sept. 1979.

[14] S. M. Ross, "Software reliability: The stopping rule problem," *IEEE Trans. Software Eng.*, Vol. SE-11, No. 12, pp. 1472-1476, Dec. 1985.

[15] N. Starr, "Linear estimation of the probability of discovering a new species," *Annals of Statistics*, Vol. 7, no. 3, pp. 644-652, 1979.

Statistical Models in Software Reliability

*Mark C. van Pul**

CWI**

P.O.Box 4079, 1009 AB Amsterdam
The Netherlands

Abstract: *In software reliability theory many different models have been proposed and investigated. Most of these models assume perfect repair and constant software size. Both restrictions oversimplify reality in a huge way. In the model we will discuss in this paper, we have tried to overcome both simplifications in such a way that statistical inference is still possible.*

1. Introduction

The reliability of hard- and software is sometimes of vital importance to their users. During the recent Gulf war a patriot missile, which was stationed in Turkey, was fired by accident, because of a bug in the software. Obviously also in case of less delicate computer applications customers want a high degree of reliability to be guaranteed. The modelling of the evolution of the reliability of a piece of software undergoing debugging will be the subject of this paper.

In the next section we will give some backgrounds and classical assumptions of software reliability theory. In the third section we describe the PGIR model, a new model with interesting features, and we will suggest how to estimate the model parameters. In section 4 we will shed some light upon the huge amount of extensions, that are possible, starting from this model; we will define a class of regression models. Finally, in the fifth and last section we give some concluding remarks. This short paper just tries to give some ideas and results. More details, derivations and proofs can be found in Van Pul (1991).

2. Backgrounds and classical assumptions

Let us consider the following test experiment. A very large computer program is executed during a fixed exposure period, say $[0,\tau]$. Inputs are selected "at random" from the input space, that is, they are generated in such a way that they are representative for the operational profile. For each input the program either produces the correct output or a software failure is detected; the software produces a wrong answer or no answer at all. After the detection of a failure the CPU-clock is stopped and the software is sent to a team of debuggers. The failure time and possibly other failure data are observed. After the bug is found and fixed, the CPU-clock is restarted again and testing continues with a new input until time τ is reached.

Efforts in describing the evolution of the reliability of computer software during test and development resulted in the proposal of dozens of new models over the past twenty years. An important class of such models is the so-called class of Error-Counting and Debugging (EC&D) models. This class consists of models that are based on the test experiment described above (with only the failure times as test data) and some strong assumptions:

(A1) *Perfect repair:* no new faults are introduced during a repair with probability 1.

(A2) *Fixed software size:* there is no addition of new software during testing.

(A3) *Independence of faults:* faults (and hence their failure times) are independent.

Although all three assumptions seem to be rather unrealistic, they form a framework on which many models are built. The most elementary and oldest software reliability model is the model of Jelinski-Moranda (1972), introduced almost twenty years ago. In this model the failure rate of the program is assumed to be at any time proportional to the number of remaining faults and the repair of each fault does make the same contribution to the decrease in failure rate. Denoting $n(t)$ for the observed counting process, we find for the failure intensity function $\lambda(t)$ the following

* This research was carried out under a grant of the Netherlands Technology Foundation (STW).

** CWI is the nationally funded institute for research on mathematics and computer science, formerly called Centre for Mathematics and Computer Science.

expression:

$$\lambda(t) = \phi\left[N - n(t-)\right], \quad t \in [0,\tau], \tag{1}$$

with model parameters N, the number of faults initially present in the software, and ϕ, the occurrence rate per fault, which can also be interpreted as the test efficiency. Musa (1975), Littlewood (1980) and many others have built more sophisticated models, for technical reasons, however, generally restricted by assumptions (A1)-(A3).

As there exist no perfect testers and programmers, there will always be a positive chance of introducing new faults, while repairing an old one. Secondly, development and testing of software usually takes place simultaneously in practice. Because the addition of software, that has never been tested before, certainly will have an effect on the reliability, it seems reasonable to take also software growth during testing into account. Furthermore certain bugs will prevent parts of the software to be inspected and therefore will hide other bugs, thus violating the assumption of independence of faults. Dropping (A3), however, would cause the mathematical problem to become highly complicated and almost untractable.

In the next section we introduce a new model, the Poisson Growth and Imperfect Repair (PGIR) model. We combined the modelling of imperfect repair and software growth in a natural way. Furthermore to a certain extent the model will account for dependencies between faults. The model has attractive statistical properties, besides.

3. The PGIR model.

Let $\tau > 0$. We consider a test experiment as described in the introduction. Let $T_0 := 0$ and $T_i, i = 1,2,...$ the failure times of the occurring failures. Repair takes place immediately after a failure is detected. For reasons of convenience the addition of new software takes only place at the failure times T_i. Due to the correction of a fault and eventually due to the addition of new software at time T_i, there is a change in the software of size $K_i, i = 0,1,....$ The K_i are hence the known outcomes of some deterministic software measure, e.g. lines of code, complexity, number of loops or subroutine-calls . At time T_i apart from deleting one fault, N_i new faults are introduced, partly due to bad repair and partly due to the addition of new software. It seems reasonable that N_i is in some sense proportional to the "size" of the total change in the software at time T_i. We therefore assume that $\mathbf{N}_i$ is a stochastic variable, Poisson distributed with mean $\mu K_i, i = 0,1,...,$ where μ a parameter. We consider the testing process during $[0,\tau]$, observing say $n(\tau)$ faults. Let

$$n(t) := \sum_{i=1}^{n(\tau)} (T_i < t), \qquad t \in [0,\tau], \tag{2}$$

the number of failures detected (faults deleted) during $[0,t]$ and let

$$N(t) := \sum_{i=0}^{n(\tau)} \mathbf{N}_i (T_i < t), \qquad t \in [0,\tau], \tag{3}$$

the number of faults introduced during $[0,t]$, where

$$\mathbf{N}_i =_d POI(\mu K_i). \tag{4}$$

We assume that the failure intensity λ, like in the Jelinski-Moranda model, at any time is proportional to the remaining number of faults, that is:

$$\lambda(t) := \phi\left[N(t-) - n(t-)\right], \qquad t \in [0,\tau], \tag{5}$$

where ϕ denotes the constant occurrence rate per fault. With use of the data (T_i, K_i), $i = 0,1,...,n(\tau)$, obtained from the experiment as described above, one can estimate the parameters (μ,ϕ) of the underlying PGIR model. We will use the maximum likelihood estimation (MLE) procedure for this purpose. The following lemma will be very useful:

Lemma 1:
For all $m \in \mathbb{N}$ and all $(a_0, a_1, \cdots a_m) \in \mathbf{R}_+^{m+1}$, we have:

$$\sum_{N_0=0}^{\infty} N_0 \frac{{a_0}^{N_0}}{N_0!} \sum_{N_1=0}^{\infty} (N_0 + N_1 - 1) \frac{{a_1}^{N_1}}{N_1!} \cdots$$

$$\cdots \sum_{N_m=0}^{\infty} (N_0 + N_1 + \cdots + N_m - m) \frac{{a_m}^{N_m}}{N_m!}$$

$$= a_0 (a_0 + a_1) \cdots$$

$$\cdots (a_0 + a_1 + \cdots + a_m) e^{a_0 + a_1 + \cdots + a_m}. \tag{6}$$

Proof:
The result follows immediately with natural induction.

We now return to the derivation of the likelihood function for the PGIR model, as described by (2)-(5). Aalen (1978) showed, that

the likelihood function for estimating the parameters of the intensity function of a counting process, observed on a fixed time interval $[0,\tau]$ is given by:

$$l(\vec{N}) := L_\tau(\mu,\phi;T_0,T_1,\cdots,T_{n(\tau)})$$

$$= \prod_{i=1}^{n(\tau)} \lambda(T_i-)\exp(-\int_0^\tau \lambda(s)ds). \quad (7)$$

As the N_i are independent Poisson distributed stochastic variables with mean μK_i we have

$$p(\vec{N}) := \mathbb{P}\left[\mathbf{N}_i = N_i, i=1...n(\tau)\right]$$

$$= \prod_{i=0}^{n(\tau)} \left[\frac{(\mu K_i)^{N_i}}{(N_i)!}e^{-\mu K_i}\right] \quad (8)$$

and defining:

$$a_i := \mu K_i e^{-\phi(\tau-T_i)}, \quad (9)$$

$$b_i := (N_0+\cdots+N_{i-1}=i), \quad (10)$$

for $i=1,...,n(\tau)$, we obtain the likelihood function under the finer filtration (observing also the sizes of the software changes) by summing Aalen's expression (7) over all possible realisations of the N_i multiplied by their joint probabilities (8):

$$L_\tau(\mu,\phi;(T_i,K_i),i=0,1,...,n(\tau)) =$$

$$\sum_{N_0=1}^{\infty}\sum_{N_1=b_1}^{\infty}\cdots\sum_{N_{n(\tau)-1}=b_{n(\tau)-1}}^{\infty}\sum_{N_{n(\tau)}=0}^{\infty} p(\vec{N})\, l(\vec{N})$$

$$= \phi^{n(\tau)} \exp\left[\phi\sum_{i=1}^{n(\tau)}(\tau-T_i) - \mu\sum_{i=0}^{n(\tau)}K_i\right] \times$$

$$\times \sum_{N_0=1}^{\infty} N_0 \frac{a_0^{N_0}}{N_0!}\sum_{N_1=b_1}^{\infty}(N_0+N_1-1)\frac{a_1^{N_1}}{N_1!}\cdots$$

$$\cdots\sum_{N_{n(\tau)}=0}^{\infty}(N_0+...+N_{n(\tau)}-n(\tau))\frac{a_{n(\tau)}^{N_{n(\tau)}}}{N_{n(\tau)}!} \quad (11)$$

We note that if $N_0+\cdots+N_{i-1}=1$, that is, if $b_i=1$ (so we have to sum N_i from 1 to ∞), then the coefficient $(N_0+...+N_i-i)$ in the i-th sum equals zero for $N_i=0$. So we can take all lower bounds equal to zero and use lemma 1 to get:

$$L_\tau(\mu,\phi;(T_i,K_i),i=0,1,...,n(\tau)) =$$

$$= \phi^{n(\tau)} \exp\left[\phi\sum_{i=1}^{n(\tau)}(\tau-T_i) - \mu\sum_{i=0}^{n(\tau)}K_i\right] \times$$

$$\times \sum_{i=0}^{n(\tau)} a_i \prod_{i=0}^{n(\tau)-1}\left[\sum_{j=0}^{i} a_j\right]$$

$$= \prod_{i=0}^{n(\tau)-1}\left[\mu\phi\sum_{j=0}^{i}K_j e^{-\phi(\tau-T_j)}\right] \times$$

$$\times \exp\left[\phi\sum_{i=1}^{n(\tau)}(\tau-T_i) - \mu\sum_{i=0}^{n(\tau)}K_i(1-e^{-\phi(\tau-T_i)})\right] \quad (12)$$

We now take the logarithm of the likelihood function (12), set the partial derivatives equal to zero and solve the system of two ML-equations, finding expressions for the ML estimators $\hat{\mu}$:

$$\hat{\mu} := \frac{n(\tau)}{\sum_{i=0}^{n(\tau)}K_i\left[1-e^{-\hat{\phi}(\tau-T_i)}\right]} \quad (13)$$

and $\hat{\phi}$ is the solution of $g(\hat{\phi})=0$ with

$$g(\phi) := \frac{1}{n(\tau)}\sum_{i=1}^{n(\tau)}(\tau-T_i) + \frac{1}{\phi}$$

$$- \frac{\sum_{i=0}^{n(\tau)}K_i(\tau-T_i)e^{-\phi(\tau-T_i)}}{\sum_{i=0}^{n(\tau)}K_i\left[1-e^{-\phi(\tau-T_i)}\right]}$$

$$- \frac{1}{n(\tau)}\sum_{i=0}^{n(\tau)-1}\left[\frac{\sum_{j=0}^{i}K_j(\tau-T_j)e^{-\phi(\tau-T_j)}}{\sum_{j=0}^{i}K_j e^{-\phi(\tau-T_j)}}\right]. \quad (14)$$

It can be shown (see Van Pul (1990)) that the ML-estimators are consistent, asymptotically normal distributed and efficient.

Let us consider the PGIR model again as given by (2)-(5). Note that the process $N(t)$ is unobservable. Thus defining the filtrations

$$\mathcal{F}_{t-} := \{\, n(s) : 0\leq s<t \,\}, \quad (15)$$

$$\mathcal{G}_{t-} := \{\, n(s),N(s) : 0\leq s<t \,\}, \quad (16)$$

we notice that the intensity λ given in (5) is actually $\lambda^{\mathcal{F}}$, the intensity function of the counting process with respect to the filtration $\mathcal{F}_t$. With use of the Innovation Theorem (see e.g. Bremaud (1977)), and another application of lemma 1 we can show that the intensity function under the filtration $\mathcal{F}_{t-}$ (only observing the counting process $n(s)$, $0\leq s<t$ and the software changes K_i, $i=0..n(t-)$) is given by

$$\lambda^{\mathcal{F}}(t) := \mu\phi\sum_{i=0}^{n(t-)}K_i e^{-\phi(t-T_i)}. \quad (17)$$

An interesting idea seems to set all the K_i equal

to some $\bar{K}$ except for $K_0 >> \bar{K}$. With parameters $N_0 := \mu K_0$ and $\bar{N} := \mu \bar{K}$ the failure intensity becomes

$$\lambda(t;\phi,N_0,\bar{N}) := N_0\phi e^{-\phi t} + \bar{N}\phi \sum_{i=1}^{n(t)} K_i e^{-\phi(t-T_i)}. \quad (18)$$

In this three parameter model, $\bar{N}$, the average number of faults introduced per repair action, can be interpreted to account for dependencies between faults. Whenever hidden faults become observable because of a fault repair, this can be considered as the introduction of new faults. Finally note that for $\bar{N} = 0$ the above model reduces to the well-known model of Goel-Okumoto (1979).

4. Regression models

The PGIR model can be seen as a a special case within a general class of regression models. In the previous section I assumed that the N_i were Poisson distributed with a parameter depending on a single software measure. Because the process of introducing new faults is so difficult to understand, it seems appropriate to use explanatory variables and apply regression analysis. We therefore suggest the following class of models given by (2),(3),(5) and

$$\mathbf{N_i} =_d POI(X_i) \quad (19)$$

$$X_i := \exp\left[\beta_1 z_i^1 + ... + \beta_m z_i^m\right], \quad (20)$$

where the $z_i^j, j = 1...m$, are the known realisations of m software measures Z^j (like e.g. size, complexity, number-of-loops) at time T_i and where the $\beta_j, j = 1...m$, denote the corresponding regression coefficients we have to estimate. Statistical methods are available to investigate whether certain explanatory variables are redundant (or not) and whether their influence is linear, via another power, or say logarithmic.

5. Concluding remarks

We have constructed a model, which is able do deal with imperfect repair and software growth. Moreover, the ML-estimators for the model parameters have desirable asymptotic properties.

In the field of regression models for software reliability, there is in my opinion a lot of interesting research still to be done. Essential will be, however, the collection of real data (computation of various software measures) by software developers. So far, we did not get much response from them. Perhaps they should read Rook's (1990) Handbook on Software Reliability. In its preface Boehm resignedly states: *"Sometime soon, software reliability is going to become a highly visible and important field. Unfortunately, given human nature, its thrust into prominence will only happen once we experience the software equivalent of the Chernobyl, Bhopal, or space shuttle Challenger disasters. Such a disaster is likely to happen in the next few years..."*.

References

Aalen, O.O. (1978), Non-parametric inference for a family of counting processes. *Annals of Statistics* **6,** 701-726.

Bremaud, P. (1977), Processes ponctuals et martingales: Résultats récents sur la modélisation et filtrage. *Advanced Applied Probability* **9,** 362-416.

Goel, A.L. and Okumoto, K. (1979), Time Dependent Error-Detection Rate Model for Software Reliability and other Performance Measures. *IEEE Transactions on Reliability* **28,** 206-211.

Jelinski, Z. and Moranda, P. (1972), Software Reliability Research. *Statistical Computer Performance Evaluation* 465-484.

Littlewood, B. (1980), Theories of software reliability: How good are they and how can they be improved? *IEEE Transactions on Software Engineering* **6,** 489-500.

Musa, J.D. (1975), A theory of Software Reliability and Its Application. *IEEE Transactions on Software Engineering* **3,** 312-327.

Rook, P. (1990), *Software Reliability Handbook* Elsevier applied Science, London.

VanPul, M.C. (1990), *Asymptotic Properties of Statistical Models in Software Reliability* Report BS-R9011, Centre for Mathematics and Computer Science, Amsterdam.

VanPul, M.C. (1991), *Modelling Imperfect Repair and Software Growth* To appear.

Statistical Methodology for Software Systems Testing

David Zeitler
Smith's Industries Aerospace & Defense Systems, Inc.
4141 Eastern Avenue, S.E. Grand Rapids, MI 49518-8727
PHONE: (616)241-8168 FAX: (616)241-7533 EMAIL: zeitler@si.com

ABSTRACT

There is much from statistical methodology that can be brought to the testing of software systems. Both a general paradigm for testing and an approach to assurance of reliability will be derived from statistical methods. The testing paradigm provides both a general approach to the specification, design and analysis of tests and potential for reduced complexity of test equipment (thus reduced cost of testing). The testing for reliable software approach gives us a model upon which to start building theory for the automated generation of tests which go beyond requirements, capturing the intelligent behavior of the experienced test engineer. Relationships between this framework for testing and work on statistical advisory systems for the design of experiments and semantic understanding of text will be identified.

Introduction

The goal of this paper is to provide connections between statistics and the testing of software based systems which are not as obvious as software reliability growth modeling. These connections have been derived from the past ten years of testing, examining the process of testing and managing the testing of embedded software in the avionics industry. We will first examine an operating paradigm for the development of effective tests and then look at how this paradigm may lead toward automation of the more creative aspects of the test development process.

This work has been evolving in the context of the development and test of systems which have significant user interfaces. A major characteristic of these systems is that the user cannot be extricated from the system itself. Proper operation depends on the appropriate interaction between the system and the user. Any breakdown of this interaction can be the trigger of a failure. The resulting system is necessarily stochastic.

In systems without significant user interface, there is at least the potential to produce a sufficiently rigorous specification and implementation to remove the stochastic problems created by the user. Such systems have the potential for formal proof of correctness which eliminates the need for an experimental approach to verification and validation.

Tests are Experiments

Testing software based systems (or any testing) is an experimental process aimed at determining the state of the system under test with respect to some standard (possibly not deterministic). As experiments, paradigms for statistical design of experimentation (DOE) can be applied to improve our understanding and further advance the state of the art of software testing.

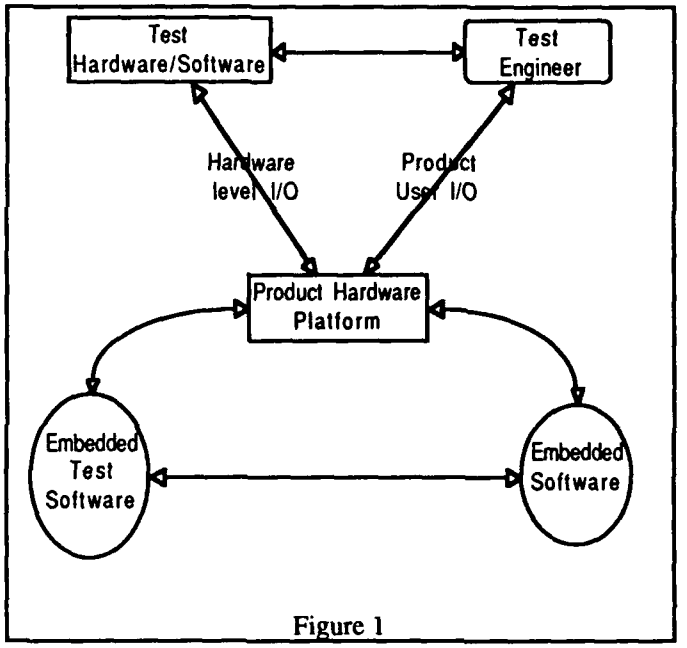


Figure 1

A Stochastic Environment

Generally software tests are considered to be deterministic with a clear pass/fail criteria. This is often not the case in embedded systems and/or systems with significant user interfaces. A normal embedded software test environment looks something like figure 1. Sources of random variation are the user (or test engineer) and often the test equipment which is attempting to simulate the operational environment of the system.

As mentioned in the introduction, the user will introduce a significant stochastic element into the test environment. It is not desirable to remove this stochastic element since

doing so makes our test input distribution even less like the operational distribution the system must operate under. It is then less likely to catch problems which are important to the user.

If we are dealing with a system which is using hardware at or near the limits of current technology, the test equipment may not be capable of providing adequate control to assure deterministic operation. The result is again random variation. Even if we are not operating at the leading edge of technology, test equipment which does not utilize expensive state of the art components to provide a fully deterministic environment is sometimes preferable.

Relation to Statistical Methods

Test objectives are equivalent to models and hypotheses in experimentation. The model defines the portion of the system of interest in the objective. Hypotheses then focus the test upon particular components of the model. Experimental design principles can then be applied to the design of the test which provides the requirements for the test environment. These requirements will include the degree of accuracy necessary to provide sufficient power to the hypothesis tests which make up the pass/fail criteria for the testing.

Requirements Based Testing

Requirements testing is based upon hypotheses which are selected from the system requirements. Models selected are subsystems 'carved' from the system architecture and chosen for compactness (minimal external connections) to reduce test environment requirements while encompassing a minimum of extraneous system components not directly related to the hypothesis of interest. From this point, the test task is just the design of the necessary environment to provide the control and data collection necessary to carry through the experiment.

This whole process is almost identical to the process of selection of appropriate experimental factors which provide a sufficiently powerful test of hypothesis about the safety or efficacy of a drug in pharmaceutical testing. The wealth of statistical methods applied in the certification of drugs in the pharmaceutical industry are then clearly applicable to the qualification (or certification) testing portion of software systems (or any other system for that matter).

In figure 2 we see a flow chart for the basic process undergone by a test engineer in developing qualification tests. Current versions of these systems are either human, or contain trivial 'engines' that just regurgitate requirements typed in by the test engineer. Next generation versions of these systems are expected to operate as an advisor, incorporating the expertise of the test engineer (user) into the process.

In better understanding this process and/or automating it, we will need to incorporate the knowledge from expert statistical advisory systems into the experiment design engine. In addition, expert knowledge about what makes 'good' hypothesis for qualification testing must be incorporated into the hypothesis engine.

Not shown in the diagram is the analysis portion of the process which would take test data and help to generate the report. Each of these components could likely be derived from existing work in various forms of statistical expert systems for design, analysis and inference.

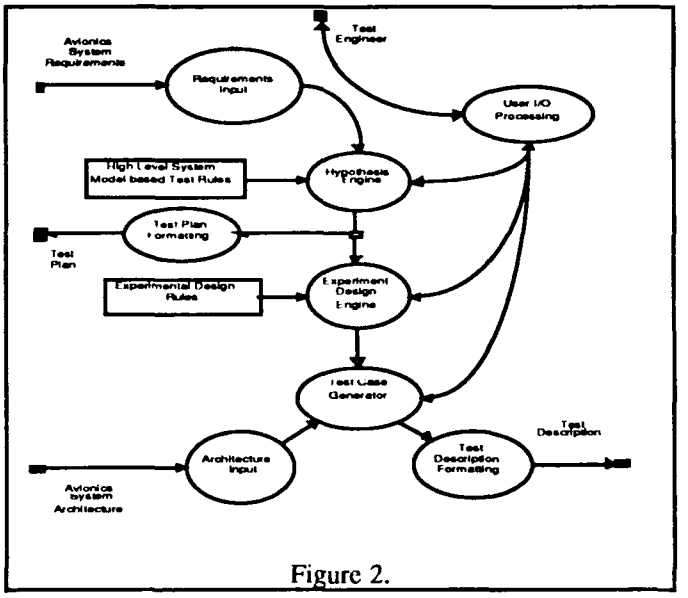


Figure 2.

DOE in a digital environment

The application of experimental design is not a straight forward exercise in the digital non-linear world of software systems. On the surface, it would appear that we will run in to a combinatorial explosion of necessary test conditions based on the discrete nature of the systems inputs and operation. This is not necessarily the case however. Fractional factorial designs can be used to reduce the combinatorial explosion, and aggregation (high level views) can be used to treat the system as essentially analog and linearizable.

Consider a software module which has as its primary input a 7 bit integer. If we view the 7 bits as independent two valued inputs, we can lay out an orthogonal design (2^{7-4} or a Taguchi L8) giving us an orthogonal cross section of the input space of the module with respect to the primary input. Assuming a pareto effect in faults and no high level interactions (singularities), we have an efficient set of test cases for the module. Certainly more efficient than selecting a couple of integer values at extrema or randomly from the

input range and not requiring more information about the module than its interfaces.

Pareto assumptions are reasonable in most software development environments today. All we're really assuming here is that the software faults are not 'dense' in the code. The absence of singularities is a bit harder to deal with. Techniques such as data dithering and data diversity are available to reduce the granularity of singularities. The problem cannot be entirely eliminated though.

Application of experimental design can be made reasonably obvious for some situations by taking a high level view of the system. Many functions which are implemented in an embedded digital system (such as navigation) are inherently analog in nature with only an overlying nonlinear mode structure. Within any particular node, the operation of the system is entirely analog at these high levels and experimental design or response surface methods are directly applicable.

Ad Hoc Testing

Testing to assure reliability of software systems (failure free operation) must go beyond adherence to requirements to address the validation of the system in an indeterminate environment. This immediately implies the application of statistics to the problem. But more than just statistical procedures are applicable. The entire paradigm of statistical modeling and hypothesis testing comes into play in this environment. This type of testing is what is called ad hoc.

Requirements based testing can only get at a portion of the aspects of a real system. This is because, as depicted in figure 3, in the real world, expectations, specifications and the reality of implementation seldom coincide. Initially, we're lucky if they are not disjoint.

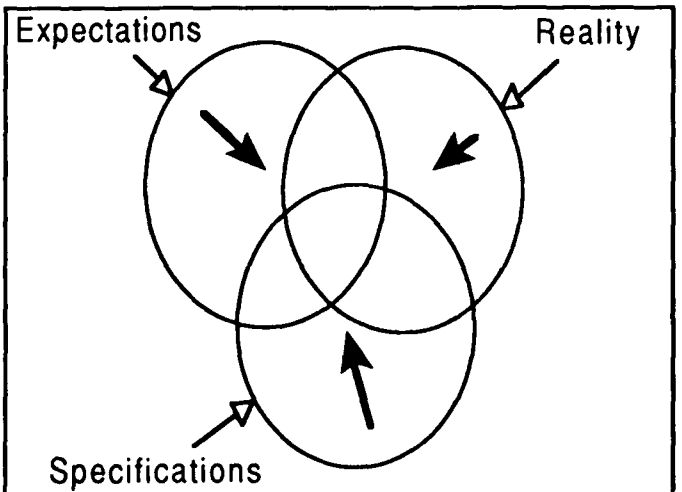


Figure 3.

In addition to the basic problem of bringing user expectations and specifications in line with what is physically realizable, we have additional difficulties which make real world systems more 'interesting' than ideal. Usually, a user's requirements are handed to the developer in the form of descriptions of responses to a limited region of the input domain represented by the lines, disk and single point in figure 4. Analysis attempts to produce a 'convex' region which encompasses these initial requirements. Design then proceeds to move this convex region toward a real implementation. In the process the highly non-convex region depicted results from errors and physical constraints.

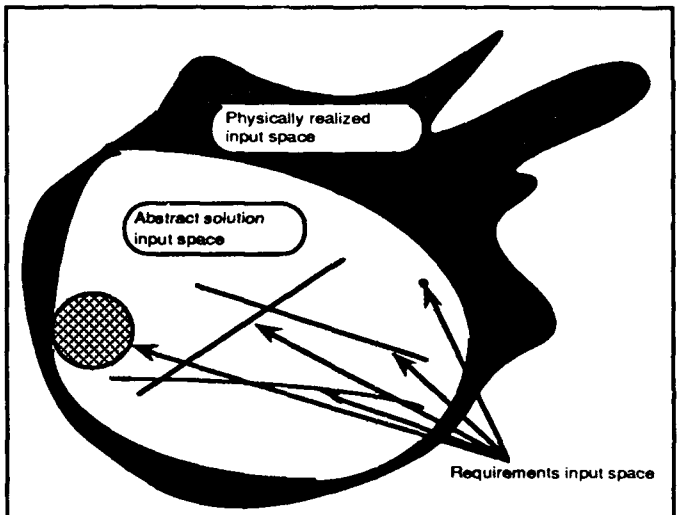


Figure 4.

Ad hoc testing then is aimed at those regions of the system shown outside the abstract solution space in figure 4. In this region lies problems which are due to implementation which goes beyond requirements and user expectations which don't show up in the requirements. For example, the well known ability of an early version of F-16 software that allowed the test pilot to raise the landing gear on the ground, probably lies near the tip of one of the lobes! Idealistically, no such aspects of the system exist, but as can be seen, this is effectively impossible if for no other reason than that the user's expectations are never constant or clear (since they are necessarily developed and interpreted by humans).

Iterative Learning

The interactive ad hoc test process is an iterative learning process much like that expressed in Box, Hunter and Hunter. A test engineer begins with an initial hypothesis and associated model of the system and iteratively hones each into a clearer understanding of the system. This iterative process is depicted in figure 5 as a tree structure. Each level

of the process takes the current hypothesis and from it specifies an appropriate model, designs and executes the associated experiment and uses the conclusions to refine and select the hypotheses for the next level. Note that the outcome of this refinement and selection may be a posterior distribution on the possible hypotheses which leads to multiple paths through the tree.

In an expert system, the inference engine takes an initial set of facts and proceeds to 'fire' rules from these facts to deduce further facts. The primary difference in the ad hoc testing process is that the rule 'firing' is actually an experimental process used to derive the rules from the real world rather than a data base.

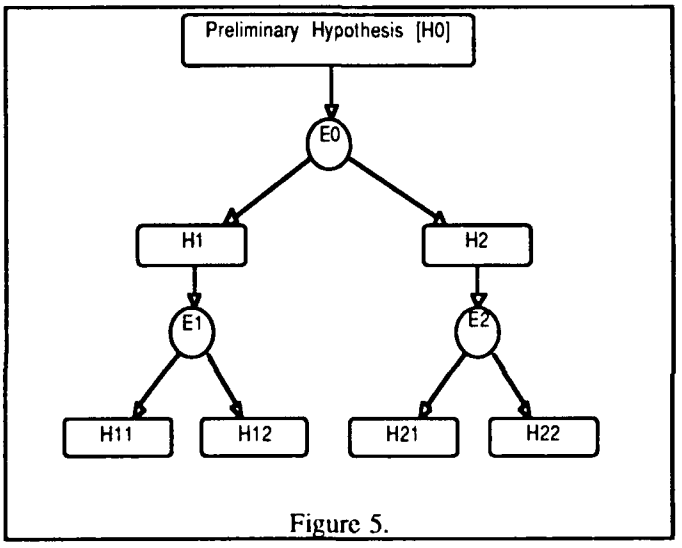


Figure 5.

Abstract system models

Many experienced test engineers can find problems in a system without knowing more than a rudimentary structure for the system. This suggests that the source of the initial hypothesis from which general hypotheses and working models can be derived is a high level abstract model of the type of system being tested. These experienced test engineers use (even if they're not aware of it) a high level abstract model of the system to guide the selection of hypothesis without falling back upon requirements. These models are a combination of operational experience and experience with the kinds of things which can and do go wrong in the development process and in real systems. Development of these abstract models can benefit from work in extracting semantic information from text.

In order to automate this process or improve our own capability of developing systems we need to understand and be able to produce this abstract model. Most test engineers don't even realize they are working within this paradigm, much less be able to transfer the knowledge of the model to an expert system. Alternatively, we can derive the abstract model from existing software systems. This is where current work in semantic recognition comes in. We can think of a program as a living book. The basic story or class of stories is fixed, but the details of the current instantiation of the story depend upon the data fed to the program.

Statistical analysis of the digraphs which represent the programs along with the variable and procedure names (assuming they're done with reasonable mnemonics) can help us to develop the kind of basic 'story' lines that are most often used in various classes of software. In these story lines we have an abstract view of the underlying abstraction which provides the basis for the software.

References

Ammann, P.E., Data Redundancy for the Detection and Tolerance of Software Faults, *Proceedings of the 22cd Symposium on the Interface*, East Lansing, Michigan - May 1990.

Box, Hunter & Hunter, Statistics for Experimenters, Wiley 1978.

Gale, William A. & Church, Kenneth W., A Statistical Approach to Aligning Sentences in Bilingual Corpora, *Preliminary Papers of the Third International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, Florida - January 1991.

Goldman, Robert & Charniak, Eugene, Probabilistic Text Understanding, *Preliminary Papers of the Third International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, Florida - January 1991.

Zeitler, David, Realistic Assumptions for Software Reliability Modeling, *Proceedings of the International Symposium on Software Reliability Engineering*, Austin, Texas - May 1991.

Interfacing Physiologically-based Pharmacokinetic Modeling and Simulation Systems

Derek B. Janszen and M.C. Miller, III Biostatistics, Epidemiology & Systems Science
Medical University of South Carolina, Charleston, SC 29425

ABSTRACT

The graphical user interface of a physiologically-based pharmacokinetic (PB PK) modeling and simulation system developed for the Macintosh™ computer is described. The user interactively specifies: 1) the anatomical structure of the model (tissues) and the anatomical structure of each tissue; 2) physiological relationships; 3) transport characteristics; 4) thermodynamic properties of the substance.

The interface utilizes four independent interactive windows: Model, Parameter, Kinetics, and Solution. The user selects tissues for the model and an exposure route from a flow diagram consisting of nine different tissues and four possible routes of exposure, or from a menu. Assumptions limiting the rate of mass transfer can be specified for each tissue. Parameters for each tissue, as well as dosage parameters, are entered *via* dialog boxes. This method of specifying the model parameters encourages "What if...?" scenarios. The model is cast in an S-system format for ease of solution and for added flexibility in simulating inherently nonlinear biological systems. The system generates a steady-state solution, which can be plotted as multiple tissue concentration–time curves on a configurable graph. The data files can be exported to other graphics and statistics packages. The pictorial flow diagram, a table of all tissue parameter values, the steady-state solution set, and the graphic plots can be printed.

INTRODUCTION

Physiologically-based pharmacokinetic (PB PK) models utilize a system of lumped compartments which are designed on the basis of the actual anatomy and physiology of the species. Model parameters fall into four broad categories: 1) **anatomical**, *e.g.*, organ volumes and tissue sizes; 2) **physiological**, *e.g.*, blood flow rates and enzyme reaction rates; 3) **thermodynamic**, *e.g.*, drug-protein binding isotherms; and 4) **transport**, *e.g.*, membrane permeabilities (Himmelstein and Lutz, 1979).

The first step in the development of a PB model is to select the number and type of tissues. Once the tissues are selected, a flow scheme is drawn with the desired regions describing the species anatomically (Figure 1). The liver, gut, spleen, and pancreas (enterohepatic system) are interconnected anatomically, maintaining the physiological basis.

Each tissue is initially considered to consist of three homogeneous subspaces: (a) a **vascular** space through which the tissue is perfused with blood; (b) an **interstitial** space, which forms a matrix for the tissue cells; and (c) an **intracellular** space consisting of the tissue cells that comprise the organ (Figure 2).

Rate-limiting assumptions may simplify the 3-subspace model to one or two subspaces. The **flow-limited model** has a single space and is used to model tissues that are not well perfused by the circulatory system. The **membrane-limited model** assumes that the transfer across the cell membrane is rate limiting, and thus reduces the tissue model to 2 subspaces. Our system admits any of the above three configurations for a tissue. The system also permits the specification of four physiological processes which can affect the distribution and flux of a substance: **transport** across a membrane, **binding**, **excretion**, and **metabolism**. A linear or non-linear formulation can be used for modeling these processes, depending on the available information for a given process. Generally, a PB model may include any or all of these processes.

IMPLEMENTATION OF PK MODELS

Although the PB approach to PK modeling is the method of choice, there is, among some, reluctance to use this approach because of the mathematics associated with the method (D'Souza and Boxenbaum, 1988). Nevertheless, progress in the development of computer software for solving the system of differential equations generated by these models is being reported.

The literature describes three modeling methods. They include utilization: 1) of a fixed model simulator where the number and/or types of tissues are fixed (Bloch *et al.*, 1980; Gabrielsson and Hakman, 1986; Menzel *et al.*, 1987); 2) of a general simulation system (Blau and Neely, 1987); and 3) of spreadsheet-based simulators (Ball *et al.* , 1985; Johanson and Näslund, 1988).

At the heart of these modeling methods are the algorithms used to solve the system of differential equations. Since algebraic solutions are not available for these complex models, they must be approximated by numerical methods. Two terms used to characterize these numerical methods are **accuracy** and **efficiency**: by **accuracy** is meant the error (difference) between the numerical solution and the true solution; by **efficiency** is meant the "cost" of the solution in terms of convergence of the estimation procedure, which is generally equated to computer time. Some of these methods are very simple, easy to program, and are efficient; their disadvantage is that they do not give very accurate results. Other methods, while achieving better accuracy, are more difficult to program and are less efficient.

The modeling system described in this paper utilizes the Power-Law Formalism (Savageau, 1969; Voit, 1991). It admits several system-modeling strategies. Table A is a mathematical representation of a generalized tissue with three subspaces, denoted by the subscripts 1, 2, and 3. This model expresses changes in mass in terms of blood and tissue concentrations, and general flux and biotransformation terms. In this representation, biotransformation (metabolism/excretion) can only occur in the "cellular" subspace. This representation also permits this set of equations to be used for modeling flow-limited and membrane-limited configurations by setting appropriate terms equal to zero.

Table B describes the characteristics of the possible tissue configurations admitted by the generalized 3-subspace model. The number of subspaces, presence or absence of biotransformation and flux terms, type of flux (ACTive or PASsive), and type of biotransformation (LINear or Michaelis-Menten) are specifically enumerated.

Table C is the S-system representation corresponding to the linear system of Table A. This set of three S-system equations can describe all possible configurations for a 3-subspace tissue. In the S-system approach the different configurations are admitted by altering the values of the parameters according to the rate laws that are in effect for a particular configuration.

Our "PB-PK" modeling and simulation system is a flexible and generic PB PK modeling and simulation system developed for the Macintosh™ computer. The user interactively specifies: 1) the anatomical structure of the model (tissues) and the anatomical structure of each tissue (*i.e.*, the parameters of the vascular, interstitial, and intracellular subspaces); 2) physiological relationships (blood flow rates for each tissue, metabolism and excretion of the substance); 3) transport characteristics, which also entails identification of flow- and membrane-limitations; and 4) thermodynamic properties of the substance (tissue partition coefficients).

The graphical user interface closely adheres to the human interface guidelines proposed by Apple Computer (1987).

The application has four independent interactive windows: **Model**, **Parameter**, **Kinetics**, and **Solution**. The content of each window can be printed, and the model (including parameters) and simulation data (sim-data) saved independently as files. The sim-data file format allows it to be exported to other graphics and/or statistical applications.

The user defines the anatomical model in the Model window (see Figure 1). This requires selection on a flow diagram consisting of a subset of the nine different tissues identified in the window: lung; heart; liver; gut; spleen; kidney; muscle; testes; and "other". There are four possible routes of exposure: intravenous (IV), intramuscular (IM), oral, and inhalation.

Parameters for the tissues are entered by means of dialog boxes (Figure 3). The user chooses the tissue configuration, depending on rate-limiting assumptions. The number of parameters to be specified in the dialog box is a function of this selection. An array showing the values of all the model parameters is displayed in the Parameter window.

Exposure route parameters are also entered *via* a dialog box. The dosage regimen (Figure 4) admits a bolus or continuous dose, with the user able to specify the time at which the dosing occurs, as well as the fraction, F, that is absorbed into the blood. Because of the modular format used in the development of this software, it will be possible to incorporate more complicated dosing regimens (*e.g.*, the universal elementary dosing regimen (Sebalt and Kreeft, 1987))

The Kinetics window (Figure 4) displays the results of the simulation once the model has been selected and the parameters entered. Dialog boxes are linked to this window to allow for configuration of the graph (time in hours/days, selection of which tissues or metabolites to graph, *etc.*), plotting of experimental data, and for any other parameters needed for solution of the set of differential equations.

The Solution window displays the resulting simulation data in a columnar format. The user can specify the frequency with which the time points are displayed (*e.g.*, every sixth time point).

The set of differential equations generated by the selection and specification of tissues are solved by incorporating the necessary modules from ESSYNS™, an interactive program written for the analysis of mathematical models expressed in S-system form (Irvine and Savageau, 1990; Voit *et al.*, 1989).

DISCUSSION

As in all simulation systems, our modeling system is dependent on external estimation of PK parameters used in the model. These estimates may be derived from: the literature; the investigator's previous experience; classical parameter estimation experiments; or reflect a hypothesized value. Although many physiological parameters are available in the literature, others, such as binding constants, frequently are not. When experimentation is not possible in humans the investigator must rely on *in vitro* or animal studies.

PB PK models are attractive for a number of reasons. First and foremost they are physiologically and anatomically correct. Second, they admit non-linear relationships. Third, they may be cast in the form of S-systems, thus making them mathematically tractable. Fourth, these systems may be easily modeled using our system. Finally, these models may be used to visually describe system dynamics and status through the graphical user interface. The classical approach to PK modeling relates dose and plasma concentration. The physiological approach goes one step further to relate dose, plasma, and tissue concentrations (Ritschel and Banerjee, 1986). Furthermore, it is adaptable to changing physiological circumstances and can allow for species-to-species and even subject-to-subject differences within the context of the physiological or anatomical parameters in the model (Himmelstein and Lutz, 1979). Perturbation of a particular parameter allows one to predict the changes in distribution or disposition of the drug during disease states, for instance, or in the presence of another drug. The combined effect of a number of complex inter-related processes can also be determined provided sufficient data are available (Ritschel and Banerjee, 1986).

SUMMARY

Physiologically-based pharmacokinetic modeling is rapidly gaining acceptance as a method for simulating tissue drug concentrations based on anatomical and physiological parameters and thermodynamic properties of the drug. Currently available software systems that use the physiologically-based philosophy are limited by the assumption of a particular type of physiologically-based model. Using a simulation language to define a complex model can be tedious. The Janszen-Miller "PB-PK" system is an interactive

generic physiologically-based pharmacokinetic model-ing and simulation system wherein specification and modification of the model is facilitated by the graphical user interface of the Macintosh™ computer. It allows great flexibilty in specifying a model, as well as ease of specifying the model parameters, and encourages "What if...?" scenarios. The user selects tissues for the model and an exposure route from an anatomical flow diagram or from a menu. Assumptions limiting the rate of mass transfer can be specified for each tissue. Parameters for each tissue, as well as dosage parameters, are entered *via* dialog boxes. The model is cast in an S-system format for ease of solution and for added flexibility in simulating inherently nonlinear biological systems. The system generates a steady-state solution, which can be plotted as multiple tissue concentration-time curves on a configurable graph. The system allows one to examine concurrent concentrations of a substance and its metabolite(s) within vascular, interstitial, and cellular components of a single tissue or organ; plot these values over time in the presence of single or repeated dosing; plot experimental data; and to generate data files for export to other graphics and statistics packages. The pictorial flow diagram, a table of all tissue parameter values, the steady-state solution set, and the graphic plots can be printed.

REFERENCES

Apple Computer, Inc. 1987. Human Interface Guidelines. Reading, MA, Addison-Wesley.

Ball, R., O. Skliar, and S.L. Schwartz. 1985. A systematic approach to the design of physiological pharmacokinetic models. *Fed. Proc.* **44**, 1121.

Blau, G.E. and W.B. Neely. 1987. Dealing with uncertainty in pharmacokinetic models using SIMUSOLV. In Pharmacokinetics in Risk Assessment. (National Research Council). Washington, D.C.: National Academy Press, pp. 185–207.

Bloch, R., G.D. Sweeney, K. Ahmed, C.J. Dickinson, and D. Ingram. 1980. 'MacDope': a simulation of drug disposition in the human body: applications in clinical pharmacokinetics. *Br. J. Clin. Pharmacol.* **10**, 591–602.

D'Souza, R.D. and H. Boxenbaum. 1988. Physiological pharmacokinetic models: some aspects of theory, practice and potential. *Tox. Ind. Health* **4**, 151–171.

Gabrielsson, J. and M. Hakman. 1986. MAXSIM: a new simulation program for computer assisted teaching of pharmacokinetics. *Am. J. Pharmaceut. Educ.* **50**, 35–38.

Himmelstein, K.J. and R.J. Lutz. 1979. A review of the applications of physiologically based pharmacokinetic modeling. *J. Pharmacokin. Biopharm.* **7**, 127–145.

Irvine, D.H., and M.A. Savageau. 1990. Efficient solution of nonlinear ordinary differential equations expressed in S-system canonical form. *SIAM J. Num. Anal.* **27**, 704-735.

Johanson G. and P.H. Näslund. 1988. Spreadsheet programming - a new approach in physiologically based modeling of solvent toxicokinetics. *Tox. Letters* **41**, 115–127.

Menzel, D.B., R.L. Wolpert, J.R. Boger III, and J.M. Kootsey. 1987. Resources available for simulation in toxicology: specialized computers, generalized software, and communication networks. In Pharmacokinetics in Risk Assessment. (National Research Council). Washington, DC: National Academy Press, pp. 229–250.

Ritschel, W.A. and P.S. Banerjee. 1986. Physiological pharmacokinetic models: principles, applications, limitations and outlook. *Meth. Find. Exptl. Clin. Pharmacol.* **8**, 603–614.

Savageau, M.A. 1969. Biochemical systems analysis. II. The steady-state solution for an n-pool system using a power-law approach. *J. Theor. Biol.* **25**, 370–379.

Sebalt, R.J. and J.H. Kreeft. 1987. Efficient pharmacokinetic modeling of complex clinical dosing regimens: the universal elementary dosing regimen and the computer algorithm EDFAST. *J. Pharm. Sci.* **76**, 93–100.

Voit, E.O. (ed.) 1991. Canonical Nonlinear Modeling: S-System Approach to Understanding Complexity. New York: Van Nostrand Reinhold.

Voit, E.O., D.H. Irvine, and M.A. Savageau. 1989. The User's Guide to ESSYNS. Charleston, SC: Medical University of South Carolina Press.

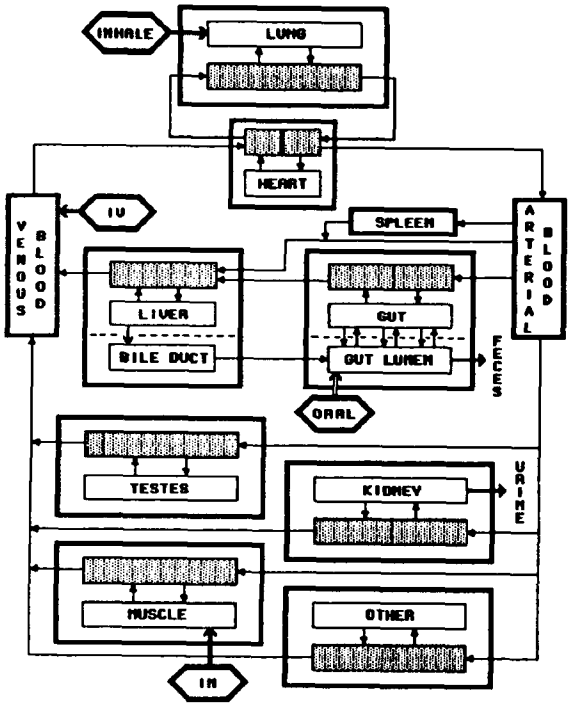


Figure 1 Flow scheme of a generic PB PK model

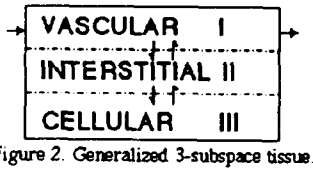


Figure 2. Generalized 3-subspace tissue.

$$V_1\dot{C}_1 = Q(C_p - \frac{1}{R}C_1) - \eta_1 - \tau$$

$$V_2\dot{C}_2 = \eta_1 - \eta_2 - \tau$$

$$V_3\dot{C}_3 = \eta_2 - \tau$$

Table A. Mathematical representation for a general 3-subspace tissue. ($\dot{C}$=dC/dt; V= volume, R=partition coefficient; η =general flux term; τ =general biotransformation term; p=plasma; 1,2,3=subspace)

Number of subspaces	FLUX I->II	FLUX II->III	BIOTRANSFORMATION I	BIOTRANSFORMATION II	BIOTRANSFORMATION III
1	–	–	–	–	–
1	–	–	LIN	–	–
1	–	–	MM	–	–
2	PAS	–	–	–	–
2	PAS	–	–	LIN	–
2	PAS	–	–	MM	–
2	ACT	–	–	–	–
2	ACT	–	–	LIN	–
2	ACT	–	–	MM	–
3	PAS	PAS	–	–	–
3	PAS	PAS	–	–	LIN
3	PAS	PAS	–	–	MM
3	PAS	ACT	–	–	–
3	PAS	ACT	–	–	LIN
3	PAS	ACT	–	–	MM
3	ACT	PAS	–	–	–
3	ACT	PAS	–	–	LIN
3	ACT	PAS	–	–	MM
3	ACT	ACT	–	–	–
3	ACT	ACT	–	–	LIN
3	ACT	ACT	–	–	MM

Table B Enumeration of biological processes for all possible configurations of a tissue (See text for details; a dash indicates a process does not occur)

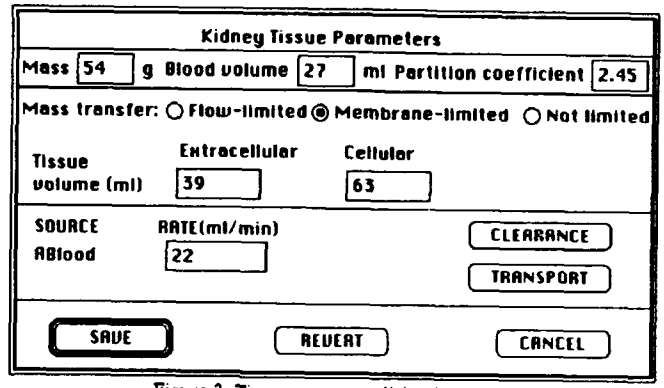


Figure 3 Tissue parameter dialog box

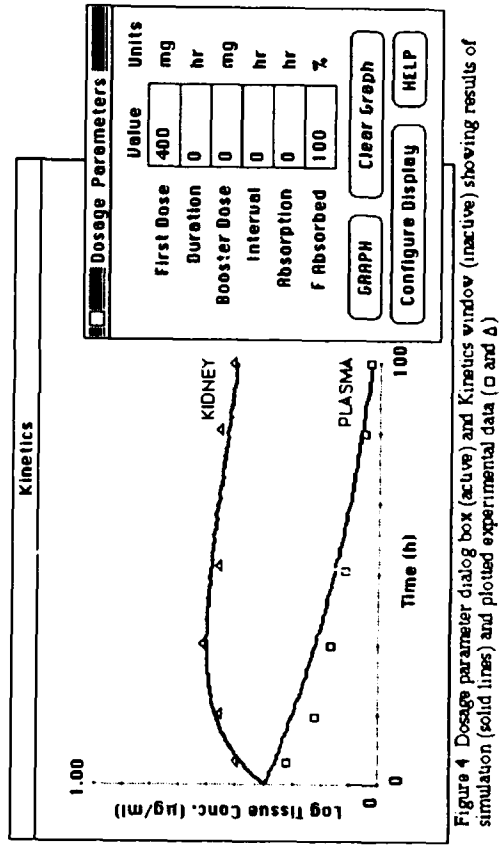


Figure 4 Dosage parameter dialog box (active) and Kinetics window (inactive) showing results of simulation (solid lines) and plotted experimental data (□ and Δ)

$$\dot{X}_1 = \alpha_1 X_p^{g_{11}} X_2^{g_{12}} - \beta_1 X_1^{h_{11}}$$

$$\dot{X}_2 = \alpha_2 X_1^{g_{21}} X_3^{g_{23}} - \beta_2 X_2^{h_{22}}$$

$$\dot{X}_3 = \alpha_3 X_2^{g_{32}} - \beta_3 X_3^{h_{33}}$$

Table C. S-system representation for a general 3-subspace tissue (g_{ij}, h_{ij} = kinetic order for all processes from jth space to ith space, α,β = rate constants, p = plasma, 1,2,3 = subspace)

Productivity at stake: Challenges for computing in the 1990's

By David A Olagunju, MS
Stephen C Smeach, Ph.D and Jack L James, MS, MBA, CDP
G.D. Searle, Skokie, Illinois

The computer technology in the 1990's will provide many opportunities to improve productivity. A company's strategy to integrate its existing technology with emerging technology will determine how well it takes advantage those opportunities. The goal of this paper is to discuss the primary factors that will impact on productivity within the computing environment. The discussion will center on coping with existing technologies, computing innovations, automation platforms and contemporary management issues.

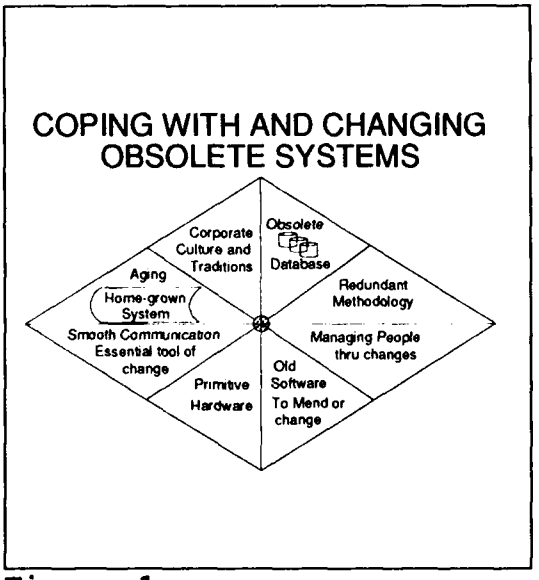


Figure 1

Coping with and changing obsolete systems.

One of the pressing challenges for management will be integrating existing systems with new technology. Many current applications have been developed in-house using methodologies that are now obsolete. For example, databases may have been developed when there was little or no formal database management system available. Failure to keep up with current technology through capital investment and continued education can lead to aging home-grown systems, housed in a collection of primitive hardware.

Perhaps one of the major obstacles to introducing new computing technology remains ineffective communication. Systems managers as well as users have communications responsibilities. Systems managers should be well informed of changes in computing technology and inform end users how it may benefit them. Users need to take the initiative to clearly define their application needs. In cooperation, these two groups can develop appropriate strategies to mend, change or replace existing systems with new technology. Additional communication is needed with the general user community so they understand how change will benefit them. Understanding the corporate culture and traditions will facilitate these communications. Figure 1, above, highlights the essential components for effective transition from obsolete systems to a new technology.

Computing Innovation.

Computing innovations in hardware, software and communication have revolutionized the way we process information. The development of fast computer chips and processors, the advent of new and flexible operating systems, and improvements to data communication provide greatly enhanced computing opportunities. Figure 2, below, illustrates some of the features of these emerging technologies.

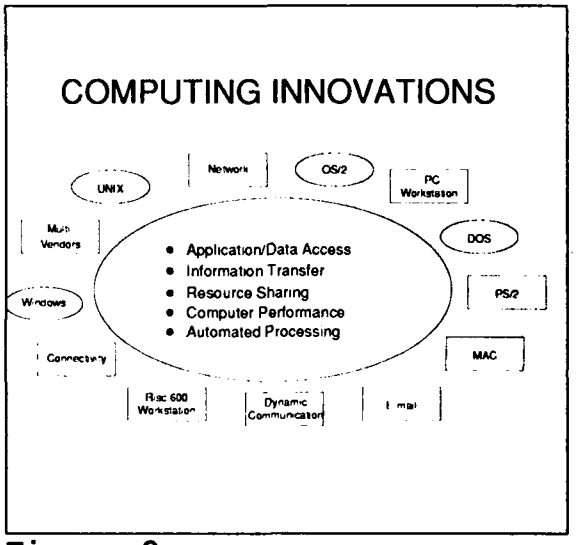


Figure 2

The challenge is to effectively employ these innovations to facilitate rapid application development, data access, faster information transfer, resource sharing, enhanced computer performance and automated processing for the benefits of computer users.

Productivity enhancement often requires large initial investments of time and capital. It is imperative that senior management understands the costs and benefits of computer enhanced productivity improvements and provide adequate funding.

Productivity in clinical information processing - an example.

Productivity enhancement in clinical information processing will involve both automation and data communication. The major factors (see Figure 3) that will impact these two areas are: the evolution of Integrated Services Digital Network (a technology that integrates data, voice and graphic information on digital lines), integration of remote and central processing capabilities and development of systems tolerant of different languages, software and hardware.

An example of productivity enhancements in clinical information processing is the development of the concept of remote study monitoring (RSM) and evolution of computing systems to support it. Traditional clinical information processing often involves a collection of remote site investigators who treat patients and fill out forms that describe their medical history and responses to therapy. These forms are usually collected or mailed to a central site for data entry, data editing and study conduct monitoring. Any discrepancies are mailed or telephoned back to the remote investigator site for resolution. This process is often complicated and time consuming. RSM technology has been developed so that data entry, editing, review and clean up can be done at the remote site in a very user-friendly manner. Data from the remote site can be automatically transferred to the central site overnight using modems and telephone lines. Study monitors at the central site can review the data and communicate with remote sites via electronic mail. In principle, such systems can eliminate some of the complications, reduce data errors and time delays in traditional clinical information processing systems. Implementation of these systems may involve all the factors impacting productivity and automation mentioned in Figure 3.

Other examples of productivity enhancing tools in clinical information processing include digital imaging and electronic note pad technologies. The first could be used to electronically convert documents to digital data. The second

could be used to directly enter clinical data on hand-held electronic note pads without the use of paper forms. Introduction of automated data processing should be a joint responsibility between systems managers and computer users as discussed above.

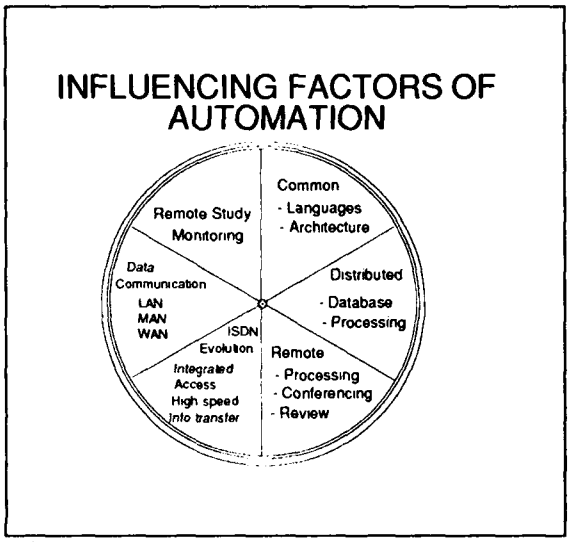


Figure 3

Contemporary Management Challenges.

The advent of the personal computer has dramatically changed the computer side of the work place for the knowledge workers. The human side of that work place has also changed. For example, there is more cultural diversity in offices today than there was ten years ago. Experts predict that this trend will continue into the future. Senior and middle managers must face the challenge of effectively managing groups of workers with different skills, backgrounds and motivations. Specifically, managers and supervisors need to stimulate and sustain motivation on the job, provide exciting career potential for their workers and in general provide a work place conducive to productivity enhancements. It is unfortunate that many organizations spend tens of thousands of dollars recruiting talented workers only to provide little or no challenge for such workers. It is encouraging to note that creative benefits are being introduced in the work place. Flexible work hours and educational assistance are two of these benefits. Figure 4, below, illustrates commonly introduced programs.

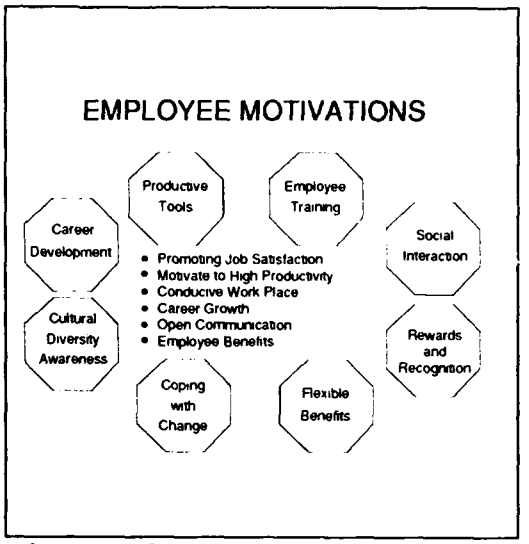


Figure 4

Conclusion.

The growth in technology and emphasis on productivity will put a tremendous amount of pressure on the knowledge workers of the 1990's. Unless properly managed, the result could be excessive stress and burnout, causing a decrease in productivity instead of an increase, lack of job satisfaction instead of sustained motivation and poor communication among peers and between supervisors and their subordinates. As the authors have described, successful implementation of new technologies to improve productivity requires clear understanding of existing systems and corporate culture, firm grasp of the benefits of new technologies, careful transition planning among systems managers and users, and management appreciation of the special needs of the knowledge workers.

A Comparison Of Some Robust Procedures For Estimating A Linear Discriminant Function

Hongzhe Li
Department of Mathematical Sciences
University of Montana
Missoula, MT 59812

A number of methods have been suggested for robustly estimating a linear discriminant function. These include substitution of robust estimates for the mean and covariance matrix and methods which choose a projection to maximize a robust measure of separation. This paper presents results of Monte Carlo simulations comparing some of these methods along with various modifications to see whether relatively simple methods works as well as complicated ones.

Introduction

For the two population discriminant analysis, if $f_1(.)$, $f_2(.)$ are the density functions of the underlying populations and assume equal costs, equal priors, then

$$\frac{f_1(x)}{f_2(x)} \quad > (<) \quad 1$$

gives the optimal discrimination rule.
If further we assume that $f_1(x)$ and $f_2(x)$ are normal with common covariance matrix, then we have

$$\beta' x > (\leq) \; c$$

where

$$\beta = \Sigma^{-1}(\mu_1 - \mu_2)$$

$$c = \frac{1}{2}(\mu_1 + \mu_2)'\Sigma^{-1}(\mu_1 - \mu_2)$$

In practice we use the sample mean $\bar{x}$ and pooled sample covariance matrix S for μ and Σ, and this gives the linear discriminant function (LDF) , which is widely used in practice. But the LDF is not robust to violations of the normality assumptions (Lachenbruch et al. 1973).

There are several approaches to deal with this problem:
1. Use nonparametric density estimates of $f_1(x)$,$f_2(x)$ (Koffler et al. 1978).
2. Transformation: e.g., rank transformation, normal score transformation (Conover & Iman 1980, Koffler et al. 1982).
3. Robust estimate of μ_1, μ_2, Σ in LDF: e.g. Huber-type M-estimates (Randles et al. 1978).
4. Estimate ß and c directly to obtain an optimal projection: e.g. nonmetric discriminant analysis(NDA) (Raveh 1989).

Of these four different procedures, nonparametric density estimates (#1) require large sample sizes and the algorithm is complicated; the disadvantage of transformations (#2) is that each time to classify a new observation, it is necessary to go back to find the rank or normal score of this new observation; projection methods (#4) are very difficult for more than a few variables. Robust substitution procedures (#3) are relatively simple and easy to compute and are the focus of this study.

The original purposes of this study were to
1. Compare effect of using different robust estimates of location and scale in LDF on misclassification rates.
2. Compare variability of misclassification rates under different procedures.

However, the result of 1 indicated that a very simple procedure which I called MLDF worked about as well as any of the other estimate procedures. Therefore, we added a third objective: compare the MLDF to other procedures of all types in the literature.

Some Results from Simulation

We used the following robust estimates of covariance in this study

$$Cov(X_i, X_j) = R(X_i, X_j) * MAD(X_i) * MAD(X_j)$$

where $R(X_i,X_j)$ is Pearson's r, Kendall's τ, Spearman's ρ, or greatest deviation correlation coefficient R_g (Gideon & Hollister 1987). $MAD(X_i)$ is the median absolute deviation. A Huber-type M-estimate for covariance was also used. Two robust estimates of location were used in addition to the mean: the median and a Huber-type M-estimate (Randles et al. 1978). We substituted these estimates of location and scale in LDF. In the simulation we considered only bivariate distributions, that is p=2. The distributional situations were normal, lognormal, mixture normal and bivariate Cauchy distributions. We found that for all these situations, the estimate of the covariance matrix had little effect on misclassification rates, at least

with the estimates we used. The median worked as well as M-estimates for location, and both were better than mean.

Results for lognormal, mixture normal and Cauchy distribution are reported in Table 1 and are representive of all the results.

For mixture normal situation, the two populations were:

$$\pi_1: 0.9*N\left[\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}4 & 1\\1 & 1\end{pmatrix}\right]+0.1*N\left[\begin{pmatrix}2.01\\0\end{pmatrix},\begin{pmatrix}400 & 100\\100 & 100\end{pmatrix}\right]$$

$$\pi_2: 0.9*N\left[\begin{pmatrix}2.01\\0\end{pmatrix},\begin{pmatrix}4 & 1\\1 & 1\end{pmatrix}\right]+0.1*N\left[\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}400 & 100\\100 & 100\end{pmatrix}\right]$$

Lognormal distributions were generated from independent normal with unit variance and mean (3,0) and (0,0). Bivariate Cauchy random variables were generated by the transformation $Y = Z/\mathrm{sqrt}(S) + \mu$, where Z is multivariate normal distribution with mean 0 and covariance Σ and S is $\chi^2(0)$ distribution with 1 degree of freedom (Johnson 1987). The underlying normal distributions to generate Cauchy distributions were:

$$\pi_1: N\left[\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}1 & 0.5\\0.5 & 1\end{pmatrix}\right]$$

$$\pi_2: N\left[\begin{pmatrix}1.78\\1.78\end{pmatrix},\begin{pmatrix}4 & 3\\3 & 9\end{pmatrix}\right]$$

A rank cutoff point was used instead of a zero cutoff point in LDF based on Randles et al. (1978).

Suggestion and Some Comparisons

Based on the results above we chose the following procedures for further study:

(1) MLDF procedure : Substitute median vector for the mean vector in LDF while still using S for Σ and with zero cutoff.

(2) RMLDF procedure: Substitute median vector for the mean vector in LDF while still using S for Σ but with rank cutoff (Randles et al. 1978). Rank cutoff point is used to balance the misclassification rates between two groups. We chose as a cutoff point a point such that the relative proportions of the misclassified observations of the two groups by the discriminant function scores were as equal as possible.

Table 1. Average Percentages Misclassified Using Different Location and Scale Estimators in LDF

		Location Estimator		
Covariance Estimator		Mean	Median	Huber-type Mean
Pearson	MN[a]	39.0	33.7	33.3
	LN[b]	12.4	13.4	12.0
	C[c]	41.6	34.8	35.7
Spearman	MN	39.2	32.5	31.9
	LN	11.8	12.0	11.5
	C	42.9	33.0	34.1
Kendall	MN	39.8	32.8	32.1
	LN	11.0	11.7	10.9
	C	42.5	33.0	33.8
Rg	MN	40.0	33.0	32.1
	LN	11.2	12.0	11.1
	C	42.7	33.0	33.6
Huber	MN			31.9
	LN			11.9
	C			34.7

[a] Mixture Normal
[b] Lognormal
[c] Cauchy

We compared these two procedures with several published studies.

(1) A Comparison with Randles et al. Study (1978)

Randles et al.(1978) introduced a generalization of LDF i.e. $R_\tau H$ procedure and also LDF with Huber-type M-estimate procedure using rank cutoff RLH. For $R_\tau H$ procedure, they took a nondecreasing, bounded odd function τ as a measure of separation and found the direction which maximizes this measure. They considered the distributional situations in Table 2. To their results, which are given in Table 3, we have added results for MLDF and RMLDF from a new simulation (with different random numbers). We see from this table that the RMLDF procedure works as well as the more complicated $R_\tau H$ procedure. In particular, consider the situational situation 8, where the distributions were contaminated, not only by changing the standard deviations but also changing the mean. The mean is a relatively bad estimate of location, but the median is not much affected by the outliers and thus produced relatively good estimates.

Table 2. Distributional Situations

	Population 1				Population 2			
	μ_1	μ_2	σ_1	σ_2	μ_1	μ_2	σ_1	σ_2
N[a]	0	0	1	1	1	1	1	1
N	0	0	1	1	1.78	1.78	2	3
LN[b]	1.65	1.65	1	1	2.65	2.65	1	1
LN	1.65	1.65	1	1	3.43	3.43	2	3
MN[c]	0	0	2	1	2.01	0	2	1
	0	0	20	10	2.01	0	20	10
MN	0	0	2	1	3.19	0	4	3
	0	0	20	10	3.19	0	40	30
MN	0	0	2	1	2.01	0	2	1
	2.01	0	20	10	0	0	20	10
MN	0	0	2	1	3.19	0	4	3
	3.19	0	40	30	0	0	20	10

[a] Normal
[b] Lognormal
[c] Mixture Normal (0.9 of first, 0.1 of second)

Table 3. Empirical Percentages Misclassified when n1 = n2 = 30

Situation	LDF	RLH	R_T H	RMLDF
1	29 29	29 29	29 29	28 30
2	17 33	27 28	28 28	24 29
3	22 31	26 26	26 26	26 27
4	14 40	25 26	26 26	26 27
5	40 35	33 29	34 30	33 32
6	23 44	30 31	33 34	30 33
7	40 39	33 31	36 32	34 33
8	41 37	30 31	32 33	30 31

Notes: 1. Maximum SE of estimates is 1.8.
2. Results for LDF, RLH and R_T H are from Randles et al.(1978).

(2) A Comparison with Koffler & Penfield(1978), Conover et al.(1980)

Koffler and Penfield (1978) used four nonparametric density estimation procedures: nearest neighbor (NN), Parzen and Cacoulos kernel estimator (P-C), Loftsgaarden and Quesenberry estimator (L-Q) and Gessaman (GESS) estimator in the lognormal distributional situations. Conover (1980) compared rank transformation method RLDF with these nonparametric procedures. These nonparametric procedures and RLDF procedure along with MLDF, RMLDF and Huber procedure were compared. Bivariate lognormal random variables were generated from independent normals with unit variance and means μ and 0 for population 1, and means 0 and 0 for population 2, where $\mu = 1,2,3$. The results appear in Table 4. For lognormal populations, RMLDF is clearly to be preferred over LDF. The MLDF and RMLDF also compares favorably with the nonparametric methods. But it seems that MLDF method doesn't work as effectively as the RLDF method.

Table 4. Percentage Misclassified when n1 = n2 = 64 (lognormal situations)[a]

	$\mu = 1$	$\mu = 2$	$\mu = 3$
LDF	34.1	26.6	22.5
NN	31.8	22.7	12.4
P-C	35.0	19.5	7.8
L-Q	34.4	17.5	7.0
GESS	30.9	17.5	12.4
RLDF	32.5	15.4	6.3
LDF	34.4	26.2	23.0
Huber (with rank cutoff)	33.5	18.5	11.9
MLDF	33.7	21.1	16.7
RMLDF	33.7	20.5	12.0

[a] Top part of table reproduces results from Koffler & Penfield(1978), and results from Conover & Iman (1980). Bottom part of table contains results of a new simulation.

Figure 1 displays the plots of the estimated standard deviation of misclassification rates versus average overall misclassification rate of the three procedures taken from several simulation situations. If two procedures have the same overall misclassification rate, but one has less variability in the misclassification rate, then the first procedure would be preferred.. The Huber-type M-estimate procedure and the RMLDF procedure have less variability of the misclassification rate than the LDF procedure.

Overall, the RMLDF method is simple and appaers to perform well relative to other nonparametric procedures.

Acknowledgement

I am grateful to Dr. David Patterson for the idea for this project and for useful suggestions which improved the presentation of this paper.

References

Conover, W.J., Iman, R.L. (1980), "The rank transformation as a method of discrimination with some Examples," Communications in Statistics, Theory and Method, A9(5) 465-487.

Johnson, M.E. (1987), "Multivariate Statistical Simulation", Wiley, New York.

Koffler, S.L., Penfield, D.A. (1978), "Nonparametric

discrimination procedures for nonnormal distributions," Unpublished manuscript.

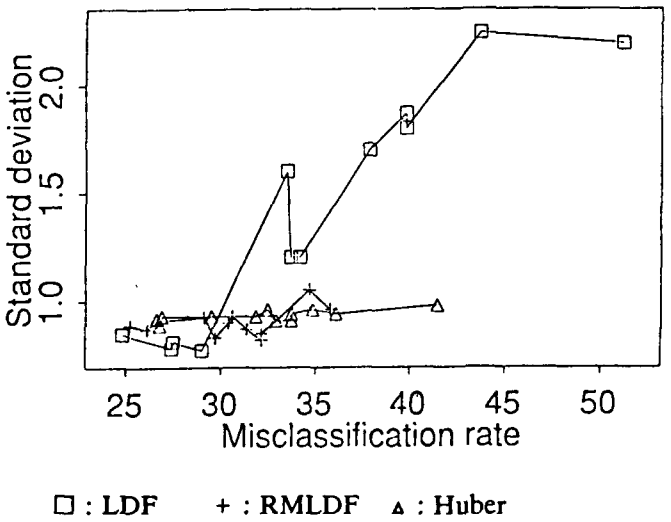
Koffler, S.L., Penfield, D.A. (1982),"Nonparametric classification based upon inverse normal scores and transformations," Journal of Statistical Computing and Simulation, 15, 51-68.

Lachenbruch, P. A., Sneeringer, C., and Revo, L. T. (1973), "Robustness of the linear and quadratic discriminant function to certain types of non-normality," Communications in Statistics, 1(1) 39-56.

Randles, R.H., Broffitt, J.D., Ramberg, J.S., Hogg, R.V. (1978), "Generalized linear and quadratic discriminant functions using robust estimates," Journal of the American Statistical Association, 73, 564-568.

Raveh, A. (1989), "A nonmetric approach to linear discriminant analysis," Journal of the American Statistical Association, 84, 176-183.

Figure 1. Standard Deviation of the Misclassification Rate vs. Average Misclassification Rate.



Robustness of Regression M-Estimators over Complex-valued Distributions

Krishnendu Ghosh
Department of Mathematical Sciences
University of Montana
Missoula, MT 59812

Richard M. Heiberger
Department of Statistics
Temple University
Philadelphia, PA 19122

Abstract

Noisy complex-valued data, for which robust regression techniques are the natural analysis approach, arise in many physical fields. Evaluation of the efficiency of such techniques requires that their behavior be charted over a series of known reference distributions. We have defined several symmetric long-tailed complex distributions (e.g., complex slash, complex Cauchy, complex double exponential) based on complex normal distribution. We have compared via the maximin method the robustness of different regression M-estimators (as defined by their weight functions) over these distributions. The variances of the estimators of the regression coefficients are obtained by simulation over all the distributions and for all the weight functions. The relative efficiencies over each distribution are obtained and then these relative efficiencies are compared over different distributions to identify the best weight function. Three different sample sizes 5, 11 and 15 have been used for this purpose. We apply our estimators to the evaluation of the Magnetotelluric response function.

KEY WORDS: Robustness; Regression; M-Estimators; Complex-Distributions.

1. Introduction

Many physical settings provide data for which linear regression is the physically appropriate analysis technique. In one such technique, the Magnetotelluric method, the complex-valued Fourier transforms of the electric and magnetic fields measured on the earth's surface are treated as the response and explanatory variables respectively. Robust techniques are needed to remove high leverage noise contamination in the electric field attributable to electrical activity in the ionosphere.

In this paper we use M-estimation, an iteratively reweighted least squares technique where the weight matrix w is a diagonal matrix with real positive weights. The distribution of the contaminating noise is not known; therefore the best function for producing the weights from the observed data is not known. In order to choose an appropriate weight function we must first explore the behavior of several weight functions with a variety of long-tailed complex symmetric distributions.

In Section 2 we briefly review the univariate complex normal. We then define several related univariate symmetric complex distributions.

In Section 3 we discuss M-estimation of the regression coefficients. We evaluate all the by now standard weight functions (Huber, Cauchy, Welsch, Logistic, Fair, Hampel, tanh, biweight, and Andrews), the Thomson weight function (Chave, Thomson and Ander, 1987), and introduce a new function that we call the Modified Thomson.

To find the best (robust) weight function for the M-estimation of the regression coefficients, we use in Section 4 the concept of relative optrim-efficiency. We compare the performance of the set of estimators over the set of long-tailed complex distributions by a simulation study.

Our recommended procedure for the M-estimation of a complex-valued regression coefficient is to use two different sets of iteration, each based on a different weight function. We have used this technique to improve the estimation of Magnetotelluric functions in a companion paper (Ghosh and Heiberger, 1991).

2. Symmetric Complex Distributions

We are interested in those complex random variables $Z = Z_R + iZ_I$ whose density functions $f_C(z)$ are real and equal to the real-valued bivariate density $g_R(z_R, z_I)$, that is

$$f_C(z) = g_R(z_R, z_I) \tag{2.1}$$

Denote the real-valued marginal densities of the real and the imaginary components of Z by $h_R(z_R)$ and $k_I(z_I)$. We also require that

$$h_R(u) = k_I(u) \tag{2.2}$$

We list in Table 1 nine different symmetric complex distributions ordered according to increasing tail weight (radius of the 93%ile), ranging from the almost-familiar complex normal to the heavy-tailed complex Cauchy with independent real and imaginary components.

We give in Section 2.1 Goodman's (1963) definition of the complex normal. We constructed the remaining

distributions and derived their density functions (Ghosh 1990). The derivations are straightforward applications of the transformation of variables method and are exceedingly tedious.

2.1 Complex Normal, *CN*

Goodman defined a complex normal random variable as a complex random variable whose real and imaginary parts are independent bivariate normal. Let Z follow the univariate complex normal, to be denoted $Z \sim CN(0,1)$. Both equations (2.1) and (2.2) are satisfied. The p.d.f. of Z is give by

$$f(z) = \frac{1}{\pi} e^{-(z_R^2 + z_I^2)} \qquad -\infty < z_R, z_I < \infty \qquad (2.3)$$

Independence of the real and imaginary components in the univariate complex normal has been assumed to allow easy extension to the multivariate complex normal.

2.2 Complex Cauchy with Independent Components, *CC(I)*

Let $X = (X_R + iX_I) \sim CN(0,1)$ and $Y = (Y_R + iY_I) \sim CN(0,1)$, independently. Then

$$Z = \left(\frac{X_R}{Y_R}\right) + i\left(\frac{X_I}{Y_I}\right) \qquad (2.4)$$

is a Complex Cauchy with Independent Real and Imaginary Components *CC(I)*.

2.3 Complex Cauchy with Dependent Components, *CC(D)*

Let $X = (X_R + iX_I) \sim CN(0,1)$ and $Y \sim N(0,1/2)$ independent of each other. Then,

$$Z = \left(\frac{X_R}{Y}\right) + i\left(\frac{X_I}{Y}\right) \qquad (2.5)$$

is Complex Cauchy with Dependent Real and Imaginary Components, *CC(D)*. Note that independence of the real and imaginary components is not required to satisfy conditions (2.1) and (2.2).

2.4 Complex Slash with Independent Components, *CS(I)*

Let $X = (X_R + iX_I) \sim CN(0,1)$ and Y_1, Y_2 both $\sim U(0,1)$ independent of each other and also independent of X. Then,

$$Z = \left(\frac{X_R}{Y_1}\right) + i\left(\frac{X_I}{Y_2}\right) \qquad (2.6)$$

is Complex Slash with Independent Real and Imaginary Components, *CS(I)*.

2.5 Complex Slash with Dependent Components, *CS(D)*

Let $X = (X_R + iX_I) \sim CN(0,1)$ and $Y \sim U(0,1)$ independent of each other. Then,

$$Z = \left(\frac{X_R}{Y}\right) + i\left(\frac{X_I}{Y}\right) \qquad (2.7)$$

is Complex Slash with Dependent Real and Imaginary Components, *CS(D)*.

2.6 Generalized Complex Slash, *GCS*

$Y = (Y_R + iY_I)$ is said to follow a univariate complex uniform distribution CU in a unit disk if its probability density function is given by $1/\pi$ for $|y|^2 \leq 1$. Let $X = (X_R + iX_I) \sim CN(0,1)$ and $Y = (Y_R + iY_I) \sim CU$(unit disk) independent of each other. Then,

$$Z = \left(\frac{X}{Y}\right) = \left(\frac{X_R Y_R + X_I Y_I}{|Y|^2}\right) + i\left(\frac{X_I Y_R - X_R Y_I}{|Y|^2}\right) \qquad (2.8)$$

has a Generalized Complex Slash, *GCS*, distribution.

2.7 Complex t Distribution, *CT*

Let $X = (X_R + iX_I) \sim CN(0,1)$ and $Y = (Y_R + iY_I) \sim CN(0,1)$ independent of each other. Then,

$$Z = \left(\frac{X}{Y}\right) = \left(\frac{X_R Y_R + X_I Y_I}{|Y|^2}\right) + i\left(\frac{X_I Y_R - X_R Y_I}{|Y|^2}\right) \qquad (2.9)$$

follows a Complex t distribution, *CT*, with 2 degrees of freedom.

Note that the familiar real variable definitions do not always generalize to complex variables in the anticipated way. The real-valued Cauchy distribution is defined as the ratio of two independent standard normal variables. But the ratio of two independent standard complex normal variables gives a complex-t distribution with 2 degrees of freedom, not a complex Cauchy. The complex Cauchy was given in Sections 2.2 and 2.3.

2.8 Complex Double Exponential Distribution, *CDE*

Let $X_j = (X_{jR} + iX_{jI}) \sim CN(0,1) \quad j = 1,2,3,4$ independent of each other. Then,

$$Z = (X_{1R}X_{2R} + X_{3R}X_{4R}) + i(X_{1I}X_{2I} + X_{3I}X_{4I}) \qquad (2.10)$$

has a Complex Double Exponential distribution, *CDE*.

2.9 Complex Logistic Distribution, *CL*

The *CL* distribution is defined so that the joint distribution of the real and imaginary parts follows a bivariate

logistic distribution and each of the real and imaginary components follows a real univariate logistic distribution.

3. Regression M-Estimators and Weight Functions

The M-estimate, or maximum likelihood type estimate, T_N based on a sample $(x_1, x_2, \cdots, x_N)$ of size N, is the value of t that minimizes the objective function $\sum_{j=1}^{N} \rho(x_j - t)$. The loss function ρ is assumed to be continuous, and has derivatives with respect to t at all values of t. We calculate T_N by finding the value of t that satisfies the equation $\sum_{j=1}^{N} \psi(x_j - t) = 0$ where $\psi(u) = \frac{d}{du}\rho(u)$.

Let us consider the linear regression model

$$\underset{N\times 1}{y} = \underset{N\times q}{X} \quad \underset{q\times 1}{\beta} + \underset{N\times 1}{r} \tag{3.1}$$

The M-estimation process minimizes a norm of residuals, as does the least squares process. But the misfit measure in M-estimation is chosen so that a few extreme values cannot dominate the answer. The M-estimate is obtained by solving

$$\min_\beta \sum_{j=1}^{N} \rho\left(\tfrac{r_j}{d}\right) = \min_\beta R^H R \tag{3.2}$$

where minimization is done with respect to β, and R is a $N \times 1$ vector whose jth element is $\sqrt{\rho(\frac{r_j}{d})}$, $r_j = y_j - X_j\beta$ is the jth residual, and d is a scale factor. In the special case with $\rho(u) = u^2$ and $d = 1$, M-estimation specializes to least squares estimation.

Equation (3.2) yields solutions of the non-linear system

$$X^H \Psi = 0 \tag{3.3}$$

Table 1. Interquartile diameter σ_{IQ} and the 50th, 80th, 91st and 93rd quantile radii for complex distributions.

Distributions	σ_{IQ}	50th	80th	91st	93rd
CN	1.66	0.83	1.27	1.55	1.64
GCS	2.52	1.26	2.16	2.35	2.37
CS(D)	3.50	1.75	2.47	2.53	2.54
CDE	2.70	1.35	2.43	3.27	3.59
CT	2.00	1.00	2.00	3.16	3.74
CL	3.72	1.86	3.11	4.08	4.46
CC(D)	3.46	1.73	4.90	10.95	14.97
CS(I)	4.23	2.11	5.60	12.39	16.91
CC(I)	4.40	2.20	6.23	13.97	19.31

where Ψ is a $N \times 1$ vector whose jth element is the influence function $\psi(\frac{r_j}{d})$. We solve equation (3.3) by expressing it as a weighted least squares problem

$$X^H w r = 0 \tag{3.4}$$

where r is $N \times 1$ residual vector, w is $N \times N$ diagonal matrix of weights whose jth diagonal element is $w_j = \psi(\frac{r_j}{d})/(\frac{r_j}{d})$. The solution to equation (3.4) is given by iteratively solving

$$\widehat{\beta} = (X^H w X)^{-1}(X^H w y) \tag{3.5}$$

The weights at each iteration are computed from the residuals and scale estimate of the previous iteration.

A practical choice of the scale factor d is $\frac{S_{IQ}}{\sigma_{IQ}}$. S_{IQ} is the sample interquartile diameter of the complex residuals and σ_{IQ} is the population interquartile diameter of the underlying distribution of r.

3.1 Modified Thomson Weight Function (M-Thomson)

Thomson's weight function is different from the others listed in Section 1 because it is data adaptive. The quantity α in Thomson's weight function is the nth quantile of the assumed underlying distribution. The point at which the downweighting begins depends on both the underlying distribution and the sample size n.

We found that Thomson's weight function is robust to several underlying distributions but does not work quite well enough for very heavy-tailed distributions. We therefore proposed a new weight function, a modification of the Thomson weight function:

$$w(u) = \alpha^{(-e^{\alpha(|u|-\alpha)})} = \exp\left[(\ln\alpha)(-e^{\alpha(|u|-\alpha)})\right] \tag{3.1}$$

Table 1 displays the α values for the 80th, 91st and 93rd quantiles of the complex distributions (corresponding to $n = 5, 11, 15$).

The advantage of the M-Thomson weight function over Thomson's function comes from the change in the base of the exponential as α changes. It downweights the potential outliers as the sample size increases to a greater extent than does Thomson's weights. For small-tailed distributions like *CN*, *GCS* and *CS(D)*, the M-Thomson function puts more weight on the valid data and also protects non-outliers from too much down-weighting. For mid-size distributions like *CDE*, *CT* and *CL*, M-Thomson's weight function rapidly downweights data points that are beyond the nth quantile. For large-tailed distributions like *CC(D)*, *CS(I)* and *CC(I)*, the M-Thomson and Thomson weight functions give almost identical results.

4. A Simulation Study

We have evaluated the weight functions listed in Section 1 over the distributions defined in Section 2.

In order to find the best M-estimator (weight function) of the regression coefficients for complex-valued data we compare the different weight functions using the *maximin* approach. Our presentation is based on Chapters 10 and 11 of Hoaglin, Mosteller and Tukey (1983). We have s different weight functions $(w_1(u), w_2(u), \cdots, w_s(u))$ and therefore s estimators of β $(\widehat{\beta}_{w_1}, \widehat{\beta}_{w_2}, \cdots, \widehat{\beta}_{w_s})$. We want to investigate which estimator among these is the most efficient over a wide range of distributions. The optrim-efficiency for a specified estimator is the ratio of the variance of the best estimator (the optrim) for a given distribution to the variance of the specified estimator. With weighted least squares estimators $\widehat{\beta}_{w_i}$ the optrim-efficiency is

$$\text{Optrim-Eff}\left(\widehat{\beta}_{w_i}\middle|F\right) = \frac{\left\|(X^H w_k X)^{-1}\right\|}{\left\|(X^H w_i X)^{-1}\right\|} \qquad (4.1)$$

where the notation w_k means the diagonal matrix whose jth diagonal element is $w_k(|r_j/d|)$.

4.1 Procedure

We did a maximin analysis of the optrim-efficiencies for three different sample sizes 5, 11 and 15 and nine different symmetric complex distributions, 27 different sample size–distribution combinations in all. The procedure we follow in calculating optrim-efficiencies for each sample size is: (1) determine by simulation the estimator variance for each weight function and distribution, (2) calculate the minimum variance over weight functions for each distribution, (3) calculate the optrim-efficiencies for each combination of weight function and distribution, (4) find the minimum efficiency over the distributions for each of the weight functions, and (5) find the maximum over the estimators of the minimum efficiencies.

Simulation of numbers from the various complex distributions is straightforward since the real uniform and real normal are available in all software libraries.

4.2 Observations

We find difficulties with most of the standard weight functions in most long-tailed complex situations. In particular, Tukey's biweight and Andrews' wave functions fail for long-tailed distributions. The redescending Huber and Hampel weight functions behave differently from the rest. They may assign full weight to potential outliers. Thomson's weight function is robust to several underlying distributions but does not work quite well enough for long-tailed distributions. The Modified Thomson function is often the best, dominating all the others except with the very long-tailed complex Cauchy distributions where it gives results similar to the Thomson function.

4.3 Recommendations

In order to estimate a complex-valued regression coefficient using the M-estimation technique, we use two different sets of iterations. In the first set of iterations we choose a weight function, usually the redescending Huber or Hampel weight function so as not to reject too many outlier points too early, and iterate until the residual norm $|r^H r|$ does not change appreciably. In the second set of iterations we choose another weight function dependent on sample size and iterate it similarly until we get the desired convergence. For most sample sizes we looked at, the M-Thomson weight function seems to dominate. Other good choices for the second set are the Thomson, logistic, or hyperbolic tan functions.

Acknowledgement

The work reported here is based on the first author's Ph.D. dissertation at Temple University. The authors would like to thank Alan Chave Department of Physics, AT&T Bell Laboratories, Murray Hill, NJ for bringing the problem to their attention.

References

1. Chave, A. D., Thomson, D. J. and Ander, M. E. (1987), "On the Robust Estimation of Power Spectra, Coherence, and Transfer Function", *Journal of Geophysical Research*, Vol. 92, No. B1, 633–648.
2. Ghosh, K. (1990), "Robust Multivariate Regression Analysis of Complex-Valued Data", Ph.D. dissertation, Temple University.
3. Ghosh, K. and Heiberger, R. M. (1991), "Robust Multivariate Regression Analysis of Complex-Valued Data: Comparison of the Jackknife and the Complex Normal Approaches", In Preparation.
4. Goodman, N. R. (1963), "Statistical Analysis Based on a Certain Multivariate Complex Gaussian Distribution (An Introduction)", *Annals of Mathematical Statistics*, Vol. 34, 152–177.
5. Hoaglin, D. C., Mosteller, F. and Tukey, J. W. (1983), *Understanding Robust and Exploratory Data Analysis*, Wiley.

Computing Multivariate L^1 Regression Estimates

George R. Terrell
Statistics Department, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

Abstract: Minimum total error, or L^1, regression estimates are a generalization of the sample median to prediction problems. Multivariate extensions therefore involve the concept of a multivariate median. There are many inequivalent characterizations of a multivariate median in the literature, all of which seem to have at least one of two major difficulties: either they lack the property of affine covariance which we have come to expect from ordinary multivariate regression, or they are computationally highly unpleasant. We here propose a definition of multivariate median, inspired by the theory of M-estimation, that transforms appropriately under linear changes of variables. Furthermore, it may be computed straightforwardly using a fixed-point property. The result is a resistant multivariate regression estimate that is intuitively appealing and, surprisingly, increasingly efficient at the normal model in higher dimensions. We share some computational experience with this estimator.

I. Introduction

Linear regression is perhaps the central tool of modern statistics; it seeks predictive models of the form $\hat{y} = Xb$. The classical criterion for fitting this model goes back at least to Legendre, the method of least squares:

$$\min_{\mathbf{b}} \sum_{i=1}^{n} (y_i - X_i\mathbf{b})^2$$

The simplicity and power of this procedure is unexcelled. However, in modern times, statisticians have become increasingly concerned with the lack of robustness of the least-squares technique—its sensitivity to a few observations for which the model fit is very poor. Perhaps the oldest technique for dealing with this problem (predating even least squares, see e.g. Stigler [1986]) is the minimum total error, or L^1, criterion:

$$\min_{\mathbf{b}} \sum_{i=1}^{n} |y_i - X_i\mathbf{b}|$$

The naturalness of this method is to some extent offset by its greater computational difficulty and by its relatively low efficiency at the normal model. However, it is robust—there is an upper bound on how much influence any poorly-fitting observation can have on predictions. Thus, the minimum total error criterion has attracted considerable recent attention.

Extension of the linear model to several dependent variables, multivariate regression, turns out to be straightforward using least-squares. However, extension of the least total error criterion to several dimensions turns out to be more problematic. Consider the simplest case of regression, the location problem $\min_{\mu} \sum_{i=1}^{n} |y_i - \mu|$. It is standard that the solution μ is the sample median. Similarly, the solution of the least squares location problem is the sample mean. When we proceed to several variables, it is still true in every sense that the multivariate sample mean, with the obvious definition, is the solution of the least squares problem. However, even the appropriate definition of a multivariate median is problematic; so that it is not obvious what is meant by multivariate L^1 regression.

A number of possible definitions of multivariate median are discussed in Small [1990]. Perhaps the simplest is the vector of medians of each coordinate by itself. This corresponds to solving the least total error problem for each dependent variable separately. For some applications this may be reasonable, but it has one obvious major flaw: if we take a rotation of the dependent variables, it is not generally true that the median of the rotated data is the rotation of the median. Another simple definition of the multivariate median is obtained by extending minimum total error to a minimum total distance criterion:

$$\min_{\mu} \sum_{i=1}^{n} |y_i - \mu|$$

This approach, called the *L^1 median*, which dates back at least to Weber [1909], amounts to choosing a point in space so that the stars are scattered as uniformly as possible over the celestial sphere. It is obviously unaffected by rotations. Unfortunately, this idea for a median fails to transform nicely if we rescale one of the coordinates differently from the others. For example, if one variable is in inches and the other in dollars, changing the scale on the first axis to centimeters will change the median in a nonobvious way. Since in statistical practice our coordinates are often inhomogeneous, the applicability of *this definition is too limited.*

One of the great virtues of the median is its covariance under any monotone transformation. It is not clear that this desideratum is achievable for any multivariate location measure. However, it is certainly desirable, as our two examples suggest, to have a multivariate median covariant under as rich as possible a set of transformations of the data. For example, the mean is *affine*, that is $E(\mathbf{a} + B\mathbf{x}) = \mathbf{a} + BE(\mathbf{x})$. Arbitrary linear changes of variables adjust the mean in the obvious way. We shall therefore restrict our attention to definitions of multivariate median that are affine; a number of these are discussed in Small's survey. However, all of these concepts have at least one of two serious drawbacks. Either they are

rather difficult to compute, or they do not generalize in any obvious way to a definition of multivariate median for distributions.

We shall propose an affine location criterion, the *m-median*, inspired by Huber's m-estimates, that is plausibly a multivariate generalization of the ordinary median. It will have an obvious characterization on distributions; and we will propose a reasonably efficient method for computing it. It has the nice property that it becomes more nearly efficient at the normal model as the dimensionality increases. Extension to a general tool for robust multivariate regression will be straightforward.

II. Multivariate L^p Location Estimation

Least squares and minimum total error are each special cases of the L^p location estimate which solves $\min_{\mu} \sum_{i=1}^{n} |y_i - \mu|^p$. It is this which we shall generalize to several variables. Following the lead of Gauss, we recognize that multivariate least squares arises as maximum likelihood estimates of the parameters in the multivariate normal family of densities

$$\frac{1}{(2\pi)^{d/2}(\det V)^{1/2}} e^{-(\mathbf{x}-\mu)^{\mathsf{T}} V^{-1}(\mathbf{x}-\mu)/2}$$

The family is affine, with the transformation rule for the mean as above and for the covariance matrix $V' = BVB^{\mathsf{T}}$; the location estimate is the multivariate mean and the scale estimate is the sample covariance matrix. This suggests a natural way to achieve an affine L^p location statistic: let it be the maximum likelihood estimate of the parameters in the family

$$\frac{c}{(\det V)^{1/2}} e^{-\left|(\mathbf{x}-\mu)^{\mathsf{T}} V^{-1}(\mathbf{x}-\mu)\right|^{p/2}/b}$$

where b will be chosen later (it is essentially arbitrary, but we need a consistent choice), and c is the constant that lets the family integrate to one (we will never need to compute it). The maximum likelihood criterion for estimating this from an i.i.d. sample of n random vectors is

$$\min_{\mu, V} \frac{n}{2} \log \det V + \frac{1}{b} \sum_{i=1}^{n} \left|(\mathbf{x}_i-\mu)^{\mathsf{T}} V^{-1}(\mathbf{x}_i-\mu)\right|^{p/2}$$

The solution μ to this problem will constitute our definition of an affine L^p location statistic. The following partial result is immediate: for a fixed nonsingular V a solution for μ exists and the collection of solutions is convex. If the observations do not lie in a hyperplane, then for fixed μ a solution V exists. Any joint solution is affine: the transformed solution is a solution for the transformed sample.

Notice that an extension of this definition to one for distributions is immediate—simply replace the sums with expectations. If the random vector possesses elliptical symmetry, then μ, if it exists, is the center of symmetry. V, if it exists, is a multiple of the quadratic form that characterizes the elliptical symmetry.

We are left to decide on an appropriate value for b. From the definition, it is clear that this decision has no effect on the definition of μ. However, a definite solution for V will be useful in various inferences about our model, and b scales V. If our goal were primarily robustification of the normal model, we would choose the constant so that V coincided with the covariance of a multivariate normal random variable. For the general problem of an appropriate definition of LP scale, the normal family plays no special role. Therefore, I propose the following criterion for deciding b: For a distribution uniform on the unit sphere, let V be identical to the ordinary covariance matrix; that is, $\frac{1}{d} I$, where I is the identity matrix. Then our scale behaves predictably on the simplest affine family, one out of which all others may readily be built. A calculation gets $b = p / d^{1-p/2}$.

III. The M-median

The case $p=1$ of a multivariate L^p location estimate will give us our desired affine multivariate generalization of the median.

Definition: The *m-median* is any vector μ which solves

$$\min_{\mu, V} \frac{n}{2} \log \det V + \sqrt{d} \sum_{i=1}^{n} \left|(\mathbf{x}_i-\mu)^{\mathsf{T}} V^{-1}(\mathbf{x}_i-\mu)\right|^{1/2}$$

The definition for distributions is analogous. The m-median is affine, but coincides with the L^1 median for spherically symmetric data. In particular it is the ordinary median in the case of one variable. Notice that it is not defined if the observations all fall in a hyperplane. In that case, use the definition that applies to the smallest-dimensional hyperplane that contains all observations.

One fact is immediate: given $d+1$ noncohyperplanar vectors, an m-median is at the barycenter of the simplex they form. For, we may transform them to the corners of a regular simplex, where the result follows from symmetry, then transform back.

Milasevic and Ducharme [1987] have shown that the L^1 median is unique for noncolinear data. But then the m-median is also unique in this case, as we may transform to the case where V is a multiple of the identity and so the two definitions coincide.

Proposition: The relative efficiency of the m-median to the mean in the multivariate normal case is

$$\frac{2}{d} \frac{\Gamma\left(\frac{d+2}{2}\right)^2}{\Gamma\left(\frac{d+1}{2}\right)^2}$$

The proof involves computing the expected square of the infinitesimal influence function of the statistic after transform-

ing to the spherically symmetric case. This generalizes a result of Brown [1983] for the L^1 median. Here are some special cases:

Dimension	Efficiency
1	0.6366
2	0.7854
3	0.8488
4	0.8836
∞	1.0000

Thus, the m-median becomes more nearly efficient in higher dimensions, and because of its robustness (since the influence function of a point is bounded) is a worthy competitor to classical measures of location.

One interesting phenomenon should be noted: since the m-median is covariant under arbitrary linear changes of coordinates, it is afflicted by a sort of nonrobustness in certain cases. If all but a few observations lie in a hyperplane, points off that hyperplane may be arbitrarily influential on the coordinates of the m-median in the directions orthogonal to the hyperplane. This seems unavoidable for nontrivial affine statistics.

IV. Computing the Estimates

In the case $p=2$, we have closed form estimates for the multivariate mean and the covariance matrix. For computing the general affine L^p location statistic, we need

Theorem: A fixed point for the affine L^p location fitting criterion is given by

$$\mu = \frac{\sum_{i=1}^{n} \frac{\mathbf{x}_i}{\left\|(\mathbf{x}_i-\mu)^T V^{-1}(\mathbf{x}_i-\mu)\right\|^{1-p/2}}}{\sum_{i=1}^{n} \frac{1}{\left\|(\mathbf{x}_i-\mu)^T V^{-1}(\mathbf{x}_i-\mu)\right\|^{1-p/2}}}$$

$$V = \sum_{i=1}^{n} \frac{(\mathbf{x}_i-\mu)(\mathbf{x}_i-\mu)^T}{\left\|(\mathbf{x}_i-\mu)^T V^{-1}(\mathbf{x}_i-\mu)\right/d\|^{1-p/2}}$$

These were derived by variation of the parameters. Our algorithm is to obtain a starting estimate for μ and V (for example, the mean and covariance), then iterate until the estimates do not change significantly. In a large number of trials this converged in all cases where $1 \le p \le 2$. Our algorithm for computing the m-median is then just the case $p=1$. Surprisingly, this procedure was successful even in the case of the univariate median, though it converged very slowly and is no competition for the usual median algorithms. In higher dimensions, it usually converged moderately rapidly; getting 6 significant figures in perhaps 20 iterations. The exceptions to this were usually cases in which μ coincided with a data point; then convergence was very slow. Presumably the algorithm could be modified to recognize this special case. Except in one dimension, it seems to be very unusual for the location to coincide with a data point.

Scott *et al* [1978] report the serum cholesterol and triglyceride levels for 320 males who reported chest pain. The sample mean was cholesterol 216.19 and triglycerides 179.35, with a correlation of .228. After fewer than 20 iterations we found the m-median was cholesterol 212.71 and triglycerides 156.37 with a "correlation" of .240. A few very high triglycerides levels apparently distorted the typical value, and even diluted the correlation slightly. The figure shows a sparse histogram of this data set (with several extreme cases unfortunately censored). The digit 2 indicates the mean and 1 the m-median.

The extension to L^p multivariate regression is straightforward: replace μ by a linear model with one or several independent variables, and V is then a sort of covariance matrix of the multiple residuals. The first fixed point equation becomes a system of weighted normal equations; our method is thus a special case of iteratively reweighted least-squares. Tests and confidence statements for such a method raise a number of interesting questions, which will be dealt with in a later paper.

V. Bibliography

Brown, B.M. (1983). Statistical uses of the spatial median. *Journal of the Royal Statistical Society* B 45 pp. 25-30.

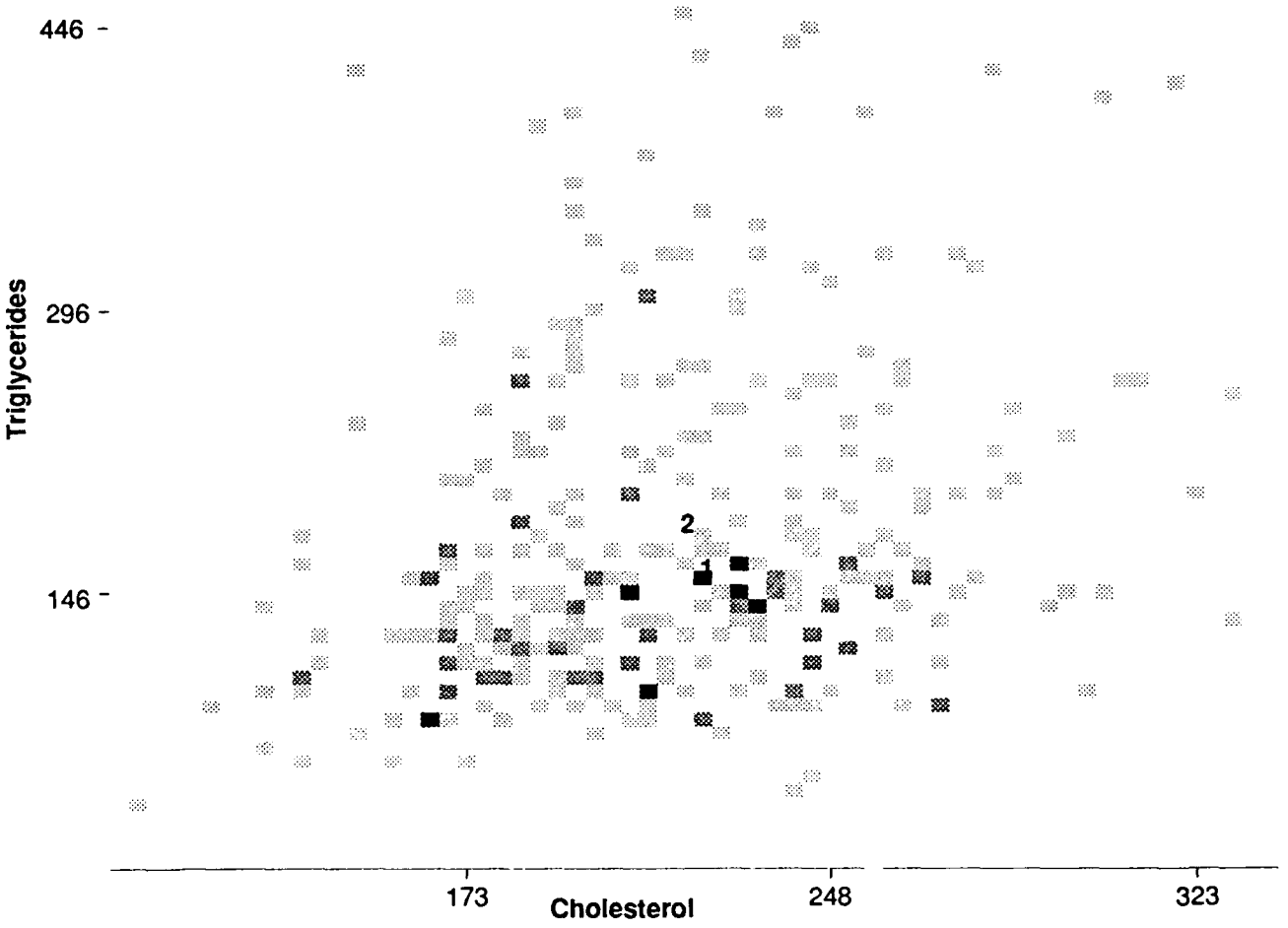
Milasevic, P. and Ducharme, G.R. (1987). Uniqueness of the spatial median. *Annals of Statistics* 15 pp. 1332-3.

Scott, D.W., Gotto, A.M., Cole, J.S., and Gorry, G.A. (1978). Plasma lipids as collateral risk factors in coronary artery disease. *Journal of Chronic Diseases* 31 pp. 337-345.

Small, C.G. (1990). A survey of multidimensional medians. *International Statistical Review* 58 3 pp. 263-277.

Stigler, S.M. (1986). *The History of Statistics: the measurement of uncertainty before 1900.* Belknap Press, Cambridge, Massachusetts.

Weber, A. (1909). Über den Standort der Industrien, Tubingen. English translation by Freidrich, C.J. (1929). *Alfred Weber's Theory of Location of Industries.* University of Chicago Press.



Serum Lipids of 320 Men with Chest Pain
Scott, *et al*, (1980)
(8 observations outside bounds of graph)

Validating a Large Geophysical Data Set: Experiences with Satellite-Derived Cloud Parameters

Ralph Kahn, Robert D. Haskins, James E. Knighton,

Andrew Pursch, and Stephanie Granger-Gallegos

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

Abstract

We are validating the global cloud parameters derived from the satellite-borne HIRS2 and MSU atmospheric sounding instrument measurements, and are using the analysis of these data as one prototype for studying large geophysical data sets in general. The HIRS2/MSU data set contains a total of 40 physical parameters, filling 25 MB/day; raw HIRS2/MSU data are available for a period exceeding 10 years. Validation involves developing a quantitative sense for the physical meaning of the derived parameters over the range of environmental conditions sampled. This is accomplished by comparing the spatial and temporal distributions of the derived quantities with similar measurements made using other techniques, and with model results.

The data handling needed for this work is possible only with the help of a suite of interactive graphical and numerical analysis tools. Level 3 (gridded) data is the common form in which large data sets of this type are distributed for scientific analysis. We find that Level 3 data is inadequate for the data comparisons required for validation. Level 2 data (individual measurements in geophysical units) is needed. A sampling problem arises when individual measurements, which are not uniformly distributed in space or time, are used for the comparisons. Standard 'interpolation' methods involve fitting the measurements for each data set to surfaces, which are then compared. We are experimenting with formal criteria for selecting geographical regions, based upon the spatial frequency and variability of measurements, that allow us to quantify the uncertainty due to sampling. As part of this project, we are also dealing with ways to keep track of constraints placed on the output by assumptions made in the computer code. The need to work with Level 2 data introduces a number of other data handling issues, such as accessing data files across machine types, meeting large data storage requirements, accessing other validated data sets, processing speed and throughput for interactive graphical work, and problems relating to graphical interfaces.

KEY WORDS: large data sets, validation, satellite data analysis

1. Introduction

NASA's Earth Observing System (EOS) will generate vast quantities of data. Hundreds of terabytes of data will be acquired from orbit to characterize the Earth's environment with the kind of spatial and temporal detail needed to study climate change. Such high resolution is required to properly sample the non-linear impact of small-scale phenomena, which can make significant contributions to the global-scale budgets of heat and momentum. It is also expected that the data will be analyzed not just in the traditional manner, concentrating on a single data set at a time, but in new ways that involve routinely comparing data sets from multiple sources. Part of the need to study multiple data sets comes from a growing appreciation for the importance to global conditions of transports across boundaries such as the air-ocean interface (e.g., Earth System Science Committee, 1988).

We are undertaking the validation of cloud parameters derived from the High Resolution Infrared Radiation Sounder 2 (HIRS2) and the Microwave Sounding Unit (MSU) instruments aboard the NOAA polar orbiting meteorological satellites. The instruments provide one of the few global measures of cloud properties extending over many years. They are also capable of obtaining near-simultaneous constraints on the physical characteristics of the atmosphere and surface needed to derive cloud properties. One goal of this work is to learn about analyzing large geophysical data sets in general.

Radiances from the HIRS2 and MSU instruments have been analyzed by Susskind and co-workers using an algorithm that accounts self-consistently for the first-order physical quantities affecting the emergent radiation (Susskind et al., 1984; 1987). The standard data products are (1) monthly mean values for forty meteorological parameters, including effective cloud amount and effective cloud top height, on a grid of boxes 2 degrees in latitude by 2.5 degrees in longitude, and (2) 'daily data' with twice-daily temporal sampling, a spatial resolution of about 125 km, and spacing between points of about 250 km. The monthly mean data are referred to as a 'Level 3' (gridded) product, and the daily

data is called a 'Level 2' product (individual measurements reduced to geophysical units) (Space Science Board, 1982; EOS Data Panel, 1986). The size of the uncompressed Level 3 data is about 4 MB/month, whereas the Level 2 product fills about 25 MB/day (750 MB/month).

By validation we mean 'developing a quantitative sense for the physical meaning of the measured parameters,' for the range of conditions under which they are acquired. Our approach involves: (1) identifying the assumptions made in deriving parameters from the measured radiances, (2) testing the input data and derived parameters for statistical error, sensitivity, and internal consistency, and (3) comparing with similar parameters obtained from other sources using other techniques. A study of this type was performed for sea surface temperature (Njoku, 1985), and our project is one of several parallel efforts currently underway to validate different cloud climatologies (e.g., Rossow et al., 1985; 1990). The validation effort we are undertaking introduces a number of problems that may be of interest to specialists in computational statistics, such as the INTERFACE community, as well as to those involved in research directly related to interpreting large geophysical data sets. This article summarizes the key data handling issues we have encountered.

2. The Need for 'Level 2' Data

Large geophysical data sets, such as cloud climatologies, are often distributed to researchers in gridded (Level 3) form. This can reduce the data volume by orders of magnitude relative to the parameter values for each individual sounding (Level 2), and provides the user with a 'spatially uniform' data product. For example, Figure 1A is the global, monthly-mean cloud amount map for July 1979 from the HIRS2/MSU data, in the original 2 degree by 2.5 degree averaging bins. All accepted cloud amount data from the individual atmospheric soundings that fell within each geographic box were summed, and mean and variance values for each box were calculated.

Several problems occur when using Level 3 products for validation. First, if only the Level 3 parameter values and associated variances are available, there is no way to assess how much of the reported variance is due to inherent non-uniformity of the parameter over the averaging region. Essentially, the instrument resolution is degraded to a scale comparable to the box size, and information originally acquired to measure smaller-scale phenomena in both the spatial and temporal domains is lost. For example, in a 2 by 2.5 degree box, the surface temperature may exhibit random fluctuations of half a degree and may change systematically by several degrees, whereas the box average variance will assign all the variability to random error.

We encountered a second problem when making comparisons among Level 3 products with different gridding schemes. The best concurrent cloud climatology available for comparison with the data in Figure 1A was derived from the Temperature Humidity Infrared Radiometer/Total Ozone Mapping Spectrometer (THIR/TOMS) on the NASA Nimbus 7 satellite (Stowe et al., 1988; 1989). The standard THIR/TOMS Level 3 data product was binned according to a global 500 by 500 km grid that is also used for Earth radiation budget studies. The July 1979 HIRS2/MSU Level 3 data, degraded using area-weighted averaging to the THIR/TOMS spatial grid, is shown in Figure 1B. We then resampled the degraded HIRS2/MSU data back to the 2 by 2.5 degree grid, and subtracted it from the original HIRS2/MSU data (Figure 1C). Note that the differences are nearly as large as the range of the signal, with both positive and negative values. The pattern of differences varies with the location of edges in the original data, and is modulated by the relative position of grid boundaries. Differences are especially large at high latitudes, where the spatial resolution of the THIR/TOMS grid is much lower than that of the HIRS2/MSU grid, and wherever there are sharp edges generated by cloud patterns, such as in the intertropical convergence zone and monsoon areas.

With the Level 2 products, we have access to physical quantities at the full resolution acquired by the instruments, and avoid introducing additional artifacts into the comparison between data sets. Level 2 data are not uniformly distributed over the surface. At low latitudes there are gores in the HIRS2 sampling between orbits, whereas at high latitudes, the surface is heavily oversampled. Data dropouts and calibration lines occur at all latitudes. The sample resolution changes by more than a factor of 2 from nadir to the limits of each scan. As a first step toward making comparisons among Level 2 data sets, surfaces that take account of non-uniform clustering of data points may be fit to the data. We have begun experimenting with locally adaptive surface fitting techniques (e.g., Renka, 1988), and are exploring the use of methods that generate variance surfaces together with each fitted surface (Cresse, 1989, and references therein).

Binning, which is traditionally used to make comparisons among global data sets, is performed as an automatic procedure. In using Level 2 data for validating data sets, geographic sub-regions of the globe must be selected for surface fitting, based upon some criterion that evaluates the density of points relative to the size of local gradients of the parameter field, possibly in several directions. Figure 2 illustrates the role of interactive geographic subset selection a part of the software we are assembling to perform the HIRS2/MSU validation. 'HDF' in this figure refers to Hierarchical Data Format, a transportable file format that eliminates all but an initial file conversion for exchanging data among DEC, Sun, MacIntosh, and other machines used in the validation (NCSA Software Tools Group, 1990).

This allows us to store single copies of data files on centrally located disks, that are accessible across the network to machines with differing architectures. We are currently investigating the criteria for accepting subsets, choice of method for surface fitting, and methods for making formal comparisons among surfaces fitted to data from different sources. The important question of interpolation in the temporal domain we set aside for the present.

To summarize: in spite of the much larger volume of the Level 2 data, relative to Level 3, and the collection of issues related to the spatial and temporal sampling of Level 2 data, we need the ability to access, store, and process Level 2 data for (1) studies of the internal consistency and precision of the data set and (2) comparisons with other cloud climatologies, that are involved in the validation of the HIRS2/MSU cloud parameters. We anticipate that similar needs will arise for interdisciplinary process studies, and in work directed toward using observations to better understand mesoscale climatological phenomena.

3. Tracking Assumptions in the Code

Another issue that bears upon the degree to which we may perform validation, and other scientific analysis on large data sets, is our ability to grasp the collection of constraints imposed on parameter values by the code that generates them. An assumption embedded in a large data handling code may produce results that hide important information in the data, or may produce patterns in the data that could be incorrectly interpreted as scientifically meaningful.

We are experimenting with methods of charting the collection of assumptions, as a way of calling the attention of the user to areas where the code may influence the output parameters. We are using standard charting symbols as much as possible (e.g., Yourdon and Constantine, 1979). An example of this type of chart is Figure 3. This shows the flow of control and the flow of assumptions made in a relatively small part of the HIRS2/MSU analysis code that produces Level 3 data from Level 2 products. This chart made clear the number and complexity of the assumptions involved in generating Level 3 products, and it played a role in our assessment of the value of Level 3 data for the validation exercise.

Charting the flow of control provides a needed context for the constraints placed on the data. These charts take a step in the direction of making it *possible* to keep track of assumptions, but they do not eliminate the work involved in carefully assessing the meaning of derived parameters.

4. Conclusions

The HIRS2/MSU cloud parameter validation effort raises a number of data handling issues that are likely to arise frequently when scientific analysis is attempted on large geophysical data sets. We need Level 2 data (individual measurements in geophysical units) (A) to perform comparisons among data sets with different sampling, and (B) to understand the effects of spatial and temporal sampling on the 'average' values obtained from a single data set. The need for Level 2 data severely complicates data handling. Among the areas where advances would be most helpful are:

1. Surface fitting software for data distributed non-uniformly in 2-dimensional space, and ways to obtain some measure of the associated variances.

2. Software for making formal comparisons among fitted surfaces from several sources, and their associated variance surfaces.

3. Ways of documenting software and data files so they may be exchanged and used by others easily.

4. Ways of documenting the assumptions embedded in retrieval and processing algorithms, so a researcher studying the data products can grasp the collection of constraints placed on the output data by the code.

5. Additional ways of storing data. For a given Level 2 data product, we need readily accessible data storage capacity of between one and two *orders of magnitude* the size of the basic data set, for intermediate and derived products that are created as part of the validation.

Several longer-term needs include:

6. The development of validation procedures that are easy enough to apply so that it will be feasible to generate and access a large number of validated geophysical data sets for interdisciplinary studies of all types.

7. Ways of fitting surfaces to data values distributed non-uniformly in 2-dimensional space and in time, and obtaining a measure of the associated variances.

8. Better ways of discovering patterns and surprises in high-dimensional data sets.

9. Ways of fitting hyper-surfaces to higher dimensional data sets, and techniques for studying them.

We have described our data, the collection of problems we are facing in the validation work, and our approaches to some of these issues. Solutions or partial solutions may exist to some of the problems that are not widely known outside specialized data handling and computational statistics communities. We hope to stimulate experts in these fields to participate in the effort to improve our understanding of Earth through the study of large, geophysical data sets.

Acknowledgments

We thank Paul Tukey for inviting us to participate in the INTERFACE 91 conference, and Daniel Carr, Jeff Dozier, Mike Freilich, Wes Nicholson, Bill Rossow, Victor Zlotnicki, and Richard Zurek for stimulating discussions on many aspects of this work. This project is supported in part by the NASA Earth Sciences Interdisciplinary Program in the Earth Science and Applications Division, and by the Jet Propulsion Laboratory Director's Discretionary Fund. The work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

Cressie, N. (1989), Geostatistics, *The Amer. Statistician*, **43**, 197-202.

Earth System Science Committee (1988), "Earth System Science: A Closer View", Report of the Earth System Science Committee, NASA Advisory Council, NASA, Washington, D.C.

EOS Data Panel (1986), The Earth observing system: Report of the EOS data panel, Vol 2a, NASA Tech. Memo. 8777. Washington, D.C.

NCSA Software Tools Group (1990), Hierarchical Data Format, National Center for Supercomputing Applications, Champaign, IL.

Njoku, E.(1985), Satellite-derived sea surface temperature: Workshop comparisons, *Bull. Am. Meteorol. Soc.*, **66**, 274-281.

Renka, R.J. (1988), Multivariate interpolation of large sets of scattered data, *ACM Transact. Math. Software*, **14**, 139-148.

Rossow, W.B., Mosher, F., Kinsella, E., Arking, A., Desbois, E., Harrison, E., Minnis, P., Ruprecht, E., Seze, G.,Simmer, C., and Smith, E. (1985), ISCCP cloud algorithm intercomparison., *J. Climate Appl. Meteor.*, **24**, 877-903.

Rossow, W.B. (1990), Report of the Workshop on Comparison of Cloud Climatology Datasets, NASA Goddard Institute for Space Studies, New York.

Space Science Board (1982), Data management and computation, Vol 1: Issues and recommendations. National Academy of Sciences/ National Academy Press, Washington, D.C.

Stowe, L.L., Wellemeyer, C.G., Eck, T.F., Yeh, H.Y.M., and the NIMBUS 7 Cloud Data Processing Team (1988), NIMBUS 7 global cloud climatology. Part I: Algorithms and validation, *J. Climate*, **1**, 445-470.

Stowe, L.L., Yeh, H.Y.M., Eck, T.F., Wellemeyer, C.G., H.L. Kyle, and the NIMBUS 7 Cloud Data Processing Team (1979), NIMBUS 7 global cloud climatology. Part II: First year results, *J. Climate*, **2**, 671-709.

Susskind, J., Rosenfield, J., Reuter, D., Chahine, M.T. (1984), Remote sensing of weather and climate parameters from HIRS2/MSU on TIROS-N, *J. Geophys. Res.*, **89**, 4677-4697.

Susskind, J., Reuter, D., Chahine, M.T. (1987), Cloud fields retrieved from analysis of HIRS2/MSU sounding data, *J. Geophys. Res.*, **92**, 4035-4050.

Yourdon, E., and Constantine, E.E. (1979), Structured Design: Fundamentals of a Discipline of Computer Program and System Design, Yourdon Press, NJ, pp 473.

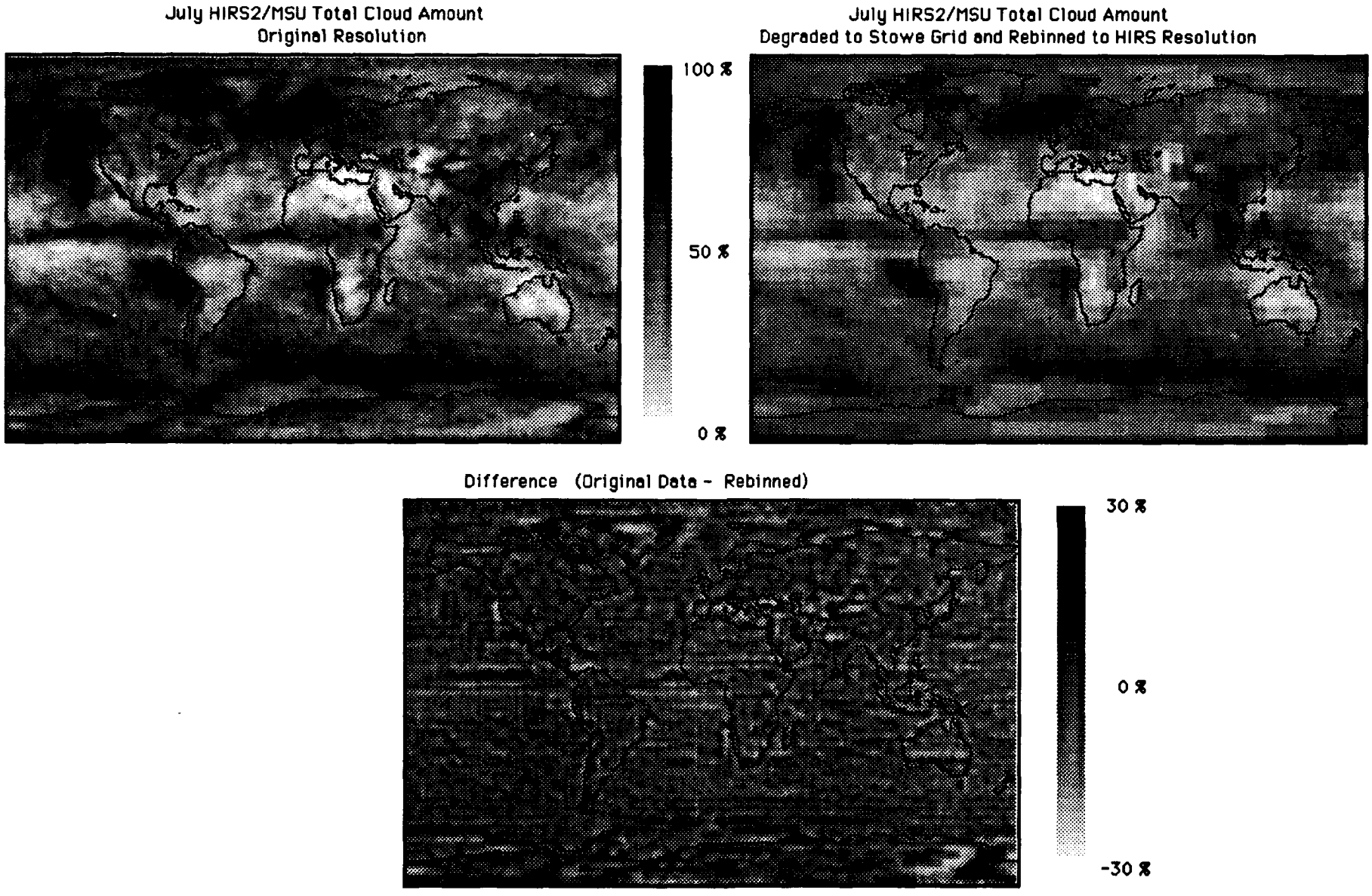


Figure 1. The Effect of Rebinning on Global Cloud Amount

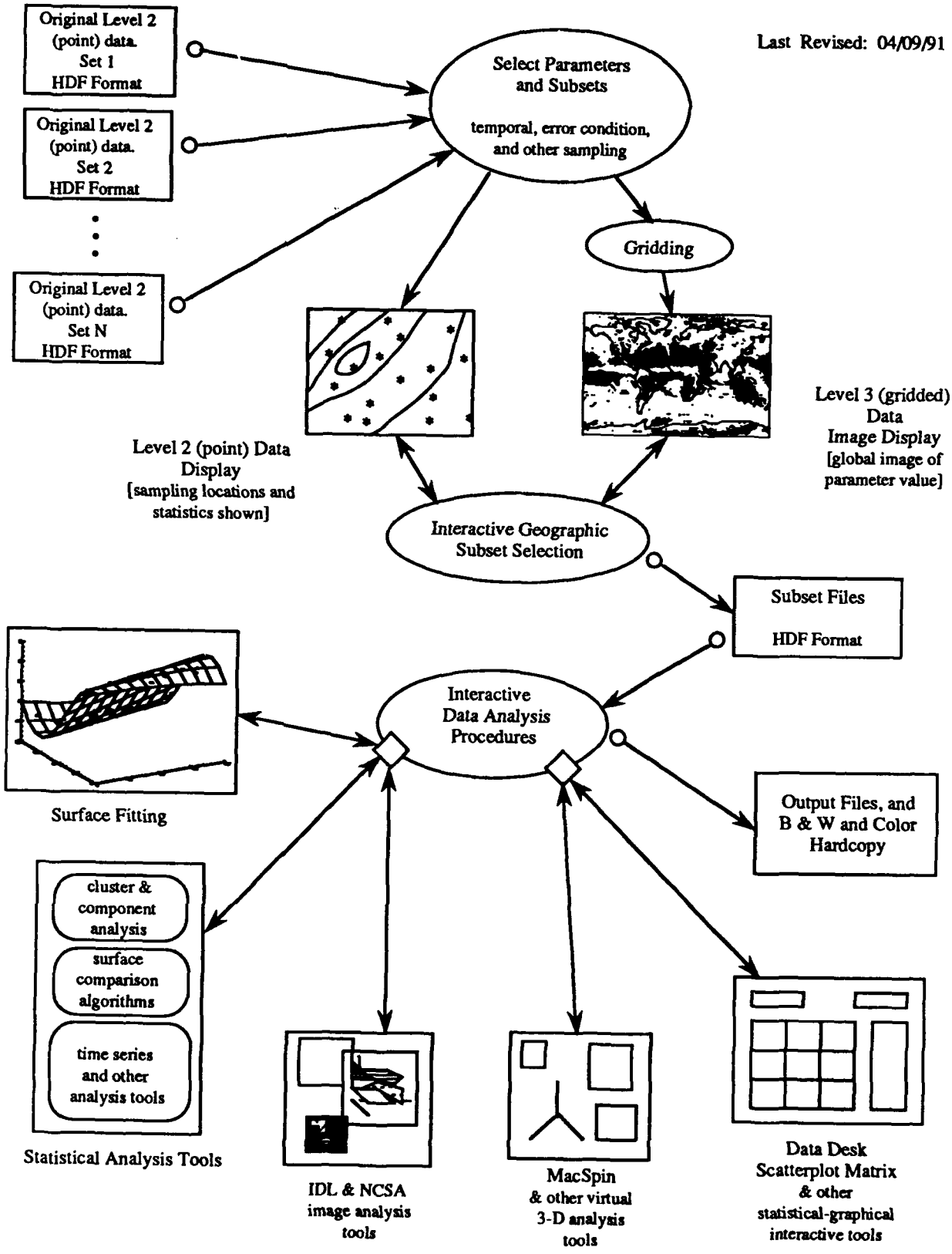


Figure 2. Level 2 Data Analysis Software

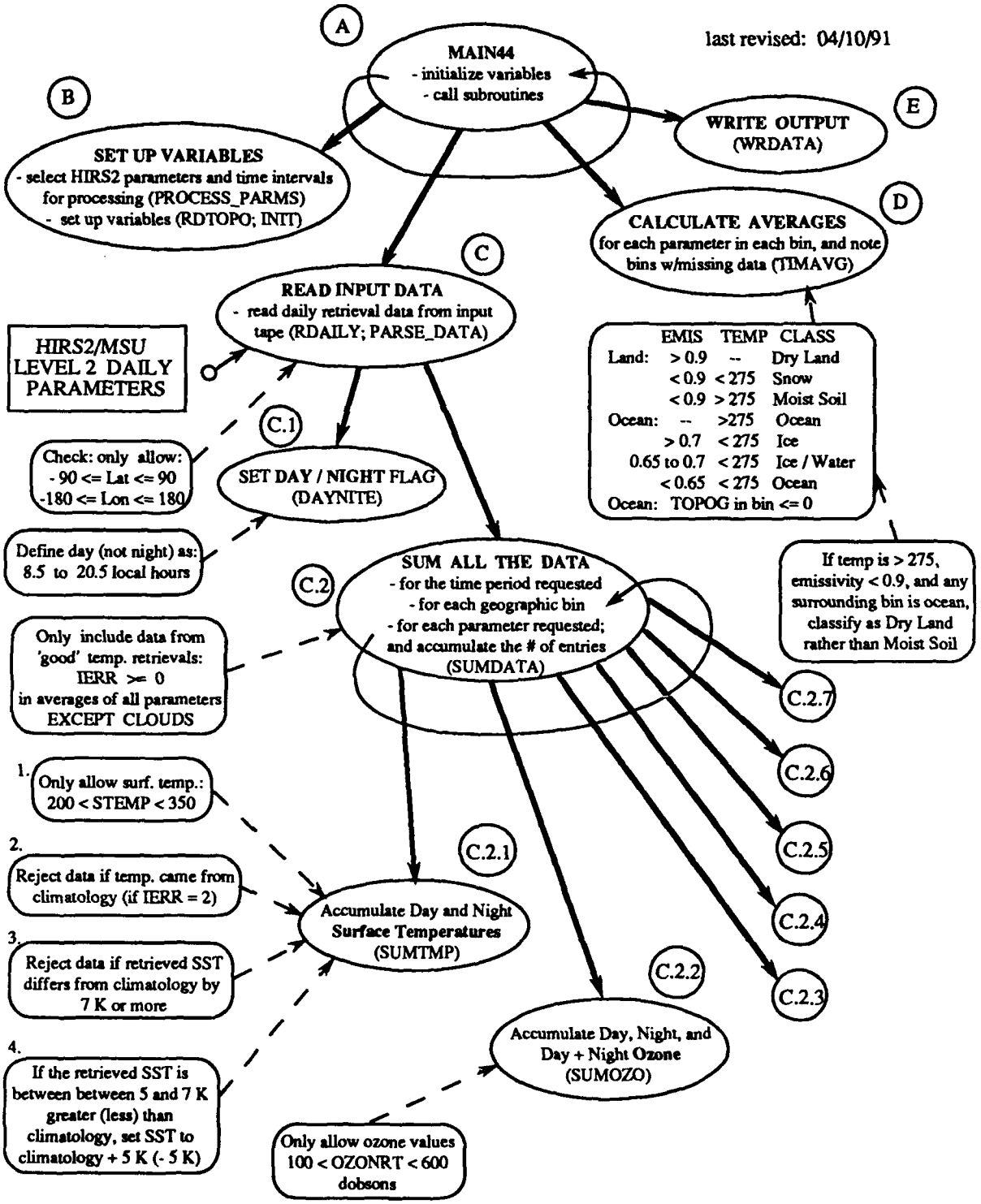


Figure 3. HIRS2 Level 2 to 3 Software Overview / Assumptions

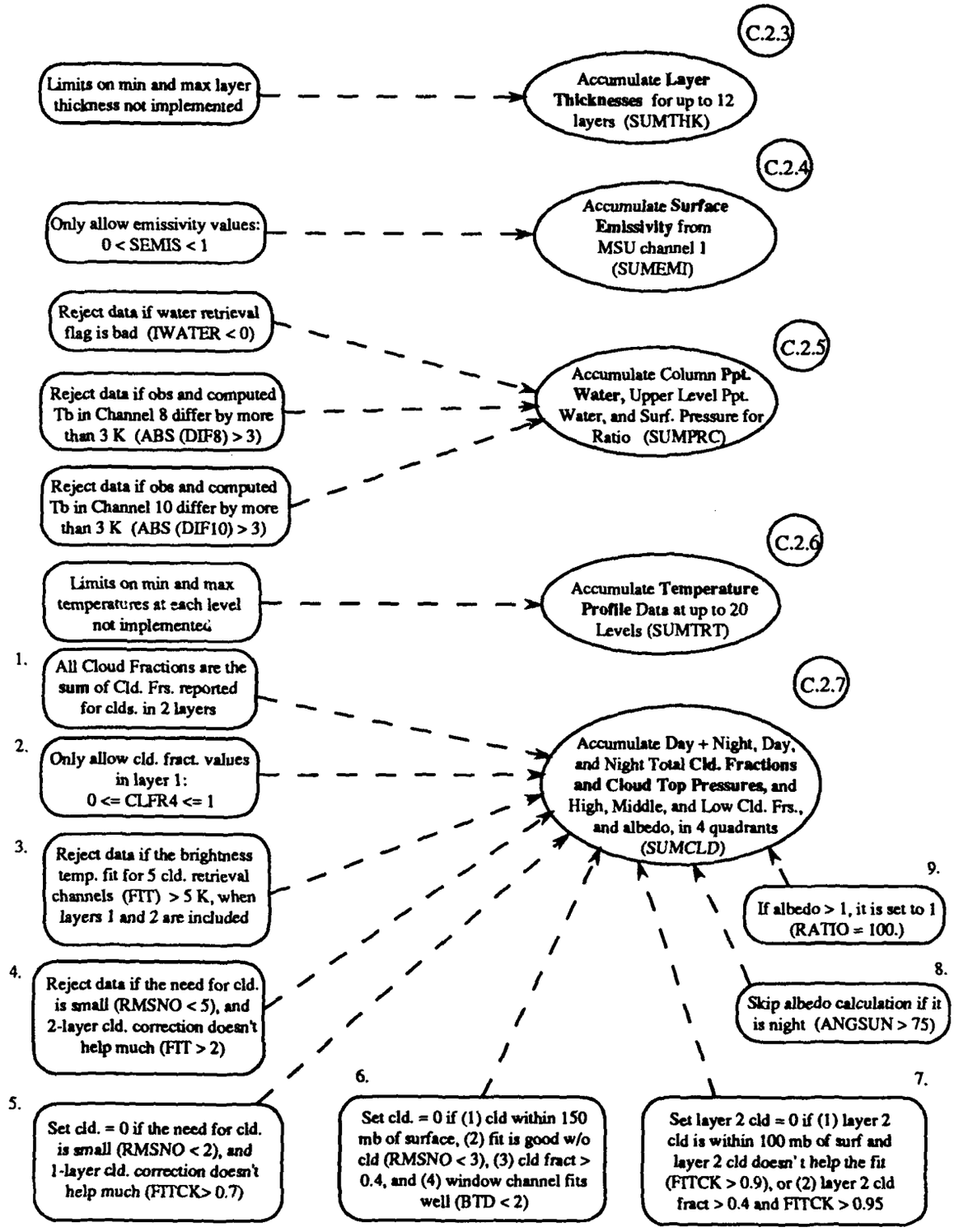


Figure 3. HIRS2 Level 2 to 3 Software Overview (Continued)

From Observed Likelihood to Tail Probabilities: An Application to Engineering Statistics

Augustine C. M. Wong
Department of Statistics and Actuarial Science, University of Waterloo
Waterloo, Ontario, Canada. N2L 3G1

Abstract

Inference for a canonical parameter in the presence of nuisance parameters usually requires high dimensional integrals to obtain the marginal or conditional tail probabilities. A simple and very accurate method is proposed to obtain any arbitrary level of significance for the parameter of interest. This method only requires a fine tabulation of the canonical parameter and the corresponding observed likelihood function, which can be either the full, marginal or conditional observed likelihood function, as input, and produces the left tail probabilities at the observed data value as output. Applications of this method to some widely used engineering statistical models will be discussed.

1. Introduction

A very accurate approximation to the density of the mean of a sample of independent and identically distributed observations was introduced to statistics by Daniels (1954). This approximation is generally referred to as the saddlepoint approximation. It focuses on an approximate conversion of a cumulant generating function to a corresponding density function. However, it was not until the appearance of the discussion paper by Barndorff-Nielsen & Cox (1979) that the importance and usefulness of this method became well known. Since then, many statistical applications of the saddlepoint approximation have been developed.

In many applications, it will be of interest to compute approximate tail probabilities or cumulative distribution functions, rather than densities. A very accurate tail probability approximation for the sample mean derived by the saddlepoint method was obtained by Lugannani & Rice (1980) and further discussed in Daniels (1987). Let $(x_1, \ldots, x_n)$ be a sample of observations, each with cumulant generating function $c(\varphi)$. Then the Lugannani & Rice formula, which approximates the distribution function for the sample mean, $\bar{x}$, takes the form

$$F(\bar{x}) \approx \Phi(z) + \phi(z)\{\frac{1}{z} - \frac{1}{\zeta}\} \qquad (1)$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the density and distribution functions of a standard normal distribution,

$$z = \mathrm{sgn}(\zeta)\{2n[\hat{\varphi}\bar{x} - c(\hat{\varphi})]\}^{1/2} \qquad (2)$$

$$\zeta = \hat{\varphi}\{nc''(\hat{\varphi})\}^{1/2} \qquad (3)$$

with $\hat{\varphi}$ satisfies $c'(\hat{\varphi}) = \bar{x}$, and $c'(\varphi)$ and $c''(\varphi)$ denote the first and second derivatives of $c(\varphi)$. If the sample mean is equal to the true mean, the Lugannani & Rice formula is undefined; however, Daniels (1987) provides a formula to handle this situation. Since our main concern is the tail probabilities, therefore we will not discuss Daniels' formulation in this paper.

A detailed review of the saddlepoint methods in statistics is given by Reid (1988).

In Section 2, a numerical program that uses the observed likelihood function as input and outputs the significance function for a real parameter of interest is developed. Some reliability models are used to illustrate the accuracy of the procedure.

Section 3 examines how the preceding numerical procedure can be applied to models with a scalar parameter of interest and the pressence of nuisance parameters. Some concluding remarks are recorded in Section 4.

2. Converting Observed Likelihood Function to Significance Function

Let us denote the observed log likelihood function of the model $f(x;\theta)$ at an observed data point, x^{obs}, up to an additive constant , be

$$l(\theta) = l(\theta; x^{obs}) = \log(f(x^{obs};\theta)).$$

Also, denote the significance function as

$$p(\theta) = P(X \leq x^{obs};\theta),$$

which is the probability to the left of the data point, x^{obs}. A $(1-\alpha)\times 100\%$ confidence interval for θ can be obtained from the significance function by $(p^{-1}(1-\alpha/2), p^{-1}(\alpha/2))$.

The aim of this section is to illustrate how to convert an observed likelihood function, or equivalently an observed log likelihood function, to a significance function.

2.1. Exponential model

For an exponential model

$$f(x;\theta) = \exp\{t\theta - c(\theta) + h(x)\}$$

with canonical parameter θ and minimal sufficient statistic $t = t(x)$, the observed log likelihood function at the data value, x^{obs} is

$$l(\theta) = t^{obs}\theta - c(\theta)$$

where $t^{obs} = t(x^{obs})$. Then the Lugannani & Rice formula gives the significance function

$$\begin{aligned} p(\theta) &= P(X \leq x^{obs};\theta) \\ &= P(T \leq t^{obs};\theta) = P(\hat{\Theta} \leq \hat{\theta}^{obs};\theta) \\ &= \Phi(r) + \phi(r)\{\frac{1}{r} - \frac{1}{q}\} + O(n^{-3/2}) \quad (4) \end{aligned}$$

where

$$\begin{aligned} r &= \mathrm{sgn}(q)\{2[l(\hat{\theta}^{obs}) - l(\theta)]\}^{1/2} \quad (5) \\ q &= (\hat{\theta}^{obs} - \theta)\{j(\hat{\theta}^{obs})\}^{1/2} \quad (6) \\ j(\hat{\theta}^{obs}) &= -\left.\frac{\partial^2 l(\theta)}{\partial\theta^2}\right|_{\hat{\theta}^{obs}}. \end{aligned}$$

The accuracy of $O(n^{-3/2})$ of (4) in nuisance parameter case is discussed in Barndorff-Nielsen & Cox (1989), Daniels (1987), and Fraser & Reid (1990).

Fraser, Reid & Wong (1991) showed that with a numerical tabulation of $l(\theta)$ over a equally and finely spaced grid of θ in steps of $\pm\delta$, and successive divided differences

$$\begin{aligned} l_1(\theta) &= \{l(\theta+\delta) - l(\theta)\}/\delta \quad (7) \\ l_2(\theta) &= \{l_1(\theta) - l_1(\theta-\delta)\}/\delta, \quad (8) \end{aligned}$$

(6) can be approximated by

$$q \approx (\hat{\theta}^{obs} - \theta)\{-l_2(\hat{\theta}^{obs}\}^{1/2}. \quad (9)$$

Thus the significance function can be obtained form converting the observed likelihood function by using (4) with (5) and (9).

Example 1: The gamma distribution has wide application in environmetrics and reliability. We construct a simple example where the scale parameter is 1 and the shape parameter is the parameter of interest. Consider a sample of size 1, the density is

$$f(x;\theta) = \Gamma^{-1}(\theta)\mathrm{e}^{-x}x^{\theta-1}$$

on $(0,\infty)$. Consider the data value $x^{obs} = 10$. The observed log likelihood function is

$$l(\theta) = \theta\log(10) - \log(\Gamma(\theta)).$$

By using (4) with (5) and (9), the significance function is obtained.

Moreover, we can also obtain the significance function from some standard approximations namely the maximum likelihood estimate,

$$(\hat{\theta}^{obs} - \theta)\{j(\hat{\theta}^{obs})\}^{1/2} \dot{\sim} N(0,1),$$

the score statistic,

$$S(\theta)\{j(\hat{\theta}^{obs})\}^{-1/2} \dot{\sim} N(0,1),$$

and the signed square root of the likelihood ratio statistic,

$$\text{sgn}(\hat{\theta}^{obs} - \theta)\{2[l(\hat{\theta}^{obs}) - l(\theta)]\}^{1/2} \dot{\sim} N(0,1).$$

Figure 1 plotted the significance functions obtained by the 4 approximations and the exact significance function obtained by exact integration. It is not surprising that the proposed method is more accurate than the 3 standard approximations because it is a third order asymptotic method, whereas the others are only first order methods. Furthermore, the first order methods depend heavily on the normality assumption, which clearly does not hold here because of the fixed left boundary.

2.2. Location model

Consider the simple location model

$$f(x;\theta) = f(x - \theta).$$

The observed log likelihood function at x^{obs} is

$$l(\theta) = \log(f(x^{obs};\theta)) = l(x^{obs} - \theta).$$

Fraser (1988) and DiCiccio, Field & Fraser (1990) showed that for this model, the significance function, $p(\theta) = P(X \leq x^{obs};\theta)$, can be obtained by (4) with (5), and (6) is replaced by

$$q = S(\theta)\{j(\hat{\theta}^{obs})\}^{-1/2}.$$

Moreover, by applying (7) and (8), we have

$$q \approx l_1(\theta)\{-l_2(\hat{\theta}^{obs})\}^{-1/2}. \qquad (10)$$

Thus, $p(\theta)$ can be obtained by (4) with (5) and (10).

Example 2: Consider the location gamma model with the shape parameter is known. For this example, we choose the shape parameter to be 3. With the sample size is 1, the model has density

$$f(x;\theta) = \frac{1}{2}(x - \theta)^2 e^{-(x-\theta)}$$

with $x \geq \theta$. With the observed data $x^{obs} = 1$, the observed log likelihood function is

$$l(\theta) = 2\log(1 - \theta) + \theta.$$

By using (4) with (5) and (10), the significance function is obtained and compared with the first order methods and is shown in Figure 2. Again, the proposed method out-performed the standard approximations.

3. Conditional and Marginal Inferences

In the preceding section, we have discussed the conversion of an observed likelihood function to a significance function for scalar parameter models. Now, let us examine some multiparameter models.

Let $\theta = (\psi, \lambda)$ with a scalar parameter of interest ψ and nuisance parameter λ. Our aim is to approximate either the conditional or the marginal observed likelihood function for ψ such that the significance function, $p(\psi)$, can be obtained by the numerical procedure described in the previous section.

3.1. Exponential model

Consider an exponential model with canonical parameter $\theta = (\psi, \lambda)$ where ψ, a scalar parameter, is our parameter of interest. The density has the form

$$f(x;\theta) = \exp\{\psi t_1 + \lambda' t_2 - c(\psi, \lambda) + h(x)\}$$

where $(t_1, t_2) = (t_1(x), t_2(x))$ is the minimal sufficient statistic.

The conditional distribution of t_1 given t_2 is free of the nuisance parameter and would typically be used for inference about ψ in the absence of knowledge of λ. An $O(n^{-3/2})$ approximation to the observed likelihood from this conditional distribution is given in Cox & Reid (1987) and Fraser & Reid (1990); and the approximated observed conditional log likelihood function takes the form

$$l(\psi) \approx l(\psi, \hat{\lambda}_\psi) + \frac{1}{2} \log \left| j_{\lambda\lambda}(\psi, \hat{\lambda}_\psi) \right| \qquad (11)$$

where $\hat{\lambda}_\psi$ is the maximum likelihood estimate of λ for a fixed ψ, and $j_{\lambda\lambda}(\psi, \hat{\lambda}_\psi)$ is the observed information concerning λ for a fixed ψ. By tabulating ψ and $l(\psi)$, the significance function, $p(\psi) = P(T_1 \leq t_1^{obs} \mid t_2^{obs}; \psi)$, can be obtained by (4) with (5) and (9). Fraser & Reid (1990) showed that this conversion is $O(n^{-3/2})$ based on the conditional distribution of t_1 given t_2.

Example 3: Consider the Proschan (1963) data which recorded the times between successive failures of air conditional equipment in 13 Boeing 720 aircrafts. For aircraft number 7909, the data is recorded in Keating, Glaser & Ketchum (1990). The model being considered is the two parameter gamma model with density

$$\Gamma^{-1}(n\psi)\lambda^{-n\psi} \exp\{-t_2/\lambda + n\psi t_2\} \times$$
$$\Gamma(n\psi)\Gamma^{-n}(\psi) \exp\{-n\psi \log(t_2) + \psi t_1\},$$

where $n = 29, t_1^{obs} = 118.8084$, and $t_2^{obs} = 2422$. From (11), the conditional observed likelihood function is obtained. Keating, Glaser & Ketchum (1990) produces various tables to obtain the observed level of significance. In particular, they tested if the gamma distribution has an increasing failure rate ($H_0 : \psi = 1$ versus $H_1 : \psi > 1$) and they reported the observed level of significance associated to the test is 3.84%.

Wong (1991) applied the proposed procedure and obtained the observed level of significance as 3.85%. The advantage of the proposed procedure is its efficiency and simplicity, and the ability to obtain arbitrary level of significance from the significance function.

3.2. Transformation model

Consider a general location model

$$f(x; \theta) = f(t_1 - \psi, t_2 - \lambda).$$

The marginal density of t_1 is free of λ and would typically be used for inference concerning ψ in the absence of knowledge of λ. Fraser & Reid (1990) showed that for this model,

$$l(\psi) \approx l(\psi, \hat{\lambda}_\psi) - \frac{1}{2} \log \left| j_{\lambda\lambda}(\psi, \hat{\lambda}_\psi) \right| \qquad (12)$$

is an $O(n^{-3/2})$ approximation of the observed marginal log likelihood function based on the marginal density of t_1. Again by tabulating ψ and $l(\psi)$, the significance function, $p(\psi) = P(T_1 \leq t_1^{obs}; \psi)$, can be obtained by using (4) with (5) and (10). This conversion from observed likelihood to significance function is also shown in Fraser & Reid (1990) to be an $O(n^{-3/2})$ approximation.

We can now consider the location-scale model,

$$f(x; \theta) = \frac{1}{\sigma} f\left(\frac{x - \mu}{\sigma}\right)$$

where μ is the location parameter and σ is the scale parameter. The sampling density can be written as

$$\prod f\left(\frac{x_i - \mu}{s} e^{\log(s) - \log(\sigma)}\right) e^{-\log(\sigma)}$$

where s^2 is the sample variance, and μ and $\gamma = \log(\sigma)$ are location parameters. Hence the joint observed log likelihood function can be written as

$$l(\mu, \gamma) = -n\gamma + \sum \log(f((x_i^{obs} - \mu)e^{-\gamma})).$$

By (12), we can tabulate the corresponding approximated marginal likelihood function and thus the significance function can be obtained.

Example 4: Let $(x_1, \ldots, x_n)$ are n observations sampled from a Weibull population with density

$$f(x) = \left(\frac{\beta}{\alpha}\right)\left(\frac{x}{\alpha}\right)^{\beta-1} \exp\left\{-\left(\frac{x}{\alpha}\right)^{\beta}\right\}.$$

Let $\mu = \log(\alpha)$, $\sigma = 1/\beta$ and $y_i = \log(x_i)$. Then $y_i = \mu + \sigma z_i$ where z_i has the extreme value distribution with density

$$f(z) = \exp\{z - e^z\}.$$

In other words, the Weibull distribution can be obtained from a location-scale transformation of the extreme value distribution. Thus the joint observed log likelihood function is

$$\begin{aligned} l(\mu, \gamma) = & - n\gamma + n(\bar{y}^{obs} - \mu)e^{-\gamma} \\ & - \sum \exp\{(y_i^{obs} - \mu)e^{-\gamma}\}. \end{aligned}$$

From (12), we can tabulate the marginal observed likelihood function for μ and γ separately.

The above model is applied to the Zieblien & Zelen (1956) data, recorded in Fraser (1979, page 33). Tables 1 and 2 compared the 90%, 95% and 99% confidence intervals for μ and σ obtained by exact integration, which is recorded in Fraser (1979) and the proposed method. Again, it shows that the numerical procedure is very accurate.

4. Conclusion

In this paper, we required the parameter of interest be a scalar canonical parameter. However, if it is not the case, Fraser & Reid (1990) derived a method to extract the canonical parameter from the observed likelihood function. Moreover, we can extend the method described in Section 3 to the non-normal regression model. Finally a generic computer program has been developed for this numerical procedure. It requires a finely and equally spaced tabulation of the canonical parameter and its observed likelihood function as input, and produces the corresponding significance function as output. The program is available from the author upon request.

References

[1] Barndorff-Nielsen, O. E. and Cox, D. R. (1979). Edgeworth and saddlepoint approximations with statistical applications (with discussion). *J. Royal Statist. Soc. B* **41**, 279-312.

[2] Barndorff-Nielsen, O. E. and Cox, D. R. (1989). *Asymptotic Techniques for Use in Statistics.* London: Chapman and Hall.

[3] Cox, D. R. and Reid, N. (1987). Parameter orthogonality and approximate conditional inference (with discussion). *J. Royal Statist. Soc. B* **49**, 1-39.

[4] Daniels, H. E. (1954). Saddlepoint approximations in statistics. *Ann. Math. Statist.* **25**, 631-650.

[5] Daniels, H. E. (1987). Tail probability approximation. *Int. Statist. Rev.* **55**, 37-48.

[6] DiCiccio, T. J., Field, C. A. and Fraser, D. A. S. (1990). Approximation of marginal tail probabilities and inference for scalar parameters. *Biometrika* **77**, 77-96.

[7] Fraser, D. A. S. (1979). *Inference and Linear Models.* New York: McGraw Hill.

[8] Fraser, D. A. S. (1988). Normed likelihood as saddlepoint approximation. *J. Mult. Anal.* **27**, 181-193.

[9] Fraser, D. A. S. and Reid, N. (1990). From multiparameter likelihood to tail probability for a scalar parameter. Technical Report #9003, University of Toronto.

[10] Fraser, D. A. S., Reid, N. and Wong, A. (1991). From observed likelihood to tail area: a two pass approximation. *J. Royal Statist. Soc. B* **52**, to appear.

[11] Keating, J. P., Glaser, R. E. and Ketchum, N. S. (1990). Testing hypotheses about the shape parameter of a gamma distribution. *Technometric* **32**, 67-82.

[12] Lugannani, R. and Rice, S. (1980). Saddlepoint approximation for distribution of the sum of independent random variables. *Adv. Applied Prob.* **12**, 475-490.

[13] Reid, N. (1988). Saddlepoint methods and statistical inference. *Statist. Sci.* **3**, 213-238.

[14] Wong, A. (1990). Converting observed likelihood functions to tail probabilities for exponential linear models. Ph.D. thesis, University of Toronto.

[15] Wong, A. (1991). Inferences on the shape parameter of a gamma distribution: a conditional approach. Submitted for publication.

Figure 1

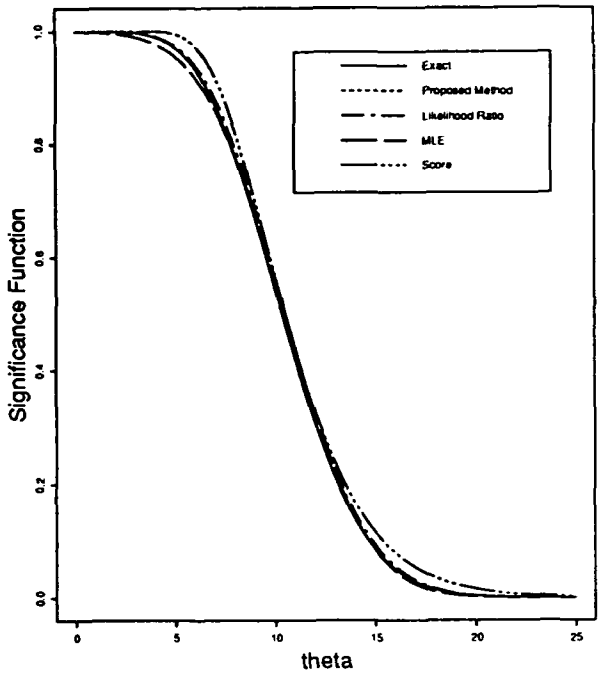


Figure 2

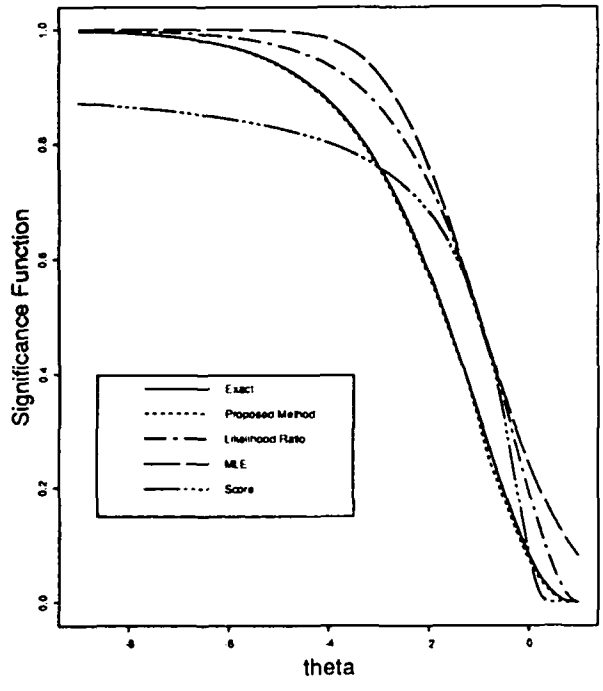


Table 1: Confidence intervals for μ
(in units of y)

	Exact	Approximation
90%	(4.221, 4.590)	(4.2205, 4.5891)
95%	(4.182, 4.627)	(4.1811, 4.6264)
90%	(4.099, 4.701)	(4.0989, 4.7041)

Table 2: Confidence intervals for σ
(in units of y)

	Exact	Approximation
90%	(0.386, 0.659)	(0.3853, 0.6589)
95%	(0.369, 0.700)	(0.3689, 0.6999)
90%	(0.340, 0.792)	(0.3398, 0.7918)

A Comparison of Approaches to Inference for Nonlinear Models

Christian Ritter[†], **Søren Bisgaard**[‡] and **Douglas Bates**[†]
Center for Quality and Productivity Improvement
[†]Department of Statistics
[‡]Department of Industrial Engineering
University of Wisconsin – Madison

Abstract

As greater computing power becomes routinely available to researchers, analyses based on Bayesian or likelihood methods become easier to perform, especially since the increase in computing power has been accompanied by development of inventive statistical algorithms for inference. We consider here the nonlinear regression model but these approaches to inference are applicable in more general circumstances and we feel the comparisons will remain useful. Several methods can be used for inference in nonlinear regression: propagation of errors, likelihood profiles, approximate marginal likelihoods and posteriors, and Monte Carlo methods such as importance sampling and the Gibbs sampler. These methods vary in computing intensity and in their ability to handle poorly conditioned situations. Furthermore, since some of these methods have only been recently developed, it is not easy for the practitioner to compare them and choose between them because they are not widely implemented. We demonstrate the respective merits of these methods in a small but instructive example.

Keywords: Nonlinear Models; Profile Likelihood; Importance Sampling; Gibbs Sampler, Approximate Marginalization

1. Our Motivating Example

Electron Spectroscopy for Chemical Analysis (ESCA) is a key technique at the Engineering Research Center for Plasma Aided Manufacturing, University of Wisconsin, to study the chemical bonding structure of polymer surfaces. In our case, the same material, a deposited polymer, will be examined several times over a period of weeks, and the experimenters want to know how the bond structure changes. A plot of the data from a spectroscopic analysis of one sample, along with fitted components and residuals, is shown in Figure 1.

The immediate objective of the analyst is to resolve these data into a known number of peaks, each of the form

$$\alpha_j \left\{ \rho_j \exp\left[-\ln 2 \cdot \left(\frac{2(x_i - \beta_j)}{\gamma_i} \right)^2 \right] \right.$$
$$\left. +(1-\rho_j)\left[1 + \left(\frac{2(x_i - \beta_j)}{\gamma_j} \right)^2 \right]^{-1} \right\}$$

Here the parameter β_j is the center (location) of peak j, γ_j is the bandwidth at half the peak height, ρ_j is the proportion of peak j in the form of a Gaussian curve (hence $1 - \rho_j$ is the

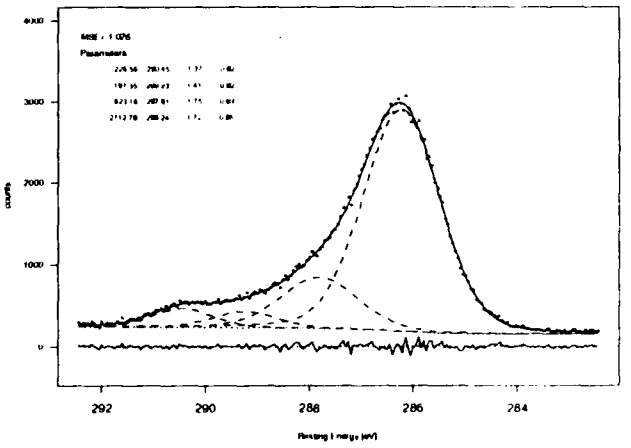


Figure 1: Observed electron intensities (*) versus binding energy for the Carbon 1S peak in plasma polymerized methyl methacrylate (PPMMA). Also shown are the fitted spectrum (solid line), its components (dashed lines), the baseline (dot-dashed line), parameter estimates using weighted nonlinear least squares, and the residuals.

[†]This research supported by the National Science Foundation under grants DMS-9005904 and ECD-8721545

proportion of the peak j in the form of a Cauchy curve), and α_j is the peak height.

Fitting four such peaks to these data using weighted least squares produces a fit as in Figure 1. Weights were used to accommodate systematic differences in the variance of the response. Moreover, for this particular fit a strong prior was enforced on the spacings between the peak locations, because these spacings are known fairly well for this polymer. Without the prior the problem would be overparameterized and the parameters would be unidentifiable.

Important characteristics of the example are that it employs a nonlinear statistical model with rather precise data and a reasonable understanding of the mechanism under study, and that the model parameters carry physical meanings.

Specifically, interest centers on the relative heights of the peaks, which are related to the relative concentrations of the corresponding chemical bonds. This requires careful inference about some of the parameters, the peak heights, and of functions of them, while the others enter as nuisance parameters. Such inference is notoriously difficult, and although we have been working on the ESCA problem for a considerable amount of time, we have not arrived at a satisfactory solution yet. However, the new methods which we shall describe in our paper seem powerful enough to handle problems of this degree of difficulty in the near future.

In Section 2 we shall introduce the new methods and in Section 3 we shall demonstrate how they perform on an example which is much simpler than the ESCA problem, but which displays some of the characteristic difficulties. In Section 4 we shall summarize our results.

2. Exploring The Objective Function

For making inferences about the parameters in a nonlinear model, we measure the "quality" of a parameter vector with an objective function such as the residual sum of squares or the likelihood or the posterior density. For point estimates, we usually quote the values of the parameters that optimize the objective function. To measure the variability of the parameters (or the variability of their estimates) jointly or individually, the most sensible and direct ways are through the objective function. Thus we want to plot contours or projections of contours of the objective function, we want to integrate the objective function over nuisance parameters and, in general, explore how the objective function depends on the parameters. We may want to do this for the original parameters or for functions of these parameters.

Several different methods can be used for exploring the objective function. The simplest method, based on a local quadratic approximation to the objective function near the optimum, is often called the "propagation of errors" method. For the nonlinear regression model, a linear approximation to the expectation function produces a quadratic approximation to the sum of squares function (Bates and Watts, 1988, chapter 2), which is used to form approximate standard errors and correlations. For likelihood and Bayesian analyses we usually approximate the log-likelihood or log-posterior density at the optimum.

Propagation of errors is very simple but often quite inaccurate. For greater accuracy, two basic approaches to exploring the objective function can be used. These are: 1) re-optimizing the objective with one of more of the parameters held fixed or 2) Monte Carlo methods designed to create a sample from a density represented by the objective function. Re-optimization is known as *profiling*. The Monte Carlo methods include importance sampling (Rubinstein, 1981) and the Gibbs sampler (Gelfand and Smith, 1990). In his discussion of this paper, Luke Tierney described the use of another Monte Carlo method, the Metropolis algorithm (Metropolis et al., 1953). Hybrid methods, where information from profiling is used to enhance the efficiency of the Monte Carlo methods, are also possible.

In profiling we chose a parameter, say θ_1, and while fixing it at a value close to but different from the estimate, say $\hat{\theta}_1 - \delta$, optimize the objective with respect to the remaining parameters. If S represents the objective, the profiled objective can be written $\tilde{S}(\theta_1)$ with the conditionally optimal values of the other parameters written $\tilde{\theta}_{-1}(\theta_1)$. This is repeated for $\hat{\theta}_1 - 2 \cdot \delta, \ldots$ and $\hat{\theta}_1 + \delta, \hat{\theta}_1 + 2 \cdot \delta, \ldots$ until $\tilde{S}$ is sufficiently different from $S(\hat{\theta})$. It produces three pieces of information: 1) the profiled value of the objective, $\tilde{S}$, 2) the conditional estimates of the other parameters, $\tilde{\theta}_{-1}(\theta_1)$, called the *profile traces*, and 3) the conditional Hessian of the objective. Piece 1) can be used by itself to define univariate empirical parameter transformations as described below. Pieces 1) and 3) are used in Laplacian integration methods to approximate marginal posterior densities (Tierney and Kadane, 1986; Tierney, Kass, and Kadane, 1988) while pieces 1) and 2) can be used to approximate projections of contours (Bates and Watts, 1988, Appendix 6).

To define the univariate empirical parameter transformations, we note that if the objective were quadratic in the θ, then $\tilde{S}$ would be quadratic in θ_1 and

$$\zeta(\theta_1) = \text{sign}(\theta_1 - \hat{\theta}_1)\sqrt{\tilde{S}(\theta_1) - S(\hat{\theta})} \qquad (2.1)$$

would be linear in θ_1. For the nonlinear regression model, dividing (2.1) by s, an estimate of standard deviation of the disturbance produces

$$\tau_i(\theta_i) = \text{sign}(\theta_i - \hat{\theta}_i)\sqrt{\tilde{S}(\theta_i) - S(\hat{\theta})}/s \qquad (2.2)$$

a nonlinear analogue of the t-statistic (Bates and Watts, 1988, chapter 6). If objective being optimized is the negative of the log-likelihood, (2.1) defines a nonlinear analogue of a z statistic. Whenever the objective is unimodal, ζ is monotone

over the range of interest and a univariate transformation. If these transformations are used on each parameter, the objective function is much closer to being quadratic. To examine the objective function in the original parameters, the transformation $\theta \mapsto \tau$ has to be inverted. Usually the τ values are sufficiently well behaved that the forward transformation can be defined by an interpolating spline but the backward transformation has to be defined with some care.

One deficiency of the profiling methods is that they give good information about the parameters chosen for the model but not about functions of the parameters. Especially for Bayesian analyses, Monte Carlo methods that generate a sample from the posterior provide a simple method of evaluating the behavior of functions of the parameters. The primary advantage of these Monte Carlo methods is that they change a problem in parametric inference into a problem in data analysis and we have good tools for data analysis in several dimensions.

3. The BOD Example

The model $y_i = \theta_1 \cdot (1 - \exp(-\theta_2 \cdot t_i)) + \epsilon_i$ is to be fitted to the data, from Table A1.4 of Bates and Watts (1988, p.270), shown in Figure 2. Note that the variability is very high and that the observation interval is too short to capture the steady state behavior with respect to time. The small sample size and an unfortunate experimental design cause pathologies of the likelihood surface, and, since there are only two parameters, these pathologies can be studied conveniently. In practice, such a problem should be approached by improving the experimental design and by taking more data; not by overanalyzing the existing observations. But inference procedures should also work in ill-conditioned cases or at least point to the causes of the ill-conditioning, so we use the BOD example as test case.

3.1. *Likelihood Contours*

The likelihood for the BOD example is of the form

$$L(\theta_1, \theta_2, \sigma^2 | \mathbf{t}, \mathbf{y}) = C \cdot \exp\left(-\frac{n}{2}\ln\sigma^2 - \frac{S(\theta_1, \theta_2)}{2 \cdot \sigma^2}\right)$$

with

$$S(\theta_1, \theta_2) = \sum_{i=1}^{n} \left[y_i - \theta_1(1 - \exp(-\theta_2 \cdot t_i))\right]^2.$$

hence contours of the sum of squares $S(\theta_1, \theta_2)$ are also likelihood contours. Using the approximation

$$\phi(\theta) = \frac{\left[S(\theta) - S(\hat{\theta})\right]/p}{S(\hat{\theta})/(n-p)} \sim \mathcal{F}_{p,n-p},$$

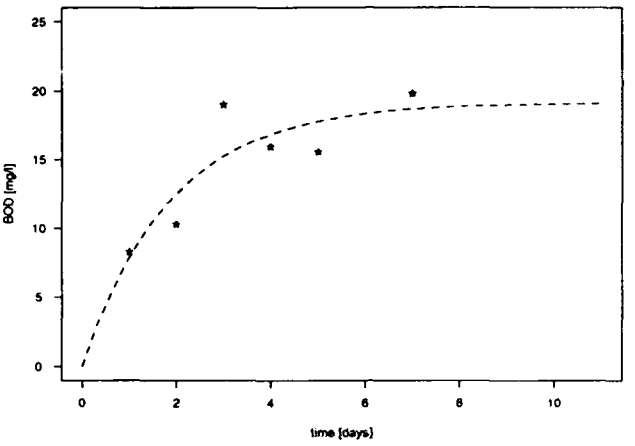


Figure 2: Biochemical Oxygen Demand (BOD) data and fitted curve

where $\hat{\theta}$ is the least squares estimate of (θ_1, θ_2), these contours can be labeled by their approximate frequency content. Such contours, created by evaluating $\phi(\theta)$ on an equispaced grid of 100 steps over $[-20, 50] \times [-2, 6]$ are shown in Figure 3.

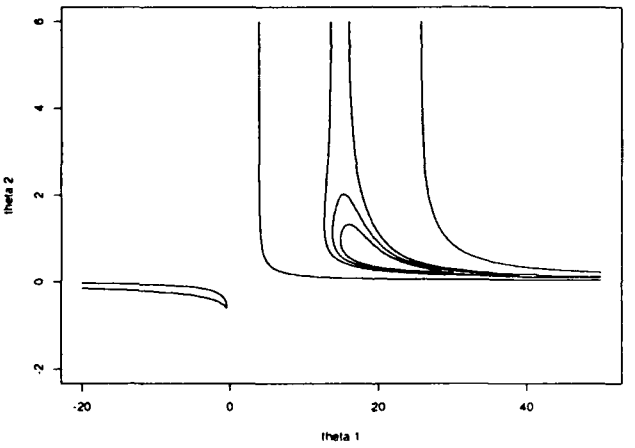


Figure 3: Sum of squares contours for the BOD data. Levels are chosen to give nominal coverage of 80%, 90%, 95%, and 99.9% as confidence regions.

The contours indicate ill-conditioning. Contours at high levels are open in the θ_2 direction and fold over as θ_2 passes zero. Since θ_2 is a rate constant, large values for θ_2 mean that the response will increase rapidly, reaching the asymptote almost instantaneously. If θ_2 is so large that the curve is near the asymptote at the first data value, the sum of squares is insensitive to further increases in θ_2. As $\theta_2 \to \infty$, the response changes instantaneously from zero to the asymptotic level. The $\mathcal{F}$ value for this case defines the level above which likelihood contours will be open in the θ_2 direction. Alternatively, if θ_2 passes zero from above, the

model will become locally overparameterized. If the absolute value of θ_2 is small, the expression $\theta_1 \cdot [1 - \exp(-\theta_2 \cdot t)]$ reduces to $\theta_1 \cdot \theta_2 \cdot t$ and hence the θ_1 that minimizes the sum of squares function for fixed θ_2 will be approximately $\theta_1 = C/\theta_2$ and will jump from $+\infty$ to $-\infty$ as θ_2 crosses zero. In practice one would tend to restrict θ_2 to be positive so the latter effect would not occur. For our purpose of illustration we will leave θ_2 unrestricted and on its original scale.

3.2. *Inference Based on First Order Approximations*

The least squares estimates are $\hat{\theta}_1 = 19.14$ and $\hat{\theta}_2 = 0.53$ with approximate covariance matrix

$$\hat{\Sigma} = \begin{bmatrix} 6.2296 & -0.4323 \\ -0.4323 & 0.0412 \end{bmatrix}. \qquad (3.1)$$

Figure 4 shows 80%, 90%, 95%, and 99.9% contours based on the approximation

$$\phi'(\theta) = \frac{\left[(\theta - \hat{\theta})^T \hat{\Sigma}^{-1} (\theta - \hat{\theta})\right]/p}{S(\hat{\theta})/(n-p)} \sim \mathcal{F}_{p,n-p}.$$

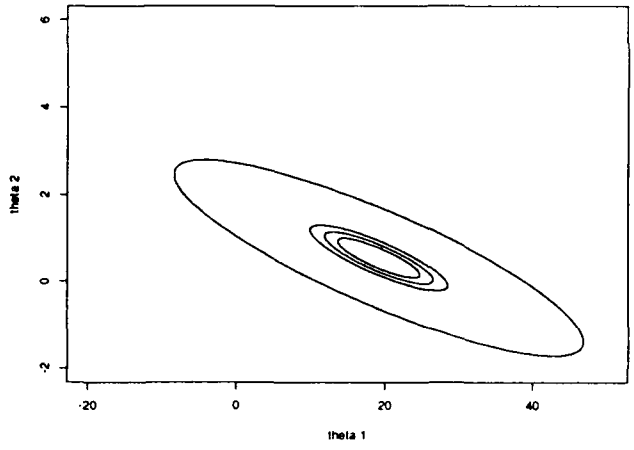


Figure 4: Approximate 80%, 90%, 95%, and 99.9% likelihood contours generated using the linear approximation to the model function.

Clearly, these regions differ greatly from the likelihood regions displayed in Figure 3. Which ones are right? The answer is "both" and "neither". Both sets of contours are based on approximations. While the likelihood contours correspond to parameter pairs which produce fits of equal quality, measured by the sum of squares, the ones obtained from the linear approximation are the correct asymptotic (large sample) contours from a frequentist's point of view. Nevertheless, in this small sample case, the likelihood contours seem more appropriate to us. An additional reason for this is that the validity of the $\mathcal{F}$ approximation for the likelihood contours is only affected by intrinsic nonlinearity while the regions from the linear approximation are affected by both intrinsic nonlinearity and parameter-effects nonlinearity (Bates and Watts, 1988, Chapter 7).

3.3. *Likelihood Profiles*

The likelihood contours in Figure 3 were obtained by evaluating ϕ over a fine grid in θ_1 and θ_2. While this approach is still reasonable for two parameters and a small region of interest, the amount of computation necessary increases exponentially with the number of parameters and quickly surpasses the available computing power. Therefore it is desirable to have methods which can be used to create good approximate likelihood contours with a computing effort which is linear in the number of parameters. Profiling the likelihood is one of these methods. We shall now show how this method performs in the case of the BOD example. Using the definition of τ_i from (2.2), we computed a selection of (θ_1, τ_1) and (θ_2, τ_2) pairs over the intervals $-3.5 \leq \tau_i \leq 3.5$ and obtained approximate $\theta \mapsto \tau$ transformations for both parameters by spline interpolation. Then we constructed approximate likelihood contours by generating ellipses based on the linear approximation in the τ coordinates and transforming them back into θ space. Figure 5 shows the back-transformed contours for the levels 80%, 90%, 95%, and 99.9%.

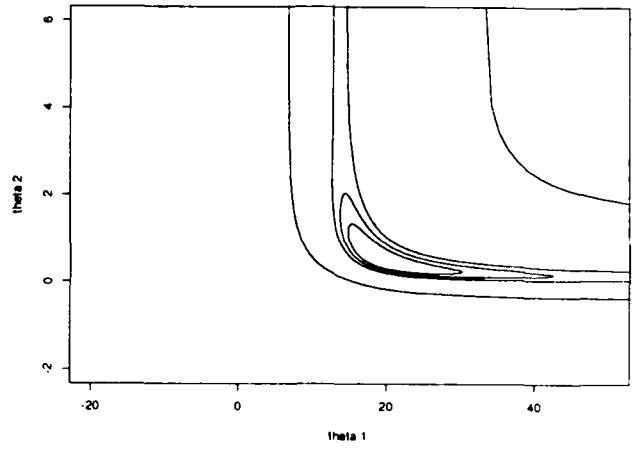


Figure 5: Approximate 80%, 90%, 95%, and 99.9% likelihood contours generated using a linear approximation in the τ parameters and back-transforming to θ.

These contours are already quite similar to the ones computed with the grid method. They can be enhanced further by using the profile traces as described in Bates and Watts (1988, Appendix 6).

3.4. *A Note on Likelihood And Bayesian Methods*

While in likelihood methods one describes features of the likelihood function (maximum, contours, etc) and uses fre-

quentist arguments to attach probability statements to these features, in the Bayesian approach one treats the posterior, whose main part is the likelihood, as a probability distribution and bases the entire inference on it. Using a likelihood together with a flat prior provides a bridge between the likelihood and the Bayesian approaches. However, a likelihood does not always define a proper posterior because the integral over the entire parameter space may be infinite. The BOD problem is an example of this. Since the high level contours are open for large values of θ_2, the integral of the BOD likelihood is infinity. The methods we shall describe now all require a proper posterior, and therefore the BOD likelihood needs to be modified. One way of doing this is by restricting the BOD likelihood to a finite domain such as range of the previous plots. This amounts to an indicator prior on the rectangle $[-20, 50] \times [-2, 6]$ and a flat prior on σ^2. Note that this prior is chosen for the purpose of illustration only. In real life, negative values for θ_2 are impossible. Therefore one should reparameterize the problem by, for example, introducing $\delta = \log \theta_2$ and possibly use a prior which is locally uniform on the expectation surface in the new parameters (Bates and Watts, Chapter 6).

3.5. *Importance Sampling*

Importance sampling is one of the Monte Carlo techniques for exploring posterior distributions. Since we are interested in θ_1 and θ_2, we first have to marginalize the posterior with respect to σ^2. The resulting posterior for θ_1 and θ_2 becomes:

$$p(\theta_1, \theta_2) = C \cdot \left[S(\theta_1, \theta_2)\right]^{\frac{n}{2}-1},$$

with

$$C = \left(\int_{[-20,50]\times[-2,6]} \left[S(\theta_1, \theta_2)\right]^{\frac{n}{2}-1} d(\theta_1, \theta_2) \right)^{-1}.$$

In importance sampling we create a sample $\left\{\theta^{(i)}\right\}_{i=1}^{m}$ from an approximation $I(\theta)$ to the posterior and attach weights ω_i proportional to $p(\theta^{(i)})/I(\theta^{(i)})$ to it. Usually we normalize the weights such that $\sum_{i=1}^{m} \omega_i = 1$. These weighted samples can then be used as substitutes to samples from $p(\theta)$ in forming histograms, integrals, etc.

Our first attempt is to use a multivariate t distribution with the least squares estimate $\hat{\theta} = (19.1426, 0.5310)$ as location parameter and $\hat{\Sigma}$ from (3.1) as the scale matrix. The sample of 10000 observations is shown in Figure 6.

Unfortunately, the highest weight is about 0.17 and the sum of the 10 highest weights is 0.6. Thus, of the 10000 samples, only 10 really enter into any further analysis. This means that the Monte Carlo variance for this sample is very high and that the statistics computed from this sample are essentially useless. Failure of importance sampling due to dominating weights results from gross mismatches between the true

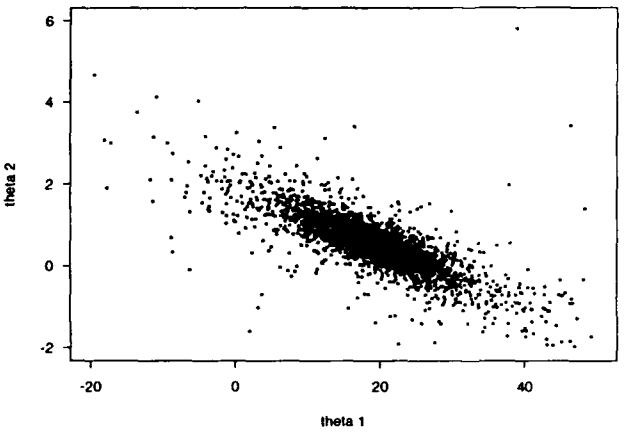


Figure 6: Importance sample for the BOD parameters from a direct approximation with a multivariate t density.

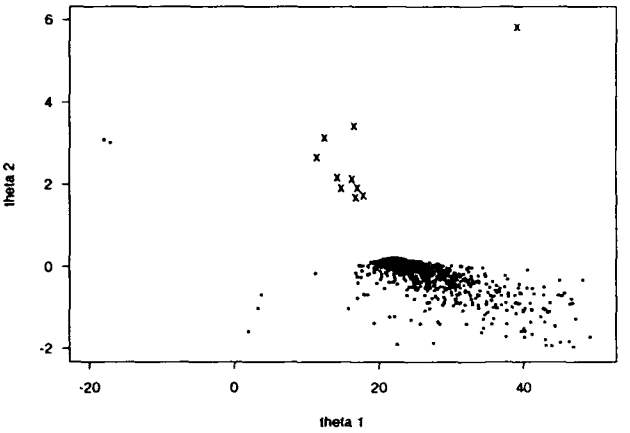


Figure 7: Highest and lowest weights for the importance sample in the original parameters. "x" indicates points of high weight and "." indicates points of low weight.

posterior $p(\theta)$ and the importance distribution $I(\theta)$. In the case of the BOD problem, this mismatch becomes apparent if one compares the contours of the likelihood with the contours based on the linear approximation. The contours of the density corresponding to the multivariate t approximation follow the contours of the linear approximation; the contours of the true posterior follow the likelihood contours. The locations of the 10 highest and the 1000 lowest weights are shown in Figure 7. The sample points with high weights are exactly in places where there is still considerable posterior density but the t density is close to zero. This means that these sample points are likely under the posterior and rare under $I(\theta)$. The weights have to make up for the difference. In turn, the weights which are essentially zero correspond to samples which are likely under the t distribution but rare

under $p(\theta)$.

Since in nonlinear regression, the likelihood, and subsequently the posterior, often has strongly non-elliptical contours, direct importance sampling based on $\hat{\theta}$ and $\hat{\Sigma}$ cannot be recommended. However, if the likelihood profile transformations are available, the situation is better. Then, one can conduct the importance sampling in the τ coordinates using $\hat{\tau}$ and the transformed covariance matrix Σ', which is just the correlation matrix in the original parameters. The likelihood contours in the τ coordinates usually look much more elliptical than in the original coordinates and therefore importance sampling based on multivariate normal or t distributions will work better there. Importance sampling done in τ coordinates helps to eliminate the dominating weights for the BOD example. Figure 8 shows the resulting sample points transformed back to θ coordinates where they trace the likelihood contours quite well.

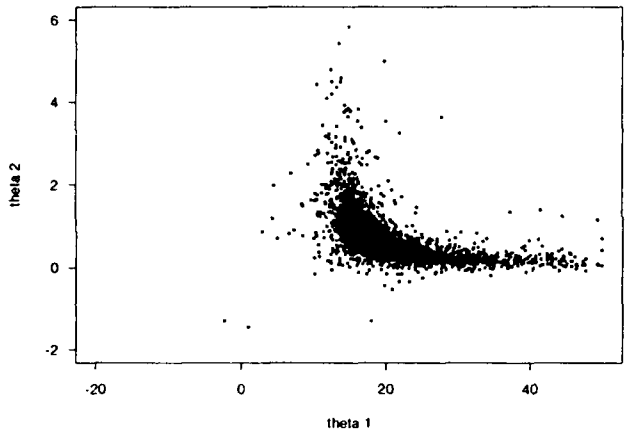


Figure 8: Importance sample from an approximation by a multivariate t density in the τ parameters back-transformed to the θ parameters.

Some caution is needed, however, when doing importance sampling in τ coordinates. Using the likelihood in τ coordinates with a flat prior is not the same as using the likelihood in the original coordinates under a flat prior. In this case a Jacobian of the transformation may be necessary.

3.6. *The Gibbs Sampler*

Rather than sampling from a rough approximation to the posterior and using weights to bridge the gap, one can attempt to sample from the posterior directly. Gibbs sampling is an iterative technique for doing so (Gelfand and Smith, 1990). However, Gibbs sampling in its usual form is not applicable to nonlinear regression since the posterior is only known up to a multiplicative constant and since the conditional distributions are not given explicitly. Grid based Gibbs sampling (Ritter and Tanner, 1990) overcomes this difficulty by working with approximations to the marginal conditionals based on evaluations of the posterior over one dimensional grids. Figures 9 and 10 show how grid based Gibbs sampling starting with 500 uniformly distributed points quickly recovers the characteristic features of the BOD likelihood. In this example the grid based Gibbs sampler was used in its simplest form with 40 equidistant grid points in both θ_1 and θ_2 directions. The sample stabilized after only five to ten iterations. A total of 40 iterations were conducted but no further changes could be observed.

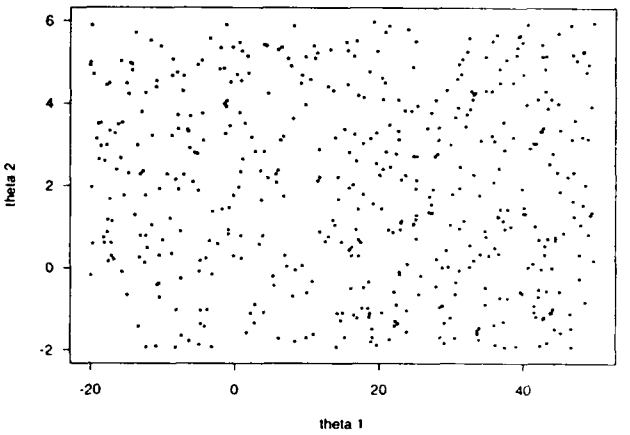


Figure 9: Grid-based Gibbs sampler - starting sample

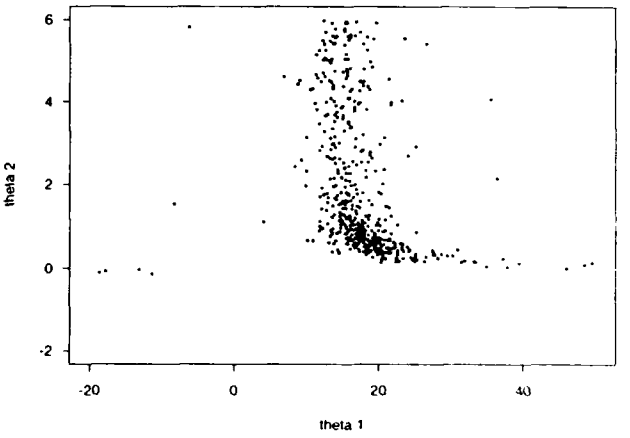


Figure 10: Grid-based Gibbs sample - after 5 iterations.

Grid based Gibbs sampling requires little. For example it does not require a least squares estimate or a covariance matrix. However, it is rather computing intensive. In the above example, the 10 first iterations required 200,000 evaluations of the posterior distribution while for the importance sampling the posterior was evaluated only 10,000 times. Yet, the above implementation of the Gibbs sampler was not the

most efficient. For example, by using flexible grids with fewer points, the amount of computation can be cut easily to about 100,000.

Gibbs sampling is just one member of a larger class of sampling algorithms based on Markov chains. Another member is the Metropolis algorithm which has also been used successfully for the BOD problem. Since these algorithms are very new as tools in nonlinear regression, little can be said about their respective strengths. Further research is needed.

3.7. *Marginal Inference*

Often, one is interested in individual components of the parameter vector or in functions of the parameters. Such inference is notoriously difficult unless one can resort to Monte Carlo type methods. In this context importance sampling and Gibbs sampling show their true strengths although they require a high computing effort. There are other methods for obtaining approximate marginal distributions, (Tierney, Kass, and Kadane, 1988; Leonard, Hsu, and Tsui, 1989) which require less computation. Recent work by Leonard, Hsu, and Ritter reformulates the approximating integrals in a t-type setting and yields for one-dimensional margins

$$p(\theta_j|\mathbf{x},\mathbf{y}) = C \cdot \frac{\tilde{S}(\theta_j)^{-(\frac{n-p}{2}-1)}}{|R^{(j)}|^{\frac{1}{2}}},$$

where $\tilde{S}(\theta_j)$ is as before and $R^{(j)}$ is the Hessian of the conditional sum of squares evaluated at the optimum. Note that in usual applications of Laplace-type approximations, the exponent of $\tilde{S}$ is $-(\frac{n}{2}-1)$. Replacing n by $n-p$ takes into account that the t-type distribution has "$n-p$ degrees of freedom". We shall now demonstrate and compare these methods.

3.8. *Marginal Distribution of Rate Parameter*

Using the previously introduced posterior in θ_1 and θ_2, we can compute the θ_2 marginal by numerical integration or, in this case, analytic integration. We will restrict ourselves to the numerical integration since the direct integration is messy and does not reveal any interesting features. Numerical integration is easy and fast for integrating out one dimensional parameters (in this case θ_1), yet it becomes difficult and computing intensive if the dimension over which the integration is to be conducted increases. In these situations Laplace-type approximations and Monte Carlo techniques are preferable.

Figure 11 shows a comparison of the integrated θ_2 marginal and a marginal histogram derived from the combined Gibbs sample of iterations 6 through 10. The match is very good.

Figure 12 shows the corresponding picture for a histogram derived from the importance sample in the original coordinates. Clearly, there are too few points with large θ_2 component and consequently, the corresponding weights are very high. The spikes in the right part of the picture are caused by single observations with high weights.

Figure 13 shows the histogram for the importance sample created using the profile transforms and a Jacobian. This importance sample performs much better than does the direct one, yet still not as good as the Gibbs sample.

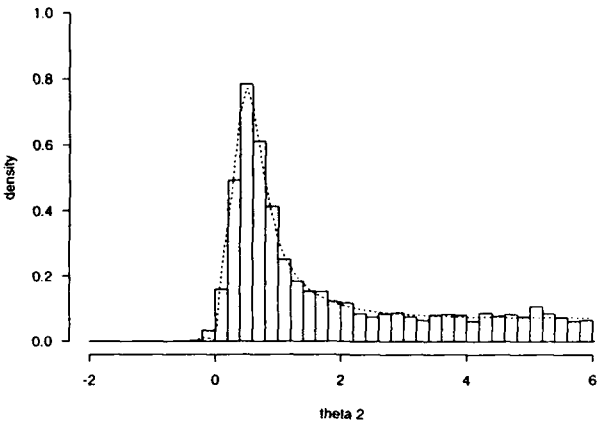


Figure 11: Marginal density for θ_2. The dotted line is from numerical integration and the bars are from pooling iterations 6-10 of the Gibbs sampler.

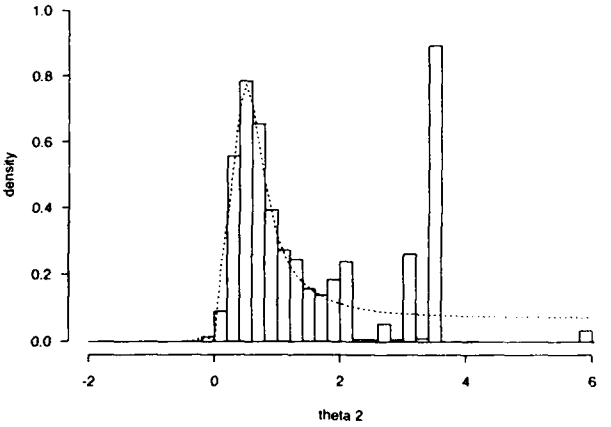


Figure 12: Marginal density for θ_2. The dotted line is from numerical integration and the bars are from the importance sample in the θ parameters.

Finally, Figure 14 shows a comparison of the integrated marginal and the marginal obtained using t-type approximate marginalization. To make the latter marginal comparable with the integrated one, we set the marginal equal to zero for the points where the conditional minimum of the sum of squares fell outside the domain for θ_1 and θ_2 and we normalized the resulting curve to integrate to unity over the θ_2 domain.

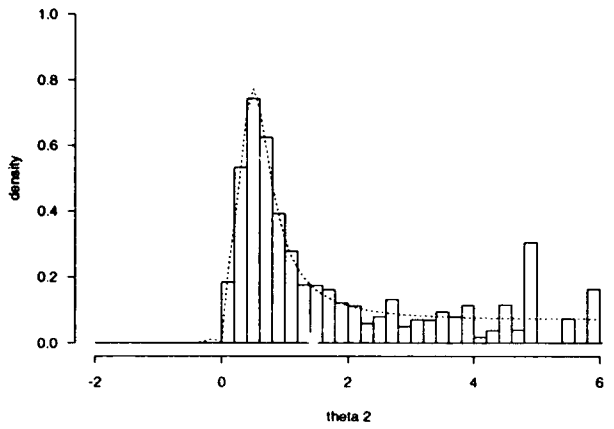


Figure 13: Marginal density for θ_2. The dotted line is from numerical integration and the bars are from the importance sample in the τ parameters.

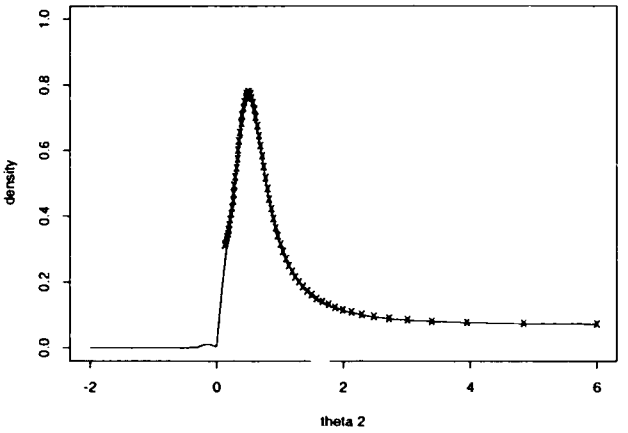


Figure 14: Marginal density for θ_2. The dotted line is from numerical integration and the "" are from the Laplace t approximation.*

The shape of the curves matches perfectly for much of the range. As, however, θ_2 approaches zero, the conditional minimum of S moves outside of the domain. Since the conditional maximization is being done routinely in the profiling algorithm, the approximate margins can be obtained as a inexpensive by-product of profiling.

4. Conclusions

There is still much to be done in comparing approaches to inference, even for the specific case of the nonlinear regression model. The methods we described based on profiling or on Monte Carlo approaches are feasible for small- to medium-sized problems. They show that it is possible routinely to go beyond quoting "asymptotic" standard errors and correlations for parameters. Obtaining an importance sample is relatively straightforward but we would recommend always using the profile-based transformations before obtaining the importance sample. Without transforming to more stable parameters, the Monte Carlo efficiency of the importance sample can be much too low. The Gibbs sampler, and other methods based on Markov chains like the Metropolis algorithm, are very robust but also very expensive. We found that we did have to pay careful attention to the prior distribution of the parameters when using such methods. This may be because of pathologies in the small example we were using but we feel it is to some extent an inherent property of the methods. Luke Tierney, in his discussion of this paper, had several comments to make about reasonable choices of a prior.

The need to consider the choice of prior carefully is a "good news/bad news" type of situation. The good news is that you are forced to look at your model and data carefully and hence create a more informed analysis. The bad news is that "black box"-style automation of the methods becomes much more difficult.

References

Bates, D. M. and Watts, D. G. (1988), *Nonlinear Regression Analysis and Its Applications*, Wiley, New York.

Gelfand, A. E. and Smith, A. F. M. (1990), "Sampling based approaches to calculating marginal densities", *J. of the Amer. Statist. Assoc.*, **85**, 398–409.

Leonard T., Hsu J. S. J, and Tsui K. W. (1989), "Bayesian marginal inference", *J. of The Amer. Statist. Assoc.*, **84**, 1051–1058.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller E. (1953), "Equations of state calculations by fast computing machines", *J. of Chem. Physics*, **21**(6), 1087–1092.

Ritter, C. and Tanner M. A. (1991),"The griddy Gibbs sampler", *Technical Report 878, Dept. of Statistics, University of Wisconsin.*

Rubinstein, R. Y. (1981), *Simulation and the Monte Carlo Method*, Wiley, New York.

Tierney, L. and Kadane, J. B. (1986), "Accurate approximations for posterior moments and densities", *J. of the Amer. Statist. Assoc.*, **81**(393), 82–86.

Tierney L., Kass R. E., and Kadane J. B. (1988), "Approximate marginal densities of nonlinear functions", *Technical Report 419, Dept. of Statistics, University of Minnesota.*

Markov Chain Monte Carlo Maximum Likelihood

Charles J. Geyer*

School of Statistics
University of Minnesota
Minneapolis, MN 55455

Abstract

Markov chain Monte Carlo (e. g., the Metropolis algorithm and Gibbs sampler) is a general tool for simulation of complex stochastic processes useful in many types of statistical inference. The basics of Markov chain Monte Carlo are reviewed, including choice of algorithms and variance estimation, and some new methods are introduced. The use of Markov chain Monte Carlo for maximum likelihood estimation is explained, and its performance is compared with maximum pseudo likelihood estimation.

KEY WORDS: Markov chain, Monte Carlo, Maximum likelihood, Metropolis algorithm, Gibbs sampler, Variance estimation.

1 Introduction

For many complex stochastic processes very little can accomplished by analytic calculations, but simulation of the process is possible using Markov chain Monte Carlo (Metropolis, et al., 1953; Hastings, 1970; Geman and Geman, 1984). The simulation can be used to calculate integrals involved in various forms of statistical inference. Most work in this area has concentrated on Bayesian inference (Geman and Geman, 1984; Gelfand and Smith, 1990; Besag, York, and Mollié, 1991). But Markov chain Monte Carlo is a general tool for simulation of stochastic processes; it should be useful, and has been applied, in other forms of inference.

One such area is likelihood inference. For complex stochastic processes such as the Markov random fields (Gibbs distributions) used in spatial statistics (and other areas, with Markov random fields defined on graphs, networks, pedigrees, and the like) exact calculation of the maximum likelihood estimate (MLE) is impossible, but several methods of Monte Carlo approximation of the MLE have been devised. One uses direct Monte Carlo calculation of the likelihood (Penttinen, 1984; Geyer, 1990; Geyer and Thompson, 1992). Another uses stochastic approximation (Younes, 1988; Moyeed and Baddeley, 1991). A third is that of Ogata and Tanemura (1989). Only the first of these permits the computation of many estimates from one Monte Carlo sample and so permits rapid parametric bootstrap computations and simulation studies. These are important ways of studying the properties of the estimators, and the other methods will not be further discussed. Coding and maximum pseudolikelihood estimates (MPLE) (Besag, 1974, 1975) have also been used for such problems, but these estimators do not approximate the MLE, except in the limit of zero dependence.

Monte Carlo maximum likelihood is illustrated using the two-parameter Ising model as an example. This model is simple enough so that extensive simulations are possible but has most of the complexity of more elaborate models, in particular, the behavior of "freezing," which presents severe problems for maximum pseudolikelihood, but none for maximum likelihood. MLE is compared to MPLE in a case where the random field has strong dependence (is near freezing) where the superiority of MLE over MPLE is clearly shown.

2 Markov Chain Monte Carlo

Before discussing the use of Markov chain Monte Carlo for maximum likelihood, it is first necessary to briefly review these Markov chain methods, since the literature is confused and contains some bad advice.

Markov chain Monte Carlo is an old method of simulation that goes back to the dawn of the computer age, but which has had, until recently, little application in statistics. The main idea is very simple. In ordinary Monte Carlo, if one wishes to evaluate an integral

$$P g = \int g \, dP, \tag{1}$$

where P is a probability measure and one has a method of simulating a sequence $X_1, X_2, \ldots$ of i. i. d. realizations

*Research supported in part by NSF grant DMS-9007833. Some of this work was contained in the author's Ph. D. dissertation done at the University of Washington under the supervision of Elizabeth Thompson, some was done during a postdoctoral year at the University of Chicago

from P, the obvious estimate is

$$\mathbf{P}_n g = \frac{1}{n} \sum_{i=1}^{n} g(X_i), \tag{2}$$

since

$$\mathbf{P}_n g \xrightarrow{\text{a. s.}} P g \tag{3}$$

by the strong law of large numbers whenever g is P-integrable. The notation in (1) and (2) is standard in the empirical process literature and very convenient; (1) treats the symbol P interchangeably as a measure and as an operator, (2) treats the empirical measure (the measure-valued stochastic process that puts mass $1/n$ at each of the points X_i in the sample) the same way. Though ordinary Monte Carlo is very powerful, it has its limitations. In particular there are no general methods for simulating independent realizations of multivariate random vectors or, more generally, from complex stochastic processes. This difficulty is gotten around by Markov chain Monte Carlo in which one simulates not independent realizations from P but a Markov chain X_1, X_2, ... with stationary transition probabilities having P as a stationary distribution. If the chain is irreducible, (3) still holds, though it is now referred to as the ergodic theorem rather than the strong law of large numbers.

Since a countable union of null sets is a null set, (3) can be taken to hold simultaneously (for the same null set of sample paths of the Markov chain) for all functions g in any countable family. If the state space of the Markov chain (the sample space of the measure P) is a second countable topological space (such as $\mathbf{R}^d$) and the countable family of functions is taken to be indicators of open sets in the countable base, then, for almost all sample paths of the Markov chain,

$$\mathbf{P}_n 1_B \xrightarrow{\text{a. s.}} P 1_B, \qquad \text{for all open sets } B,$$

that is

$$\mathbf{P}_n \xrightarrow{\mathcal{D}} P \tag{4}$$

(the empirical converges in distribution to the truth).

This is the sense in which Markov chain Monte Carlo "works." The samples X_1, X_2, ... are neither independent nor identically distributed, and none has marginal distribution P (though typically the marginal distribution of X_n is close to P for large n). They behave like samples from P, however, in the sense that (4) holds, just as if X_1, X_2, .. were i. i. d. P.

Some confusion in the literature has resulted from failure to understand this basic nature of Markov chain Monte Carlo. One sees described without justification in various places the following way to do Markov chain Monte Carlo. Let X_{11}, ..., X_{m1} be independent realizations from some distribution. For $j = 1, \ldots, m$, simulate X_{j2}, ..., X_{jn} a Markov chain starting at X_{j1}, all m chains having the same transition probabilities and stationary distribution P. Take

$$\frac{1}{m} \sum_{j=1}^{m} g(X_{jn}) \tag{5}$$

as an estimate of $\int g\, dP$. This formula, which may be referred to as the "many short runs" school of Markov chain Monte Carlo (as opposed to the "one long run" school) has some problems. As $m \to \infty$ (5) converges to something by the strong law of large numbers; it does not, however, converge to $\int g\, dP$. That would require that both m and n go to infinity. One can, of course, collect multiple samples in each short run, and this does ameliorate the problem but relies on the "short" runs actually being "long." The closer many short runs is made to one long run, the better it is. This was well understood in the statistical physics literature and in some of the early statistics literature, but needs reiteration.

This is not a purely theoretical point; many short runs also has practical drawbacks. To see these we need some discussion of the practice of Markov chain Monte Carlo. Typically a chain is run for a while to "forget" its starting point before samples are collected; then the chain is subsampled, a sample being taken every kth step. The number of samples m thrown away at the beginning of the chain will be termed the "burn-in" (there is no standard terminology), and k will be termed the "spacing." The empirical estimate for such a subsample is defined by

$$\mathbf{P}_n g = \frac{1}{n} \sum_{i=1}^{n} g(X_{m+k\,i}), \tag{6}$$

rather than (2). Of course the subsample is again a Markov chain with stationary transition probabilities, and (3) still holds. The reasons for choosing any m other than zero and any k other than one have not been made clear. The spacing k is often chosen to be large in order that the samples $X_{m+k\,i}$ be "almost independent" as if reliance were being placed on some hypothetical "almost" law of large numbers rather than the ergodic theorem. Simple variance calculations, which will be explained below, show that in many cases $k = 1$ is optimal and in almost all cases the optimal k is less than five. The role of the burn-in m is also not well understood. It is often thought that m must be chosen large enough so that X_m "almost" has marginal distribution P, something that typically cannot be checked. This leads to using very large m for "safety." If the one long run method is being used, a fairly large burn-in, say five per cent of the total run length, is not excessive and will usually be more than adequate. In any case, the accuracy of the method is relatively insensi-

tive to the burn-in. Even inadequate burn-in will have only a small effect on the results. The many short runs method perversely arranges the calculation so that not only does burn-in dominate the cost of the calculation (the method is really only valid as the burn-in becomes infinite), but also the accuracy critically depends on the adequacy of burn-in, which is uncheckable. The many short runs method arranges to have many burn-ins at much cost and to no benefit.

At this point many people remark that even if one is willing to concede the point just made, multiple runs have some diagnostic value, at least. This is, of course, correct. It is clear that if two runs produce completely different answers, the runs are too short. But this diagnostic value is a "one-edged" sword. It is *not* valid to draw any comfort from the agreement of short runs, even many short runs. Counterexamples exist that prove such hopes illusory. The best diagnostic is a very long run, which will find places in the state space that one never thinks to start.

With these general comments out of the way, we now turn to specific algorithms. The first Markov chain Monte Carlo method was given by Metropolis et al. (1953) and is generally known as the "Metropolis algorithm." This algorithm received wide use in the statistical physics community from the beginning, but has, even today, had little use in the statistics community.

Suppose the desired stationary distribution has a density p with respect to some measure μ. The algorithm employs an auxiliary function $q(y, x)$ such that $q(\cdot, x)$ is a probability density with respect to μ for each x and $q(x, y) = q(y, x)$ for all x and y. The Markov chain is generated by repeatedly applying the following update step.

1. simulate y from the distribution with density $q(\cdot, x)$.
2. calculate the odds ratio $r = p(y)/p(x)$
3. if $r \geq 1$ go to y
4. if $r < 1$ go to y with probability r, else stay at x

Simple calculations show that the Metropolis algorithm has the desired distribution with density p as one stationary distribution (see, for example, Ripley, 1987). If the chain can be shown to be irreducible (which depends on the specific structure of p and q), it is ergodic and can be used for Monte Carlo.

One problem with the Metropolis algorithm is the requirement that q be symmetric. Hastings' (1970) algorithm drops this requirement. In order to maintain the correct stationary distribution, this requires that in step 2 of the Metropolis update, r be redefined as

$$r = \frac{p(y)}{p(x)} \frac{q(x, y)}{q(y, x)}$$

(so it can no longer be called an "odds ratio.") The algorithm works just as well with this modification. The Hastings algorithm allows an essentially arbitrary choice of "candidate" points.

A more recent algorithm is the Gibbs sampler (Geman and Geman, 1984). This algorithm is applicable only when the state variable is a random vector $x = (x_1, \ldots, x_p)$; it does not apply to arbitrary state spaces. At each step one variable, say x_i, is changed by giving it a realization from the conditional distribution of x_i given the rest of the variables under the stationary distribution.

Though this looks very different from the Metropolis and Hastings, it is almost a special case of the Hastings algorithm in which the one-dimensional conditional distributions play the role of the auxiliary function q. The analogy with Hastings does suggest that when one cannot sample exactly from the one-dimensional conditionals, one can do a Hastings-like rejection to correct inexact sampling, as long as one does know the density one is sampling from. For more on this subject see Besag (this volume).

3 New Methods

All of the literature on Markov chain Monte Carlo describes using chains with all Metropolis update steps (a Metropolis algorithm) or pure Gibbs steps (a Gibbs sampler), although there is no reason for this. Any steps that preserve the stationary distribution can be mixed in any order. To make a chain with stationary transition probabilities, it is necessary that a fixed sequence of steps (called a "scan") be repeated over and over and that samples be collected only after complete scans or multiples of complete scans. This is typical for the Gibbs sampler, a scan consisting of updating each x_i, running through the variables in some fixed order. But much more general scans are possible. There is no reason not to mix Gibbs, Metropolis, and Hastings steps in a single chain, or for that matter, other update steps yet to be invented. Large increases in speed can be obtained by clever choices of update steps.

A simple example is to attempt to make a variety of steps of various sizes. When the distribution of interest has two (or more) modes, it is important to make attempts to jump from one mode to the other, if at all possible. This will be illustrated below in the discussion of the Ising model, where the modes are roughly symmetrically distributed in the sample space and hence

easy to identify and one can jump between modes via a "symmetry swap," changing the sign of all variables at once. Metropolis rejection of the swaps steps preserves the desired stationary distribution.

It is not always possible to find steps that jump between modes, or even to find out (apart from Monte Carlo experiments) how many modes there are. What is needed is some way to make large steps without explicit detailed knowledge about the distribution of interest. A device which we are calling Metropolis-coupled Markov chain Monte Carlo, $(MC)^3$ for short, provides a way to do this (Geyer, 1991b). Suppose we run m Markov chains in parallel, having different, but related, equilibrium distributions, $P_1, \ldots, P_m$. For example, if the distribution of interest is a Gibbs distribution with density proportional to $e^{U(x)/\tau}$, $U(x)$ being the potential function and τ the temperature, we could take P_k to have density proportional to $e^{U(x)/k\tau}$. After each scan (in which all of the chains attempt one step for each variable) we attempt to swap the states of two of the chains. This is a Metropolis update since swapping is symmetric, so the swap of chains i and j is accepted or rejected according to the odds ratio

$$r = \frac{p_i(x_j)p_j(x_i)}{p_i(x_i)p_j(x_j)}. \tag{7}$$

The coupling induces dependence among the chains, and they are no longer (by themselves) Markov. The whole stochastic process (the m chains together) does form a Markov chain on the m-fold cartesian product of the original state space. Since (7) is the odds ratio assuming independence of the distributions for the chains, the stationary distribution of the whole process, is the product of the P_i. The chains are asymptotically independent with the desired stationary distributions.

If the coupling does not change the stationary distributions, what is the point? It may make all of the chains mix much faster, faster than any one of them uncoupled. This effect is due to the chains having different distributions. It is clear that if the distributions are the same, every swap is accepted and the chains produce the same realizations with or without swapping. If one untangles the swapped chains (following one state as it jumps back and forth among the distributions), one gets a different process. Now, by symmetry, all of the untangled chains have the same marginal distribution, though they are no longer even asymptotically independent, and this marginal distribution must be the equal mixture of the distributions P_i. This says that in some sense the speed of the chains is that of a mixture of the update steps for the separate chains. This mixture may run faster than any of the pure chains.

Examples of these devices will be given later after the Ising model is described. For now, let us close this section with the point that if one is worried that the Gibbs sampler, or whatever Markov chain scheme one is using, mixes too slowly, one should try to speed it up. There are many possible tricks for doing so. These are examples of what is possible.

4 Variance Calculations

Given the consistency (3) of Markov chain Monte Carlo, the natural next question is to examine the error $\sqrt{n}(\mathbf{P}_n g - Pg)$. Typically one would like there to be a central limit theorem

$$\sqrt{n}(\mathbf{P}_n g - Pg) \xrightarrow{\mathcal{D}} N(0, \sigma_g^2) \tag{8}$$

(note that σ_g^2 depends on g). When the state space of the Markov chain Monte Carlo is finite, the central limit theorem (8) always holds, (see, for example, Chung, 1967, p. 99 ff. or Ibragimov and Linnik, 1971, pp. 365–369). There are Markov chain central limit theorems for nonfinite state spaces, but the regularity conditions seem difficult to apply (this is a subject of active research by a number of investigators).

Markov chain limit theory is of use only in demonstrating that (8) holds with σ_g^2 finite; it does not yield the value of σ_g^2, which must be estimated from the Markov chain. This is easily done using standard time-series methods. Hastings (1970) gave references to methods then current; only slight changes are needed to bring these recommendations up to date. In cases of practical interest σ_g^2 will have the form

$$\sigma_g^2 = \sum_{t=-\infty}^{\infty} \gamma_t \tag{9}$$

where

$$\gamma_t = \gamma_{-t} = E\left(g(X_0), g(X_t)\right)$$

the expectation being with respect to the stationary distribution. The γ_t are easily estimated by

$$\hat{\gamma}_t = \gamma_{-t} = \frac{1}{n}\sum_{i=1}^{n-t} g(X_i)g(X_{i+t})$$

For why we divide by n rather than $n - t$ see Priestly (1981, pp. 323-324). One might think that the sum of the $\hat{\gamma}_t$ would be a natural estimator of σ_g^2, but this is a bad idea for the following reason. For large t the variance of γ_t is approximately constant

$$\mathrm{Var}(\hat{\gamma}_t) \approx \frac{1}{n}\sum_{s=-\infty}^{\infty} \gamma_s^2 \tag{10}$$

(Bartlett, 1946); the right hand in (10) does not depend on t. This assumes that $g(X)$ has a fourth moment and

that some mixing condition holds (ρ-mixing suffices). Thus the sum of the γ_t differs from (9) by n terms of size $1/n$. It does not decrease with n; the estimate is not even consistent. In order to get a good estimate it is necessary to downweight the terms for large lags, which are essentially noise. One estimates σ_g^2 by

$$\hat{\sigma}_g^2 = \sum_{t=-\infty}^{\infty} w(t)\gamma_t \tag{11}$$

where w is some weight function that satisfies $w(t) = 1$ for small t, $w(t) = 0$ for large t, and makes a smooth monotone transition between these levels.

The right hand side of (10) is useful in choosing w. One can take $w(t) = 1$ for t such that γ_t exceeds two "large t" standard deviations. Since it is usually impossible to arrange a chain with significant negative autocorrelations, one can take $w(t) = 0$ when $\gamma_t < 0$ and for all larger t. Any smooth curve connecting these two points is satisfactory. We use a scaled cosine.

Before leaving this subject, the frequency domain version of the same procedure should perhaps be explained, since one may see this described instead and the equivalence of the two methods is not obvious. (9) is 2π times the value of spectral density at the origin (of the time series $g(X_t)$). To estimate the spectral density one may use a kernel smoother with kernel $\tilde{w}$ on the empirical spectral estimate, which is the Fourier transform of the γ_t. If one uses the Fourier transform of w for the smoothing kernel $\tilde{w}$, one obtains exactly the same estimate as (11). In the usual time-series parlance w is called a lag window and $\tilde{w}$ a spectral window.

5 Choosing the Spacing

Having a method of estimating variances gives us a method of measuring the "speed" of a Markov chain scheme. A chain is rapidly mixing if the autocorrelations decrease rapidly enough so that the variance of our estimate(s) of interest is small. This is a relative term, we can only say that one chain mixes more rapidly than another. There is no absolute standard.

One obvious comparison is between chains that are alike except for different spacing. Suppose that the chain is ρ-mixing (always true if the state space is finite) so the γ_t decrease exponentially fast. Then the asymptotic variance for a chain with spacing k will be

$$s_k = \sum_{t=-\infty}^{\infty} \gamma_{kt} \leq \gamma_0 + 2\frac{A\rho^k}{1-\rho^k}$$

for some constants $A > 0$ and $0 < \rho < 1$. Clearly as $k \to \infty$ the variance s_k converges to the marginal variance γ_0 that would be obtained if one could do independent sample Monte Carlo. Since the convergence is exponentially fast, there is little benefit to large spacings. To see this more clearly, let B be the cost of sampling (typically computer time), and let C be the cost of "using" a sample. If the samples cost almost nothing to use, one may take $C = 0$. If one uses n samples with spacing k, the cost is $Bnk + Cn$, because the chain runs for nk steps and n samples are used. The variance of the estimate is approximately s_k/n. Hence to get a fixed accuracy one must have n proportional to s_k. Thus the cost for spacing k is proportional to $(Bk + C)s_k$. For large k this increases linearly in k. The minimum cost will be attained for some small value of k, the optimal spacing. Note that if $C = 0$ the optimal spacing is greater than one only if $s_1 > 2s_2$, which is typically not the case. One needs some cost of using samples (cost of calculating estimates, cost of storing samples, plotting samples, or whatever) to make subsampling a good idea.

If one is interested in calculating integrals of many functions g, there is no one spacing that is optimal for all, nor would one want to do variance calculations for all. Fortunately, this is not necessary. Typically the cost curves will be U-shaped with a broad bottom and the curves for a representative sample of functions will have minima in roughly the same place. We do not recommend elaborate variance calculations accompanying every Markov chain Monte Carlo estimate, but there is no substitute for *some* variance calculations for comparing methods, for selecting spacings, and just generally getting a feel for how well a scheme works.

6 The Ising Model

The model employed for our example is a standard two-parameter Ising model on a 32×32 square lattice with periodic boundary conditions. Let x_i denote the random variable at lattice site i which takes values in $\{-1, 1\}$, and $x = \{x_i\}$ denote the whole random field. Let $i \sim j$ denote that sites i and j are nearest neighbors. Every site has four neighbors, since the lattice is considered a torus. The statistical model is a two-parameter exponential family with natural statistics $t_1(x) = \sum_i x_i$ and $t_2(x) = \sum_i \sum_{j \sim i} x_i x_j$. For concreteness we will call the lattice sites with $x_i = 1$ "white pixels" and the rest "black pixels" following the language of image processing. Then t_1 is the excess of white over black pixels, and t_2 is the excess of concordant nearest neighbor pairs over discordant pairs.

The probability of a point x in the sample space is

$$p_\theta(x) = \frac{1}{c(\theta)} e^{\langle t(x), \theta \rangle}$$

where $\langle t, \theta \rangle = t_1\theta_1 + t_2\theta_2$ and

$$c(\theta) = \sum_{x \in \mathcal{S}} e^{\langle t(x), \theta \rangle}. \tag{12}$$

The parameters θ_1 and θ_2 are referred to here as the "level" parameter and "dependence" parameter respectively. We also use the notation $\alpha = \theta_1$ and $\beta = \theta_2$.

At $\beta = 0$, the pixels are independent; for large β the distribution has two modes, almost all of the pixels are the same color with just a speckle of the other. The proportion of realizations that are predominantly white or black depends on α; when $\alpha = 0$, the modes are equally probable. This behavior occurs for all lattice sizes, even for an infinite lattice, where the transition from patches of both colors to (almost) all one color occurs sharply at the critical value $\frac{1}{2}\sinh^{-1}(1) = 0.4407$. The transition is not sharp for finite lattice sizes, but occurs in roughly the same place.

For any lattice site i, let x_{-i} denote the rest of the variables besides x_i. The conditional distribution of x_i given x_{-i} plays an important role in both likelihood and pseudolikelihood methods. This conditional distribution is denoted $p_\theta(x_i|x_{-i})$. Let $n_i = \sum_{j \sim i} x_j$ denote the sum of the nearest neighbors of lattice site i. Then

$$\begin{aligned} \operatorname{logit} p_\theta(x_i = 1|x_{-i}) &= \operatorname{logit} p_\theta(x_i = 1|n_i) \\ &= 2(\theta_1 + \theta_2 n_i). \end{aligned} \tag{13}$$

The first equality, that the distribution of x_i given the rest depends only on its neighbors, is called the spatial Markov property. It simplifies calculations, but otherwise plays no role in the analysis.

A Metropolis algorithm for the Ising model runs over the variables in either fixed or random order attempting to swap the state of the variable at each step (from 1 to -1 or vice versa) according to the odds ratio of these two states. A Gibbs sampler does the same thing but instead samples from the conditionals. Metropolis makes more transitions and hence is a bit better, but there is not much difference.

Whichever is used, it is wise to follow each scan of all the variables with a symmetry swap, attempting to change x for $-x$, where $-x$ denotes the state derived from x by changing the sign of all the variables. The odds ratio for this swap is $r = \exp(t_1(-x)\alpha - t_1(x)\alpha)$ since $t_2(x) = t_2(-x)$. When α is small and β is large so the model has a bimodal distribution, these swaps jump between modes. For other parameter values, the swaps are not useful, but they are also not needed since the distribution is unimodal and the Markov chain mixes rapidly in any case. The swaps do no harm, though, since they consume a small fraction of the running time.

With symmetry swaps the Markov chain for the Ising model runs fast no matter what the parameter values, *provided* it is started in the right place: all pixels the same color. If one chooses a random starting point, and β is well above the critical point, it takes a very long time to get to any likely configuration.

Symmetry swaps solve all difficulties of simulating Ising models (and other lattice processes with only a few colors). Hence Metropolis-coupling is not needed. To avoid introducing another model, however, let us also solve the Ising model difficulties using Metropolis coupling. At values of β well below the critical value, a single chain runs fast, the distribution is unimodal, and the region of high probability is rapidly explored. For very high β the chain runs arbitrarily slowly; the waiting time for a transition between modes can be arbitrarily long. If low and high β chains are coupled with a sequence of intermediate β chains, swaps will occur frequently if adjacent β's are close enough, and all of the chains will mix rapidly. Thus Metropolis coupling can produce an arbitrarily large speed up in some situations. This solution to problems of slow mixing is completely general, it does not even require knowledge of a good starting point (as did symmetry swapping). All that is required is that some of the coupled chains mix rapidly.

It is possible to get an infinite speed up from coupling chains. If one couples a chain that is not ergodic (so that it would never get the right answer) with one that is, this can make both chains ergodic. Thus coupling can be used to solve difficult problems of finding a Markov chain that is ergodic as well as problems of slow mixing.

7 Monte Carlo Maximum Likelihood

Consider a family of probability densities $\{f_\theta\}$ with respect to some measure μ, where the densities are known only up to a normalizing constant

$$f_\theta(x) = \frac{1}{c(\theta)} h_\theta(x)$$

where h_θ is a known function for each θ but nothing is known about c except that

$$c(\theta) = \int h_\theta(x)\, d\mu(x),$$

the integral being analytically intractable. The Ising model serves as an example with $h_\theta(x) = e^{\langle t(x), \theta \rangle}$. Other examples include spatial lattice and point processes, Markov graphs, logistic regression with dependent responses (see Geyer and Thompson, 1992).

The unknown normalizing constant c is no bar to Markov chain Monte Carlo which can provide a sample $X_1, X_2, \ldots$ from any θ in the parameter space. This can be used to estimate the log likelihood ratio for an

observation x

$$l(\theta) = \log \frac{f_\theta(x)}{f_\phi(x)} = \log \frac{h_\theta(x)}{h_\phi(x)} - \frac{z(\theta)}{z(\phi)} \qquad (14)$$

as follows. Since

$$\frac{z(\theta)}{z(\phi)} = \frac{1}{z(\phi)} \int h_\theta(x)\, d\mu(x) = E_\phi \frac{h_\theta(X)}{h_\phi(X)}$$

we have the natural estimate

$$\log \left(\frac{1}{n} \sum_{i=1}^{n} \frac{h_\theta(X_i)}{h_\phi(X_i)} \right) \qquad (15)$$

of the last term in (14). Let $l_n(\theta)$ denote (14) with the last term replaced by (15). By the ergodic theorem we have that $l_n(\theta) \to l(\theta)$ simultaneously for all θ in any countable set, which if the parameter takes values in $\mathbf{R}^d$ may be chosen to be dense. This along with the "usual" regularity conditions may be enough to ensure that if $\hat{\theta}_n$ is any maximizer of l_n and $\hat{\theta}$ the maximizer of l, then $\hat{\theta}_n \xrightarrow{\text{a. s.}} \hat{\theta}$, i. e., the Monte Carlo MLE converges to the true MLE as the size of the Monte Carlo sample goes to infinity. For the Ising model no regularity conditions are needed because both l and l_n are concave functions. Second order theory, $\sqrt{n}(\hat{\theta}_n - \hat{\theta})$ converging to some normal distribution is also available, again under the "usual" regularity conditions, when the asymptotic variance of $\sqrt{n}\nabla l_n(\hat{\theta})$ can be shown to be finite, since this can then be estimated empirically using the methods of Section 4. Details will appear elsewhere.

This method can be generalized to use Monte Carlo samples from distributions other than those in the parametric family, in particular to mixtures of distributions in the family. This improves performance when θ is far from ϕ, and is the method used for the example in Figure 1. Details of the theory and the calculation of this example are given in Geyer (1991a).

Given that maximum likelihood can be done, how well does it compare with other methods? Is it worth the effort of the elaborate Monte Carlo calculations? What is analytically tractable about the Ising model (and other Markov spatial processes) is the conditional distributions $p_\theta(x_i = 1|x_{-i})$ defined by (13). The pseudolikelihood is the product of these conditionals. This is not, of course, a likelihood, since these conditionals do not combine in the right way to make a probability. The MPLE is found by maximizing the log pseudolikelihood

$$\psi(\theta) = \sum_i \log p_\theta(x_i = 1|x_{-i})$$

(Besag, 1975). For the Ising model this is computationally equivalent to doing a logistic regression of each pixel on its neighbors. The estimate takes negligible time to compute compared to Monte Carlo MLE.

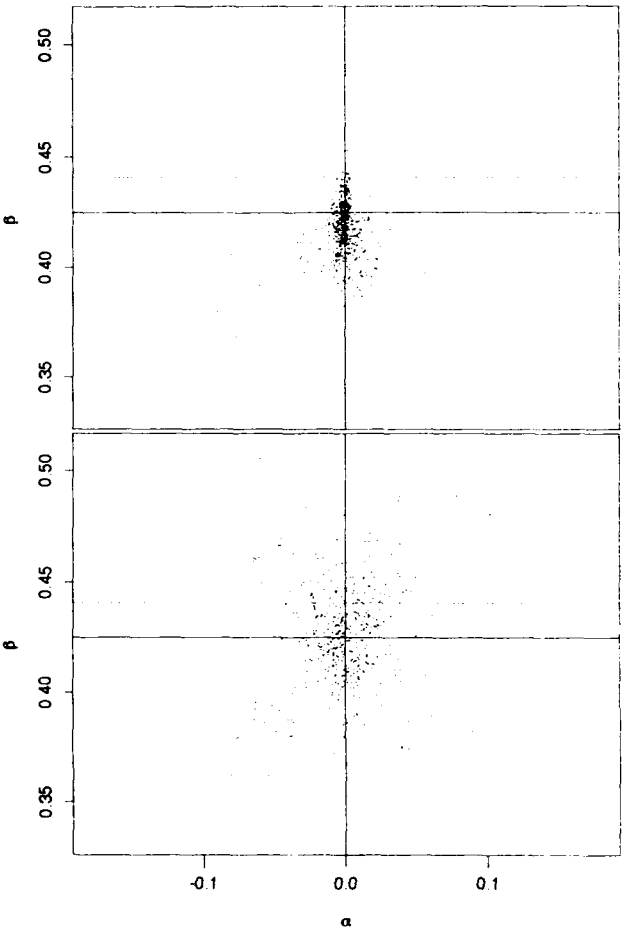


Figure 1: Comparison of MLE and MPLE. Top MLEs, bottom MPLE for sample of 500 points from Ising model with $\alpha = 0$ and $\beta = 0.425$.

Furthermore, it is a good estimate for small dependence, when $p_\theta(x_i = 1|x_{-i}) \approx p_\theta(x_i)$ when it well approximates maximum likelihood. For high dependence, MPLE can do much worse than MLE, as shown in Figure 1. The true parameter value is where the solid lines cross. Both estimators cluster around the truth, but MPLE has much wider scatter. Moreover, maximum likelihood "senses" the critical point, shown by the dotted line, in a way that MPLE does not. Of the 500 points in the sample, only six are above the critical point, only two appreciably so. The dotted line in the figure is like a cliff of the likelihood surface. These samples from a process below the critical point do not look at all like they came from a process above the critical point.

Pseudolikelihood is oblivious to the critical point, which is not surprising, since it only looks at local dependence and the critical point phenomenon is a global property. There are 134 of the MPLE lying above the critical point. Some so high that true realizations

from such parameter values would be hard frozen, not remotely resembling the observation from which the MPLE was calculated.

8 Discussion

Though consistency and asymptotic normality of MPLE has been proved in a variety of situations, these results do not guarantee good behavior at finite sample sizes. It has never been claimed that MPLE would provide good estimates for parameters of a frozen (or nearly frozen) Markov random field, so the message that in some cases MLE behaves well when MPLE does poorly is no surprise. That MPLE can be inefficient had been noted for Gaussian random fields on lattices (Besag, 1977), where the efficiency goes to zero at the boundary of the parameter space where stationarity is lost. Moderately large efficiency is maintained, however, for fairly large dependence, which gives the impression that MPLE is a reasonable method of estimation for Gaussian fields so long as the true parameter value is not near the boundary of the parameter space.

Ising models and other non-Gaussian random fields can have critical parameter values not on the boundary of the parameter space at which the qualitative behavior of the field changes. Near such values, and for high dependence in general, MPLE can give bad results. One Ising model example is given here; a more complex example is given in Geyer and Thompson (1992). This does not say MPLE is bad in all problems; it seems that comparisons must be made problem by problem.

Acknowledgement

The author wishes to thank Julian Besag, Augustine Kong, Alan Lippman, Elizabeth Thompson, and Luke Tierney for discussions of this subject.

References

Bartlett, M. S. (1946) On the theoretical specification of sampling properties of autocorrelated time series. *J. R. Statist. Soc. Suppl.* 8:27–41.

Besag, J. (1974) Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. R. Statist. Soc. B* 36:192–236.

—— (1975) Statistical analysis of non-lattice data. *Statistician* 24:179–195.

—— (1977) Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrika* 64:616–618.

Besag, J., York, J. and Mollié, A. (1991) Bayesian image restoration, with two applications in spatial statistics (with discussion). *Ann. Inst. Statist. Math.* 43:1–59.

Chung, K. L. (1967) *Markov Chains with Stationary Transition Probabilities*, 2nd ed. Springer-Verlag.

Gelfand, A. E. and Smith A. F. M. (1990) Sampling-based approaches to calculating marginal densities. *J. Am. Statist. Assoc.* 85:398–409.

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.* 6:721–741.

Geyer, C. J. (1990) *Likelihood and Exponential Families*. Ph. D. Thesis. University of Washington.

—— (1991a) Reweighting Monte Carlo Mixtures. in preparation.

—— (1991b) Metropolis-Coupled Markov Chain Monte Carlo. in preparation.

Geyer, C. J. and Thompson, E. A. (1992) Constrained maximum likelihood and autologistic models with an application to DNA fingerprinting data (with discussion). *J. R. Statist. Soc. B* to appear.

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57:97–109.

Ibragimov, I. A. and Linnik, Yu. V. (1971) *Independent and Stationary Sequences of Random Variables*. Wolters-Noordhoff.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21:1087–1092.

Moyeed, R. A. and Baddeley, A. J. (1991) Stochastic approximation of the MLE for a spatial point pattern. *Scand. J. Statist.* 18:39–50.

Ogata, Y. and Tanemura M. (1989) Likelihood Estimation of Soft-Core Interaction Potentials for Gibbsian Point Patterns. *Ann. Inst. Statist. Math.* 41:583–600.

Penttinen, A. (1984) Modelling interaction in spatial point patterns: Parameter estimation by the maximum likelihood method. *Jyväskylä Studies in Computer Science, Economics, and Statistics* 7

Priestly, M. B. (1981) *Spectral Analysis and Time Series*. Academic Press.

Ripley, B. D. (1987) *Stochastic Simulation*. Wiley.

Younes, L. (1988) Estimation and annealing for Gibbsian fields. *Ann. Inst. H. Poincaré Probab. Statist.* 24:69–294.

Parallel and Sequential Implementations for Combining Belief Functions*

Mary McLeish and Fei Song
Dept. of Computing and Information Science
University of Guelph
Guelph, Ontario, Canada N1G 2W1

Abstract

This paper reports our experiments about parallel and sequential implementations for combining belief functions with an application to a medical diagnostic system. We use as a basis existing methods for combining two belief functions: a direct combination based on Dempster's rule and an indirect combination through Möbius transforms. We further explore various parallel algorithms for combining more than two belief functions, as different belief functions can be combined in any order as long as they are independent of each other. Our results indicate that for the general case, the parallel implementation based on fast Möbius transforms proves to be the most efficient. However, for practical applications where most subsets of a frame of hypotheses have zero probabilities, the parallel implementation based on an improved direct combination rule remains the most efficient.

1 Introduction

This paper presents parallel and sequential algorithms for combining belief functions. The *Belief Function* approach for approximate reasoning, also called the Dempster-Shafer theory [Shafer, 1976], can be seen as a generalization of the *Probability* approach [Pearl, 1988], since probabilities are assigned directly to subsets of a set of mutually exclusive and exhaustive hypotheses rather than each of the hypotheses.

One important problem for the application of the DS-theory is the efficiency for combining the belief functions from different evidences. Barnett [1981] proposed a polynomial algorithm which only applies to sets of single hypotheses or singletons. Work by ([Shafer and Logan, 1987] and [Shafer *et al.*, 1987]) deals with extended subsets that form a hierarchical structure. More recently, Kennes and Smets [1990] apply fast Möbius transforms to reduce redundant computations and thus improve the efficiency even for the general case.

In this paper, we are concerned with the efficient combination for more than two belief functions. We use as a basis existing methods for combining two belief functions: a direct combination based on Dempster's rule and an indirect combination through Möbius transforms. We further explore parallel algorithms for combining more than two belief functions in order to improve the efficiency, as different pieces of evidence can be combined in any order as long as they are independent of each other.

To further test our algorithms, we consider a medical domain that involves the diagnosis of different types of canine liver diseases (McLeish et al. [1989], [1990], [1991]). This is a domain on which doctors have difficulty predicting precise or single outcomes, as both the numbers of possible outcomes (14) and available tests (40) are quite large. In terms of the DS-theory, this would require a combination of 40 belief functions over a frame of 14 different hypotheses[1]. Although our parallel algorithms can largely speed up the implementation, the amount of time used is still quite long. Fortunately, for practical applications, especially our domain, we found that most of the subsets have zero probabilities; the number of subsets that have non-zero probabilities, called the focal elements, are just about 10 on average. Thus, special versions of our algorithms can be designed to facilitate the practical application. Our algorithms are all implemented on a Sequent machine using the parallel C language and the experimental results are reported later in detail.

2 Review of the DS-theory

In DS-theory, probabilities are assigned directly to subsets of a frame of hypotheses, called a mass function (m). Two pieces of evidences can be combined using the Dempster's rule, where m_1 and m_2 are the mass functions for the given evidences:

$$m(B) = \frac{\sum\{m_1(B_1)m_2(B_2) \mid B_1 \cap B_2 = B\}}{\sum\{m_1(B_1)m_2(B_2) \mid B_1 \cap B_2 \neq \emptyset\}}$$

The rule, as stated in [Buchanan and Shortliffe, 1984], provides a way of narrowing the hypothesis set with the

*This research has been supported by the NSERC Networks of Centers of Excellence Program in Canada.

[1]See [McLeish and Song, 1991] for the general framework of our expert system for diagnosing canine liver diseases

accumulation of evidence and naturally captures the process of diagnostic reasoning in medicine and expert reasoning in general.

There are two ways for combining mass functions proposed in the current literature ([Shafer, 1976] and [Kennes and Smets, 1990]). One is the direct combination based on the Demspter's rule, for which it can be shown that the following theorem holds:

Theorem 2.1 *The direct implementation of the Dempster's rule needs* $(2^n-1)^2$ *additions and* $2^n(2^n-1)$ *multiplications.*

The other way for combining mass functions is the indirect combination through Möbius transforms. Based on a mass function, a commonality function (Q) is further defined in [Shafer, 1976]:

$$Q(A) = \sum\{m(B) \mid B \supseteq A\}$$

With commonality functions, the combination of different evidences is reduced to the multiplication of the commonality functions,

$$Q(A) = KQ_1(A)\ldots Q_n(A)$$

where K^{-1} is a constant that does not depend on A.

A Möbius transform is a function defined over a partially ordered set. For example, the computations from m to Q and vice versa are all Möbius transforms. The idea of a fast Möbius transform is to decompose the whole transform into a series of simple transforms [Kennes and Smets, 1990]. In each step, as illustrated in figure 1, we only consider one hypothesis and its related transform. For example, the first step will achieve the transform: $\{(X,Y) \mid X \neq \emptyset$ and $(Y = X$ or $Y = X \cup \{c\})\}$, where X and Y are two subsets of Θ. Then, by recursively doing this for all the hypotheses, we will be able to transform from one function to another function.

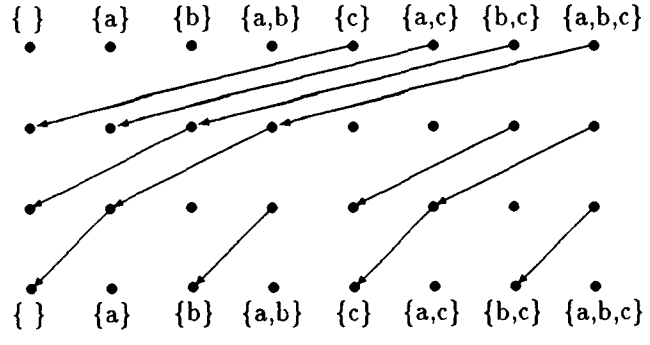


Figure 1: Diagram for the Transform: $m \rightarrow Q$

To combine mass functions, we follow the path from $\{m_i\}$ to $\{Q_i\}$ to Q to m, as shown in figure 2. However, although the transform from Q to m is not provided in [Shafer, 1976], it can be proved, following a similar approach, that the following lemma holds.

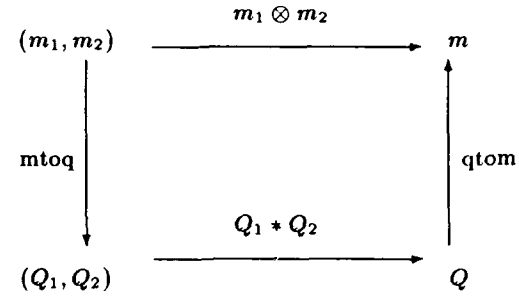


Figure 2: Combination through Möbius Transform

Lemma 2.1 *Suppose* m *and* Q *are two functions defined over a frame* Θ*, then we have:*

$$Q(A) = \sum_{B \supseteq A} m(B) \text{ iff } m(A) = \sum_{B \supseteq A} (-1)^{|B-A|} Q(B).$$

Based on the above lemma, we can now construct a fast Möbius transform from Q to m. It is the same as the transform from m to Q except that all the links have weighting factor (-1) (see [Kennes and Smets, 1990] for detailed discussions).

Theorem 2.2 *The indirect implementation of Dempster's rule through Möbius transforms needs* $3n2^{n-1}$ *additions and* 2^{n+1} *multiplications.*

3 Algorithms for Combining Belief Functions

In this section, we consider how to combine r pieces of evidence efficiently, with $r > 2$. In particular, we present three pairs of algorithms for combining r mass functions: sequential, parallel, and practical methods.

3.1 Sequential Combination Methods

Based on the two methods introduced earlier, we can provide two sequential algorithms for combining more than two belief functions. A sequential algorithm based on Dempster's rule can be given as follows:

algorithm 3.1 sequential & direct implementation
input $m[1:r][0:2^n-1]$, r bodies of mass functions, and n, the cardinality of the frame
output $m[1][0:2^n-1]$, the combined mass function
begin
  for $i = 2$ **step** 1 **until** r **do**
    comb-two($m[1], m[i]$)
  endfor
end

Here, we use a n-digit binary number to represent a frame of size n, and for each subset, the ith element is 1 if the corresponding element is in the subset. Also, "comb-two" is a procedure for combining two mass functions.

Corolary 3.1 *Algorithm 3.1 needs* $(r-1)(2^n-1)^2$ *additions and* $(r-1)2^n(2^n-1)$ *multiplications.*

Another way of implementing the Dempster's rule is to compute the combined mass function indirectly through Möbius transforms. A sequential algorithm for this method can be given as follows:

algorithm 3.2 sequential & indirect implementation
begin
  for $i = 1$ **step** 1 **until** r **do**
    mtoq($m[i]$)
  endfor

  for $i = 0$ **step** 1 **until** $2^n - 1$ **do**
    for $j = 2$ **step** 1 **until** r **do**
      $m[1][i] \leftarrow m[1][i] * m[j][i]$
    endfor
  endfor

  qtom($m[1]$)
end

Corolary 3.2 *Algorithm 3.2 needs* $n(r+1)2^{n-1}$ *additions and* $r2^n$ *multiplications.*

3.2 Parallel Combination Methods

Since in DS-theory, different pieces of evidence can be combined in any order as long as they are independent of each other, we can further explore parallel algorithms for the combination of more than two belief functions.

algorithm 3.3 parallel & direct implementation
begin
  while $r > 1$ **do**
    $r' = r/2$
    for $i = 1$ **step** 1 **until** r' **do in parallel**
      comb-two($m[i], m[r'+i]$)
    endfor
    if odd(r) **then**
      $m[r'+1] = m[r]; r = r'+1$
    else $r = r'$
  endwhile
end

Corolary 3.3 *Algorithm 3.3 needs* $\lceil \log r \rceil (2^n-1)^2$ *additions and* $\lceil \log r \rceil 2^n(2^n-1)$ *multiplications, where* $\lceil \log r \rceil$ *stands for the smallest integer that is greater or equal to* $\log r$.

algorithm 3.4 parallel & indirect implementation
begin
  for $i = 1$ **step** 1 **until** r **do in parallel**
    mtoq($m[i]$)
  endfor

  for $i = 0$ **step** 1 **until** $2^n - 1$ **do in parallel**
    for $j = 2$ **step** 1 **until** r **do**
      $m[1][i] \leftarrow m[1][i] * m[j][i]$
    endfor
  endfor

  qtom($m[1]$)
end

Corolary 3.4 *Algorithm 3.4 needs* $n2^n$ *additions and* $2^n + r$ *multiplications.*

3.3 Practical Combination Methods

To further test our algorithms, we choose a medical domain that involves the diagnosis of canine liver diseases. We found that for such a domain, most of the mass functions only have a small number of non-zero subsets, or focal elements. Although the above algorithms work for general cases, for practical reasons, we must revise them to facilitate the almost null distribution of mass functions.

In the following we first provide a revised procedure for direct combination based on Dempster's rule.

function comb-two'(m_1, m_2, L_1, L_2)
begin
  for $i = 1$ **step** 1 **until** L_1 **do**
    for $j = 1$ **step** 1 **until** L_2 **do**
      $s \leftarrow s_1[i]$ & $s_2[j]$
      $m[s] \leftarrow m[s] + m_1[i] * m_2[j]$
    endfor
  endfor
  $K \leftarrow 1 - m[0]$
  for $i = 1$ **step** 1 **until** $2^n - 1$ **do**
    if $m[i] > 0$ **then**
      $L \leftarrow L + 1$
      $s_1[L] \leftarrow i$; $m_1[L] \leftarrow m[i]/K$
    endif
  endfor
  return L
end

Here, "&" is the bitwise operator for the logical operation "AND", corresponding to the intersection operation between two subsets.

Then a parallel algorithm for combining more than two mass functions can be designed as follows:

algorithm 3.5 practical par. & dir. implementation
begin
  while $r > 1$ **do**
    $r' \leftarrow r/2$
    for $i = 1$ **step** 1 **until** r' **do in parallel**
      $L[i] \leftarrow$ comb-two'($m[i], m[r'+i], L[i], L[r'+i]$)
    endfor
    if odd(r) **then**
      $m[r'+1] \leftarrow m[r]; r \leftarrow r'+1$
    else $r \leftarrow r'$
  endwhile
end

To see how speed can be gained for the above algorithm, let us consider our domain of canine liver diseases. For a frame of size 14, 2^{14} gives us 16,384. Thus, the direct combination of two mass functions would require $(2^{14}-1)^2$ additions and $2^{14}(2^{14}-1)$ multiplications.

However, the above improved direct combination would only need about 100 additions and 110 multiplications, as the average number of focal elements is 10 for any mass functions in our domain (see [McLeish and Song, 1991] for different methods of extracting mass functions from medical data collected over time).

Similarly, we can add a testing statement in a Möbius transform and only perform an addition when the new element is non-zero. Since the cost of a testing statement is usually less than an arithmetic operation, we would expect some saving of time when most of the subsets have zero probabilities. The modified algorithm based on the Möbius transforms will be called algorithm 3.6 in our experiments.

4 Experimental Results

Our algorithms are all implemented on a Sequent Symmetry machine using the Parallel C language [Osterhaug, 1989]. A Sequent machine has an architecture of truly multiple processors and a shared memory, all connected through a system bus. This provides a way for increasing the accessibility of data and minimizing the communication cost. As a result, we can actually run our algorithms on this machine and observe the improvement of speed for a problem of reasonable size.

In our experiments, we run our algorithms on a machine of ten processors. Our results can further be improved when more processors are available, say 16 or 32, which become more and more common for Sequent machines. Although our system is not large, it already shows the potential of using parallel algorithms for efficiently combining belief functions.

# Mass	Alg3.2	Alg3.4	Alg3.5	Alg3.6
02	13.26	9.18	0.43	7.56
03	17.71	9.25	0.95	7.59
04	22.19	9.31	0.95	7.72
05	26.74	9.37	1.51	7.76
08	40.22	13.88	1.49	11.30
10	49.19	14.00	2.04	11.46
15	71.68	18.65	2.35	15.15
16	76.21	23.01	2.34	18.65
20	94.21	23.30	2.88	18.86
25	117.40	27.93	3.52	22.57
30	140.49	32.60	3.54	26.29
32	149.74	37.03	3.86	29.70
35	164.69	37.20	4.39	30.01
40	188.18	41.84	4.39	33.67

Table 1: Results of Sequential and Parallel Experiments

As our results illustrate, for the general case, the parallel implementation based on the fast Möbius transforms (algorithm 3.4) is the most efficient. However, for many real applications where most of the subsets have zero masses, the parallel implementation based on the improved direct combination (algorithm 3.5) is still the most efficient[2].

Further work is being carried out to minimize redundant computations in a Möbius transform and explore parallelism in Dempster's rule. Methods working with continuous data are also being investigated with an application to our domain of liver disease diagnosis.

References

Barnett, J.A. 1981. Computational methods for a mathematical theory of evidence. In *Proceedings of the IJCAI Conference.* 868–875.

Buchanan, B.G. and Shortliffe, E.H. 1984. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley Publishing Company.

Kennes, R. and Smets, P. 1990. Computational aspects of the Möbius transform. In *Proceedings of the Sixth Uncertainty Management Conference.* 344–351.

McLeish, M. and Song, F. 1991. A framework for medical expert systems using Dempster-Shafer theory. Submitted to First World Congress on Expert Systems.

McLeish, M.; Cecile, M.; Yao, P.; and Stirtzinger, T. 1989. Experiments using belief functions and weights of evidence on statistical data and expert opinions. In *Proceedings of the 5th Uncertainty Management Conference.* 253–264.

McLeish, M.; Stirtzinger, T.; and Yao, P. 1990. Using weights of evidence and belief functions in medical diagnosis. In *Proceedings of the AAAI Spring Symposium, AI in Medicine.* 132–136.

McLeish, M.; Yao, P.; and Stirtzinger, T. 1991. Experiments on the use of belief functions for medical expert systems. *Journal of Applied Statistics, Special Issue on Statistics and Expert Systems* 155–174.

Osterhaug, A., editor 1989. *Guide to Parallel Programming on Sequent Computer Systems.* Prentice Hall, second edition.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers.

Shafer, G. and Logan, R. 1987. Implementing Dempster's rule for hierarchical evidence. *Artificial Intelligence* 33:271–298.

Shafer, G.; Shenoy, P.P.; and Mellouli, K. 1987. Propagating belief functions in qualitative Markov trees. *International Journal of Approximate Reasoning* 1:349–400.

Shafer, G. 1976. *A Mathematical Theory of Evidence.* Princeton University Press.

[2]The experiments for algorithms 3.1 and 3.3 are not fully conducted as even for one combination, it already takes about 1.5 hours on our Sequent machine.

A Network Representation of the Multiprocess Dynamic Linear Model

Sharon-Lise Normand
Department of Health Care Policy
Harvard Medical School
25 Shattuck Street, Parcel B - 1st Floor
Boston, MA 02115

David Tritchler
Ontario Cancer Institute
Division of Epidemiology and Statistics
500 Sherbourne Streeet
Toronto, Canada M4X 1K9

Abstract

Dempster [1] has characterized the dynamic linear model (DLM) as a probabilistic belief network, showing that recent algorithms for propagation of information in such networks generalize Kalman filtering, prediction and smoothing algorithms for the DLM. Recently the Bayesian network technology has been extended to model mixed discrete and continuous random variables using conditional Gaussian (CG) distributions [5] with analogous propagation schemes [6]. This paper applies the theory of CG probability networks to characterize the multiprocess dynamic linear model (MPDLM) and its requisite computations in a unified way. The complexity of exact computations is determined and approximate methods are proposed.

1 Introduction

In this paper we apply the theory of conditional Gaussian networks to a class of dynamic linear models that incorporate uncertainty as to the underlying generating model. This class of models has the property that its dependency structure can be modelled graphically. The resulting graph falls under an umbrella of names: a causal probability network, a Bayes belief net, a causal network, a Bayes network, or an influence diagram. Interest centers on the posterior distributions of various sets of random variables. The motivation for using a graphical representation is for computational convenience; the calculations are reduced to a series of efficient local computations. To implement the computations, the graph is transformed into another structure called a *junction tree* [4]. It is in the junction tree that the calculations are performed.

2 Multiprocess Models

2.1 The Dynamic Linear Model

The dynamic linear model [3] is a discrete time linear model that captures a variety of familiar models: regression models, time-dependent covariate models, exponential smoothing models and linear time-series models. The series described by the DLM is a 2-stage hierarchical model with stage 1 of the hierarchy defined by the observation equation and the second stage described by the system equation. The system equation describes how the underlying process that drives the observed series evolves with time t..

$$\begin{aligned} \mathbf{Y}_t &= X_t\beta_t + \epsilon_t \text{ observation equation} \\ \beta_t &= G_t\beta_{t-1} + \mathbf{u}_t \text{ system equation} \end{aligned}$$

where β_t is a $p \times 1$ state vector , G_t is a $p \times p$ known transition matrix, $\mathbf{u}_t$ is a $p \times 1$ vector of system errors, Y_t is a $r \times 1$ observation vector, X_t is a $r \times p$ known regressor matrix, and ϵ_t is a $r \times 1$ vector of observation errors. It is assumed that $\mathbf{u}_t \sim$ independent $N_p(0, V_t)$, $\epsilon_t \sim$ independent $N_r(0, \Sigma_t)$, where V_t and Σ_t are known for all $t > 0$. We also assume for simplicity $\mathbf{u}_t$ and ϵ_t are mutually independent.

Suppose we have prior information available about β_0, say $\beta_0 \sim N_p(\mu_0, S_0)$. Interest centers on infererences for β_t, $t = 1, 2, \cdots$. If we denote the present time by T, then for $t < T$, $t = T$ and $t > T$ the problem becomes one of smoothing, filtering and forecasting respectively. Recursive equations for filtered, smoothed and forecasted estimates are available [7, Pages 216-224],[8].

2.2 An Extension: MPDLMs

The class of multiprocess models [3] reflect uncertainty about the model by formally allowing the model generating the process, henceforth the *generating model*, at any given time to be a random choice from a discrete number of alternative DLMs.

Denote the model at time t with generating model j by $M_t^{(j)}$ for $j = 1, 2, \cdots, N$ where N represents the total number of alternative model choices. Assume that the probability that model j obtains at time t is $\pi_t^{(j)}$. It can be shown that by appropriately characterizing the system and observation variances, a set of DLMs can be constructed that reflect level and trend changes in the series as well as accommodate observation outliers [3].

The estimation method proceeds in the same manner as that for the DLM. However, passing from time t to $t+1$, N^2 posteriors are obtained. This number increases with time, indicating the need for an approximation. One technique [3] approximates the mixture of Gaussian posteriors by a Gaussian with a mixture mean and mixture variance. The mixture mean is calculated by weighting the posterior mean for each generating model by the posterior probability that model $M_t^{(j)}$ obtained at time t and summing over all possible generating models. A similar characterization of the mixture variance holds.

3 Junction Trees

A junction tree [4] J for a graph is a tree whose nodes are the cliques of the graph, and separator sets S_i, associated with the edges, which are the intersections of each clique C_i with its parent. The defining property of a junction tree is that if C_i and C_j have elements in common, all the separators on edges connecting C_i and C_j in J contain those common elements.

Propagation algorithms [4] for computing clique marginals in the junction tree involve only local operations between neighboring cliques. The clique size determines the complexity of the operations. The algorithms lend themselves to object-oriented inplementation and parallel processing. The pattern common to these algorithms is: to propagate information from one clique to another, a marginal for the separator on the link connecting the two cliques is taken in the source clique, and then that marginal multiplies a conditional distribution calculated in the destination clique.

4 Graphical Representation

The causal graph D for the MPDLM is given in Figure 1. The generating model at time t is indicated by the generating variable I_t. Dropping directions and joining parents yields an undirected graph G with cliques of the form $(I_t, \beta_{t-1}, \beta_t), (\beta_t, Y_t), t = 1, 2, \ldots, T + K$, where T is the current time and prediction is K steps ahead. The potential for the first clique is given by the system equation multiplied by the prior for I_t and the potential for the second clique is given by the observation equation. Additionally, the prior for β_0 is a factor in the potential on (I_1, β_0, β_1). The joining of parents has insured that these potentials are defined on the cliques of G. Thus G can form the basis for a junction tree.

4.1 CG-junction trees and the MPDLM

Lauritzen and Wermuth [5] proposed modeling mixed discrete and continuous random variables using CG potentials. Lauritzen [6] gave approximate algorithms based on CG potentials which maintain a CG representation. [6] assumes that the joint distribution be expressed as a product of CG-potentials on the cliques of G and the existence of a junction tree for which each separator S_k satisfies the following additional property:

$$S_k \subseteq \Delta \text{ or } C_k \setminus S_k \subseteq \Gamma \qquad (1)$$

where Δ is the set of discrete variables and Γ is the set of continuous variables. We refer to a junction tree with the above property as a CG-junction tree. The algorithm [6] is approximate in that it employs an operation called weak marginalization when propagating information away from the root of the CG-junction tree. The weak marginal approximates the true marginal by a CG distribution with compatible moment properties.

A junction tree satisfying (1) exists if and only if it represents the clique structure of a graph G' which 1) does not contain any path between two non-adjacent discrete vertices passing through only continuous vertices and 2) is triangulated. The first condition can be satisfied for the MPDLM only if all discrete vertices are adjacent. It follows that there will be a clique of G' containing all $T + K$ discrete variables, and hence a multivariate marginal distribution of very high dimension. After connecting each I_t to all other generating variables we must triangulate the resulting graph by filling in edges to break cycles of length four or greater. We used the method of [10]. The cliques of the triangulated graph are next organized in a junction graph, where cliques with nonempty intersection

are adjacent. Finally, we determine a spanning tree J_{CG} satisfying (1). The root clique is not arbitrary. In fact, the root clique must have CG distribution. Figure 2a gives a CG-junction tree J_{CG} for the MPDLM. In the figure, the rectangles are separators and the round nodes are cliques. We have chosen a fill-in which yields J_{CG} rooted at time T. This has advantages for implementation and also allows the distribution of β_T to be, in theory, exactly known at time t. The updating of clique distibutions to the right of the root is prediction, smoothing is in the left subtree, and conditioning the distribution of the root on y_T is filtering. As the current time is incremented to $T+1$, the tree grows by adding leaves for $T+K+1$ and identifying the root with $T+1$. In general, the root is the only clique whose distribution is known. Only the moment characteristics of other cliques are known.

Although this method allows recursive, local computation, it does not solve the computational problem. The root clique must store the mean and variance for β_T, and a probability, for each cell in the high dimensional table formed by all combinations of the $T+K$ generating variables. For N models, the complexity is of order N^{T+K}. There is no hope of avoiding this with Lauritzen's method, since no matter how we contruct the CG-junction tree, it must include a clique containing all of the state variables.

4.2 An approximate topology

Lauritzen has suggested reducing computations by carrying the idea of weak marginalization further, introducing weak marginalizations when propagating toward the root. This is equivalent to implementing Lauritzen's method in a modified CG-junction tree, illustrated in Figure 2b. Essentially, when we propagate evidence, we 'forget' all but R generating variables.

The junction tree provides a unified computational framework. Filtering, prediction, and smoothing are seen to be the same operation, evidence propagation, in different parts of the tree. The sequence of data collection is arbitrary. At time any time a missing observation can be 'found' and its influence propagated thoughout the tree. We thus generalize all operations in the MPDLM.

The CG-juntion tree for $R=2$ duplicates the filtering calculations of [9]. The power of the network representation is that for any R it also implements prediction, smoothing, and the handling of non-sequential (e.g. missing or delayed) data collection. For filtering, [9] report that the approximation for $R=2$ is adequate, but [2] express dissatisfaction with the approximation. Work needs to be done to investigate the adequacy of the method for different choices of R, for filtering, smoothing, and prediction.

References

[1] A.P. Dempster (1988). Construction and local computation aspects of network belief functions. In "Influence Diagrams, Belief Nets and Decision Analysis", John Wiley and Sons Inc., Ch. 6, pp. 121-141.

[2] K. Gordon and A.F.M. Smith (1990). Modeling and Monitoring of Biomedical Time Series. *Journal of the American Statistical Association*, 85, No. 410, 328-337.

[3] P.J. Harrison and C. F. Stevens (1976). Bayesian Forecasting. *J. R. Statist. Soc. B*, 38, 205-247.

[4] V. Jensen, K. Olesen and S. Andersen (1990). An Algebra of Bayesian Belief Universes for Knowledge Based Systems. *Networks*, To appear.

[5] S. L. Lauritzen and N. Wermuth (1989). Graphical Models for Associations Between Variables, Some of Which are Qualitative and Some Quantitative. *The Annals of Statistics*, 17, No. 1, 31-57.

[6] S. Lauritzen (1990). Propagation of probabilities, means and variances in mixed graphical association models. Res Rep R-90-18, Inst Elec Systems, Aalborg University, Denmark.

[7] J.S. Meditch (1969). *Stochastic Optimal Linear Estimation and Control.* McGraw-Hill.

[8] S.L. Normand and D. Tritchler (1989). Kalman Filtering in a Bayesian Network. *American Statistical Association, 1989 Proceedings of the Statistical Computing Section*, 259-264.

[9] A.F.M. Smith and M. West (1983). Monitoring Renal Transplants: An Application of the Multiprocess Kalman Filter. *Biometrics*, 39, 867-868.

[10] R.E. Tarjan and M. Yannakakis (1984). Simple linear time algoritms to test chordality of graphs, test acyclicity of hypergraphs and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13, 566-579.

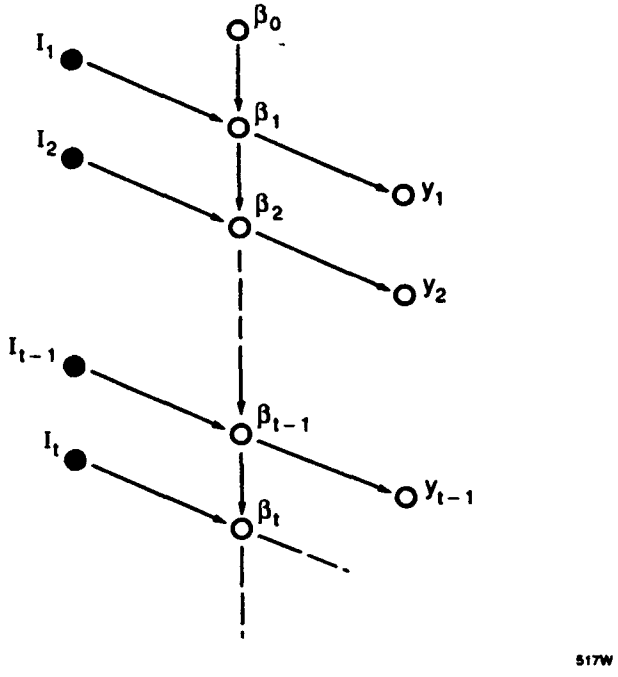


Figure 1. The causal graph for the MPDLM. Continuous variables are circles, discrete variables are dots.

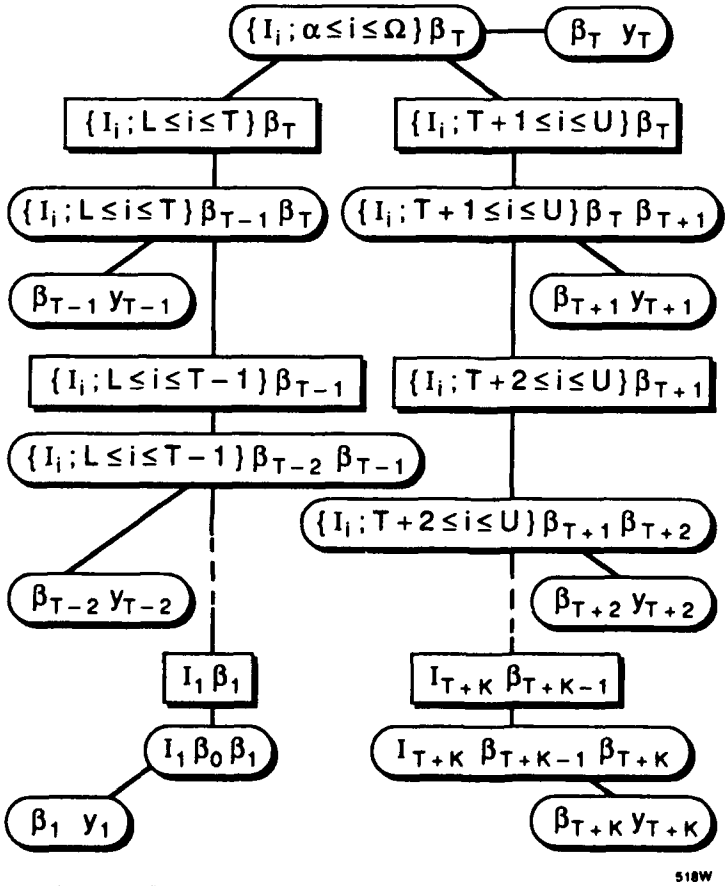


Figure 2. CG – Junction Trees For The MPDLM. a) the full tree is given by $\alpha = L = 1$, $\Omega = U = T + K$ b) an approximate tree with range R is given by $\alpha = T - R + 1$, $\Omega = T + R - 1$, L = upper bound – R + 1, U = lower bound + R – 1; e.g. $L \le 1 \le T$ gives $L = T - R + 1$.

Some Interface Issues for Interactive Statistical Graphics

Catherine Hurley
George Washington University

1 Introduction

Conventional statistics packages such as S [3] or SAS [10], have a limited choice of data structures for representing statistical datasets. In particular, for an operation to be invoked on a dataset or a subset of the data, the data has first to be converted to a specific format, typically 1-d or 2-d arrays. This leads to confusion for the analyst who ends up having to juggle many versions of the same data. In addition the analyst has to remember the connections between the different versions, and connections between analysis results and the original data.

Multiple data versions are especially problematical for interactive statistical graphics. Ideally, a plot acts as graphical interface to the underlying data, allowing all sorts of queries such as requests for information on individuals and variables in the dataset. The plot could then be modified by choosing a new variable to replace one currently appearing in the plot. None of this is possible if a scatterplot (for example) was constructed using two 1-d arrays extracted from the data.

Interactive techniques for linking plots such as painting (brushing) [2,7,8] require that the system be able to determine which point (if any) in one plot 'corresponds' to a point in another plot. Corresponding points typically represent the same dataset individual, so multiple data versions are a nuisance: either the analyst or the system has to remember the connections.

In this note I consider the domain of interactive and dynamic graphics. I describe how such a graphics system need not enforce a particular choice of data representation. By identifying the components of the plot-data interface, I construct an *abstraction barrier* between plot and data. Implementation of the interface relies on *generic functions*, (see, for example Steele [11], or Keene [6]) which may then be specialized for an arbitrary data representation. The need for multiple data versions is reduced by encorporating a general data transformation capability within the plot system.

The statistical graphics system referred to here is part of the forthcoming Zed system [9], and described in Hurley and Oldford [5]. Implementation is in Common Lisp and CLOS [11,6], so examples given here use a small amount of Common Lisp syntax.

2 Plot-data connection

Using examples, the benefits of a general plot-data interface are discussed. A software model for statistical graphics based on a tight coupling of plot and data is outlined.

The following data taken from Andrews and Herzberg [1] will be used as an example. There are 42 apple trees in a designed experiment with 4 treatments and 4 blocks, with 8 qualitative variables measured on the fruit from each tree. Each treatment x block combination initally had 4 trees, but some trees bore no fruit. Figure 1 shows a scatterplot matrix of 4 variables, the lower left plot displays treatment number versus block number for each of the 16 groups, and the plot on the lower right shows mean weight for each of the 16 groups plotted against block number.

2.1 Conventional plot-data interface

At this time, many statistical graphics systems are primarily drawing programs, yielding static plots and supporting little or no interaction (for example, commonly available versions of S [3]). For purposes of illustration, I describe a plot-data interface for such a system.

The convention in statistics packages is to represent data by multi-way arrays. For example, the apple data could be a 2-d array with each row representing a tree. Typically plotting functions require as arguments one or more 1-d arrays (depending on the dimensionality of the plot), so the plot-data interface consists of selecting slices of the data for plotting. Suppose two columns are selected for a scatterplot, then each pair of column entries becomes the subject of a point in the plot.

To construct either of the lower plots in figure 1 requires constructing a new 1-d array whose entries are averages of weight for each treatment x block combination. Other possible plots would use medians instead of averages for example, or weight rather than calcium, but we must first compute new 1-d arrays for these quantites.

The advantage of such a plot-data interface is that it is familiar and relatively easy to work with. The disadvantage is that we must first of all convert the data to arrays, and then construct new arrays for derived variables. As the analysis becomes more involved it is easy to loose track of all the

derived data and their inter-relationshiops.

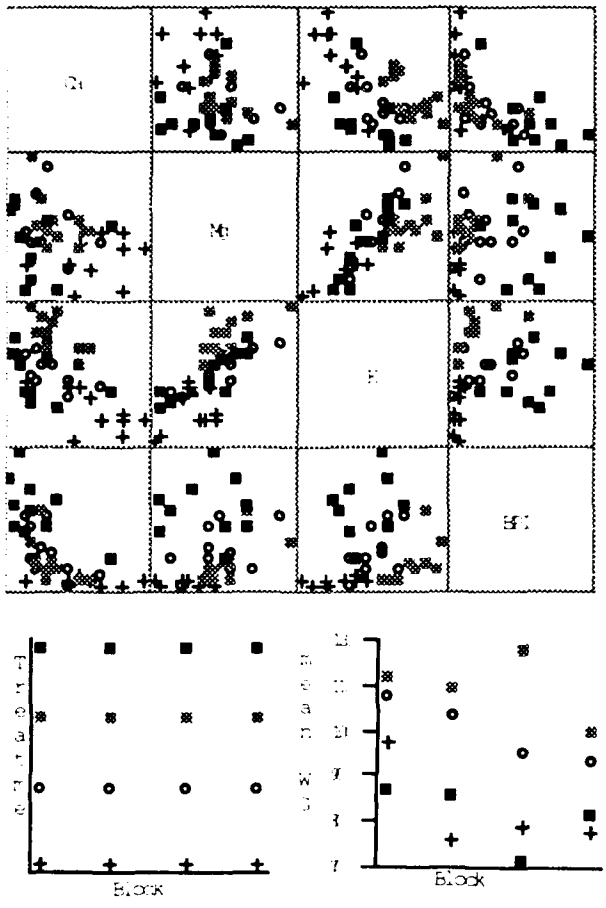


Figure 1

2.2 Interactive graphics

The plots displayed in figure 1 are both interactive and dynamic. An interactive plot is one with which the user can communicate, whilst dynamic plots are plots that can change instantaneously, typically but not necessarily in response to a user action. Of couse, interactive, dynamic plots require a richer plot-data interface.

We use a point and click style interface to retrieve information on the underlying data: for the scatterplot matrix selecting the 'identify' operation on a point gives the name of the tree it represents, whilst in the lower plot 'identify' returns the names of all trees in the group represented by a point. Similarly the 'inspect data' operation returns all information available on the tree (or trees) represented by the selected point.

Changing to another variable provides a simple example of plot modification. The new variable could be present in the dataset, or a derived variable computed using a transformation of existing variables. Buja et al.[4] describe a series of data transformations called a *viewing pipeline* for obtaining plot coordinates from the data, where any pipeline element could be modified resulting in an updated plot. The plot-data interface should accomodate such a viewing pipeline, so plot system rather than the analyst looks after computation of derived variables, and subsequent plot updating.

Linking of plots using interactively modifiable drawing style attributes of points is another common example of dynamic graphics. In figure 1 each of the 4 treatment groups are represented by a different plotting symbol.

In standard implementations of linking [2,12,13], each plot has one point per case, and all points representing a case are required to have the same drawing style. This is also true for points contained in the scatterplot matrix of figure 1. However, in the lower plots each point represents the trees (typically 4) in a particular treatment x block combination. In fact, all three plots are linked. The lower left plot was constructed specifically so that selecting a particular treatment x block combination would be easy: using the painting operation we color the points for the fourth treatment red say, then all points representing trees in this treatment group change to red. Conversely, we could color a single point in one of the scatterplot matrix panels blue, and all points representing the same tree change to blue. This tells us immediately the treatment and block assigned to the selected tree.

More accurately, a point in the lower plot now represents 3 red trees and 1 blue tree, but because a point is small we do not allow proportional coloring. (We do however use proportional coloring for the bars of a histogram or barplot.) One could draw a point using its majority color (red), and use blue for short time following the color change of the linked point. This way we still obtain the subset membership information for the changed point.

2.3 Plot design

In the previous section we noted that many plot modifications in dynamic graphics are characterized by modifying a transformation applied to the data. Also, for linking the system needs to know which point represents which piece of data. Here I outline an organisational scheme for statistical graphics, where the system keeps track of the associations between plot and data. More details are given in Hurley and Oldford [5].

Statistical plots are collections of objects such as points, lines, labels and axes. These objects may be arranged in a hierarchy- a scatterplot consists of axes, label and a pointcloud which itself consists of points. Similarly a scatterplot matrix consists of pointclouds and labels, though arranged in a different format. An object appearing in the plot has an associated piece of statistical data- for the scatterplot it's

the entire dataset, for a point it's typically a case and for the pointcloud the collection of cases. Each component of a plot we term a *view*, so-called because it provides a graphical representation of some piece of data, called the *viewed object*. A view object contains a reference to its viewed object, and an image of a view is used as a graphical interface to the viewed object.

The following discussion relies on such a conceptual model for statistical graphics.

3 A general plot-data interface

In this section, a general plot-data interface is described, without assuming any particular data representation. We will use the scatterplot as an example.

3.1 Plot construction

Suppose we invoke the following function:

```
(scatplot :data apple :x "calcium"
                      :y '(log "wgt"))
```

This builds a scatterplot of the `apple` data with `"calcium"` on the x-axis and `(log "wgt")` on the y-axis. The steps involved are:

1. Construct a scatterplot with the dataset `apple` as its viewed object.
2. The scatterplot then extracts subjects (trees) from the dataset, constructs a pointcloud and two axes each viewing the subjects, constructs title and axis labels, and assigns positions to these views.
3. The pointcloud in turn constructs one point object for each subject, and positions the points by extracting `"calcium"` and `(log "wgt")` from the subjects.
4. The x-axis computes tic marks and tic labels by extracting `"calcium"` coordinates from the subjects, similarly the y-axis.

There are two stages where the plot obtains information from the data:

1. In construction of subjects. The scatterplot by default uses the `list-subjects` function.
2. For obtaining coordinates from the subjects. The pointcloud obtains the x-coordinates by applying the `value-of` function to each subject with `"calcium"` as argument. For the y-coordinate the subjects do not have a variable called `(log "wgt")`; this assumes the `value-of` function will extract `"wgt"` and compute the log.

These steps are easily generalized to arbitrary data representations using *generic functions*. A generic function differs from an ordinary function in that its implementation is distributed across one or more *methods*. When a generic function is invoked, there is an automatic mechanism in place that chooses a method approriate to the arguments, whereupon that method is executed and its values returned, see for example [6,11]. The implementor of the graphics package assumes that methods for the generic functions exist. In order to use the graphics package,the implementor of a dataset should define appropriate methods `list-subjects` and `value-of`.

As demonstrated by figure 1, for a given dataset there are many possible interpretations of subject. In the lower plots a subject is a number of trees instead of just one. Note that any subset of the data may be considered a subject. In general a subject will be a data item (typically a case) or a list of data items. Therefore, as in the following example, we may supply the plot with a function to use to extract the subjects:

```
(scatplot :data apple
          :x 'treat-no
          :y "wgt" :y-function
          :subj-fn 'treat.block)
```

- Subjects are obtained by applying `treat.block` to the dataset, rather than the default `list-subjects`, yielding a list of subjects, where each subject is a list of trees corresponding to a particular treatment x block combination.
- The x-coordinate is obtained by applying the `treat-no` function to the list of trees.
- The y-coordinate is obtained by extracting `"wgt"` from the subjects, then applying the function `mean`.

This assumes functions `treat.block`, `treat-no` and `mean` have been appropriately defined.

The plot system uses additional generic functions to extract (i) the dataset name (used for the plot title), (ii) a list of variables (which are used when constructing a menu for choosing a new variable) (iii) a subject label, and (iv) for inspecting the underlying data. When necessary, the default generic functions may be overidden by providing additional arguments to the view constructors.

3.2 Data transformations

We extend the plot-data interface to include a series of data transformations, so the plot system rather than the analyst looks after computation of derived variables, and subsequent plot updating.

The discussion in the previous section suggests modifying the plot by (i) changing subjects and (ii) changing coordinates.

The subject selection may be modified by deleting or adding in subjects. Most generally, subjects would be modified by supplying a new function to extract them from the dataset. However, this could result in a plot bearing little relation to the original, and so it is just as easy to make a new plot.

There are three steps involved in extracting say the x-coordinates from the data:

1. Extract the subject value (or values). The value extracted is specified by the `:x` argument, which may be any legal argument to `value-of`. In my implementation using a simple dataset representation the `value-of` arguments can be used to extract functions like `(log "wgt")` , or linear combinations `'(+ "carbon" "wgt")`. The `:x` argument may also be a function which is then applied to the subjects.

2. Transform the value from step 1 to a single real number. This transformation is specified by the `:x-function` argument (the default is the identity transformation). The transformation could be log or square root, assuming step 1 returns a single value, or the mean or median function when step 1 returns a list of values. Of course, we could eliminate this step (incorporate it in 1) but this way is simpler for the analyst.

3. Transform the values from step 2 from R^n to R^n, where n is the number of subjects. The transformation is specified by the `:x-transform` argument, again the default is the identity. This step allows for projections of variables, common in linear regression. With transformations defined for the space spanned by a selection of predictor variables and the orthogonal subspace, one immediately obtains residual plots and added-variable plots.

Any of the above arguments may be changed to modify the plot coordinates, either by a command-style interface or selecton from a menu. For 1), the menu offers choice of all variables known to subjects in plot, for 2) the menu offers choices like square root and log, while the menu for step 3 is empty by default. The user can add other choices to the menus for steps 1 and 2, or add transforms to the transformation menus.

3.3 Linking views

Usually we link views in the sense of using common drawing style attributes (color, shape, size) when displaying a data item. In the example of figure 1, points whose viewed objects are identical or have a non-empty intersection are linked. A generic function `eqc` is used to compare a pair of data items to see if their views may be linked. The default just checks for identity. Linking can be used with alternative data representations simply by defining the appropriate method for `eqc`. For instance, if dataset individuals were identified by position or by a label, the `eqc` method could test for identical positions (labels).

References

[1] Andrews, D.F., Herzberg, A (1985) *Data: A Collection of Problems from many Fields for the Student and Research Worker* Springer Verlag, New York.

[2] Becker, R.A, Cleveland, W.S., Wilks, A.R. (1987) Dynamic Graphics for Data Analysys *Statistical Science* 2: 355-395

[3] Becker, R.A. Chambers, J.M, Wilks, A.R. (1988) *The New S Language* Wadsworth and Brooks/Cole.

[4] Buja, A., Asimov, D.A., Hurley, C., McDonald, J.A. (1988) Elements of a Viewing Pipeline for Data Analysis In *Dynamic Graphics for Statistics*, Cleveland, W.S., McGill, M.E. (eds) Wadsworth and Brooks/Cole.

[5] Hurley, C., Oldford, R.W. (1991) A Software Model for Statistical Graphics, In *Computing and Graphics in Statistics*, IMA Volumes in Mathematics and its Applications, vol. 36, Buja, A.,Tukey, P. (eds), Springer-Verlag.

[6] Keene, S.E. (1988) *Object-Oriented Programming in Common Lisp*, Symbolics Press and Addison Wesley.

[7] McDonald, J.A. (1982) Interactive Graphics for Data Analysis, PhD thesis, Stanford University.

[8] Newton, C.M., (1978) Graphics: From Alpha to Omega in Data Analysis, In *Graphical Representation of Multivariate Data*, Wang, P.C.C. (ed), Academic Press.

[9] Oldford, R.W. and others, Zed, undocumented software, University of Waterloo.

[10] SAS Institute Inc.(1985) *SAS User's Guide*, SAS Institute:, Cary, NC.

[11] Steele, G.L. (1990) *Common Lisp, The Language* 2nd ed. Digital Press.

[12] Stuetzle, W. (1987) Plot Windows, *J. Am. Stat. Assoc* 82(398):466-475.

[13] Tierney, L. (1990) *LISP-STAT An Object-Oriented Environment for Statistical Computing and Data Analysis*, Wiley.

An Empirical Evaluation of 3D Spinplots

Richard A. Faldowski, Forrest W. Young & Nada L. Ballator

L.L. Thurstone Psychometric Laboratory, University of North Carolina, Chapel Hill, NC 27599-3270

Abstract

Despite the recent explosion of software development and computer programs capable of bringing dynamic visual data analytic techniques to a wide range of users, little empirical evidence has been offered to justify or support claims about the *potential* usefulness and efficacy of dynamic graphical data analytic procedures as a class. In the current investigation, artificial data with a known three-dimensional "clustery" structure was submitted to a sophisticated data "visualizer" who attempted to identify the structure in the data. Additional variables which were taken into account included (1) the number of true clouds of points present in each data set, (2) the number of data points per each cloud, and (3) the distance between pairs of clouds within a data set. Results indicate that distance between clouds relates positively to the accuracy of cloud membership judgments.

1.0 Purpose

Does the visual exploration of multivariate data using dynamic three-dimensional spinning scatterplots (which we call "3D spinplots") provide a sophisticated user with information and insights about the data he is examining? Do 3D spinplots let an experienced visualizer identify real structure which exists in the data? For virtually anyone who has seen videos of 3D spinplot software such as PRIM-9 (Tukey, Friedman and Fisherkeller, 1973), Dataviewer (Buja & Tukey, 1987) or VISUALS (Young & Reinghans, 1991), the answer would unambiguously be "quite possibly". Unfortunately, a more definitive answer is not available since these methods have not been evaluated with data having known structure.

The current investigation was undertaken to directly address the question of how well and to what degree 3D spinplots permit an experienced user to identify real structure which has been built into sets of artificially generated data. Specifically, the second author, acting as a subject in this experiment, attempted to accurately identify the true cluster-structure of artificial three-dimensional data spaces generated by the first and third authors. Each data space consisted of 1 or more trivariate normal point clouds.

A priori, one might expect the success of a subject in this task to depend on the separation of the true clouds of points, and on the internal compactness of the true clouds of points. Perhaps the greater the separation between two clouds relative to their compactness, the more accurate will be the identification of the correct cluster-structure. Furthermore, clouds containing many data points may be more accurately identified than clouds with few points.

2.0 Design and Methods

Twenty-four trivariate data sets were constructed by the first and third authors. Data sets contained between one and six clouds of data points randomly sampled from trivariate normal distributions with a constant variance-covariance of $\Sigma = \begin{bmatrix} 4.0 & 1.2 & 1.2 \\ 1.2 & 4.0 & 1.2 \\ 1.2 & 1.2 & 4.0 \end{bmatrix}$. The centroid of each cloud within a data set was positioned (within sampling error) at a discrete location (x,y,z) on a three dimensional lattice (where $x \in \{-2.5, 0, 2.5\}$; $y \in \{-2.5, 0, 2.5\}$; & $z \in \{-2.5, 0, 2.5\}$). Sizes of individual clouds within a data set varied to contain either 'Large' (approximately 60), 'Medium' (approximately 30), or 'Small' (approximately 10) numbers of data points. In addition, three data sets contained a supplemental 'Tiny' cloud of 3 data points, one data set contained an 'X-Large' cloud of 120 points, and another an 'XX-Large' cloud of 180 points.

Using the VISUALS (Young & Kent, 1987) data visualization software system on a 22Mhz. 80386-based microcomputer equipped with a 640x480 pixel VGA monitor, the second author was presented with the task of identifying the cluster structure of the data points by classifying them into subjective groups. Initially, the data points appeared as white

dots on a black background with no cues to the true structure of the data clouds apparent. The subject knew only that each data set contained between one and six data clouds and that they were generated by sampling from trivariate normal distributions. He was not informed of the exact number of clouds in a data set, their relative sizes, shapes or locations, or even that the trivariate normal distribution used to generate all clouds had the same variance-covariance structure.

The subject spun the three dimensional visualization space and formed subsets using the VISUALS software system. He worked at his own schedule, dividing his work into 6 sessions, spending about 17 minutes per dataset (range: 2-55 minutes; standard deviation: 12 minutes).

3.0 Data

Figure 1 schematically depicts (in 2-dimensions) a fictitious data set containing five clouds of points designated by the circular regions labelled 'A' through 'E'. The sum of the Arabic numbers within the boundaries of each cloud-circle denote how many data points it contains. Let us assume that these clouds were generated by randomly sampling from trivariate normal distributions with the common variance-covariance (or correlation) matrix specified above. In the figure the subject's classification of the data points into subsets are indicated by the polygonal regions labelled with Roman numerals. The Arabic numbers at the intersections of cloud-circles and subset-polygons specify the number of data points in a particular cloud which the subject classified as members of a particular subset. Thus, the subject classified all 5 data points from cloud 'A' into his subset 'I' along with 5 out of 12 data points from cloud 'D'. The subject's subset 'II' contained 4 data points from cloud 'D', 1 from cloud 'E', and 6 from cloud 'B'. Likewise for the remaining subsets. The information in Figure1 can be summarized by a **p** (number of clouds) by **g** (number of judged subsets) matrix *C*, which cross-tabulates the number of data points in each cloud that were placed into each subjective subset.

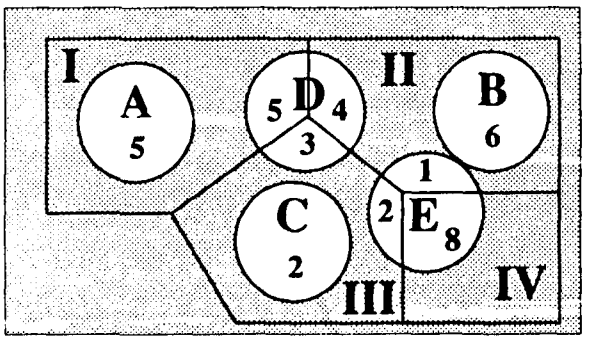


Figure 1

$$C = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{cccc} I & II & III & IV \end{array} \\ \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 5 & 4 & 3 & 0 \\ 0 & 1 & 2 & 8 \end{bmatrix} \end{array} \qquad \text{(EQ 1)}$$

Note that elements of C indicate the frequency with which members from different clouds (rows) were judged to be in the same subset (columns).

4.0 Objective Cloud Measures

Since VISUALS displays objects in a Euclidean space, we use measures based on the relationships between data points and data clouds in Euclidean space. Our measures concern each cloud's **compactness** and the **separation**. Note that these are "objective" measures, since they measure the "true" characteristics of the point clouds. In the next section we define "subjective" measures based on the subject's judgments about cloud characteristics.

Compactness of data cloud **k**, denoted α_k, is defined as the root-mean-squared distance between the points in the cloud and the cloud's centroid. If $\underline{x}_{ik}$ equals the vector of coordinates for the **i**'th data point in the **k**'th data cloud, $\bar{\underline{x}}_k$ equals the vector of coordinates of the centroid of the **k**'th cloud, and n_k equals the number of data points in the **k**'th cloud, then in matrix algebra form:

$$\alpha_k = \sqrt{\frac{1}{n_k}\left[\sum_{i=1}^{n_k} (\underline{x}_{ik} - \bar{\underline{x}}_k)'(\underline{x}_{ik} - \bar{\underline{x}}_k)\right]}. \qquad \text{(EQ 2)}$$

In Figure 1, the α_k for any data cloud is represented by the radius of its cloud-circle.

Separation of data clouds **i** and **j** in the same data set, denoted β_{ij}, is the distance between the cloud centroids:

$$\beta_{ij} = \sqrt{(\bar{\underline{x}}_i - \bar{\underline{x}}_j)'(\bar{\underline{x}}_i - \bar{\underline{x}}_j)} \qquad \text{(EQ 3)}$$

where $\bar{\underline{x}}_i$ and $\bar{\underline{x}}_j$ are defined above. In Figure 1, this is the distance between the centroids of two cloud-circles.

5.0 Subjective Subset Measures

Each of the measures described above reflects "objective" characteristics of the data clouds-- known only because the data were artificially generated. However, in order to fully understand what information the subject took into account as he formed his subsets, we must also develop "subjective" measures paralleling the "objective" measures. These "subjective" measures should reflect the *perceived* characteristics and relationships among the clouds.

Our "subjective" measures use the relative frequency with which *pairs of points* in a judged subset are *correctly or incorrectly grouped together.* Notice that our indices are not based on "*correctly or incorrectly classified points*". This is because as judged by the subject, subsets of points cannot be bound to any prior point classification scheme, making notions of "correct" and "incorrect" classification meaningless. However, after judgments are made, we do know whether two points judged to be in the same subset actually do or do not belong to the same "objective" cloud, allowing us to define correctly or incorrectly co-classified pairs of points.

Even though our measures concern the subjective judgments, they are measures about the objective clouds: They measure how the objective clouds are subjectively perceived. Our two measures are the perceived **cohesiveness** of each cloud (the subjective analog of compactness), and the perceived **distinctiveness** of pairs of clouds (the subjective analog of separation).

Cohesiveness of cloud **k**, denoted a_k, is defined as a function of the square-root of the frequency with which points in one cloud are **correctly** judged to belong with other data points in the same cloud, summed over all subsets, and divided by the maximum possible number of correct co-occurrences of the cloud members. If C *is* the classification matrix described above, and M *is* the diagonal matrix formed from its' row martingales, $M = diag[C1]$, (1 being a column of **g** ones), then (unadjusted) cohesiveness is:

$$\ddot{A} = diag[M^{-1}CC'M^{-1}] \qquad \textbf{(EQ 4)}$$

The diagonal matrix $\ddot{A}$ contains the (unadjusted) cohesiveness $\ddot{a}_{ii}$ of the **i**'th cloud on its diagonal. We call $\ddot{a}_{ii}$ "unadjusted" because its' lower bound, in this study, is

$$c_i = \frac{r(w+1)^2 + (6-r)w^2}{n_i}; \text{ where } \begin{bmatrix} r = Rem[n_i/6] \\ w = Int[n_i/6] \end{bmatrix} \qquad \textbf{(EQ 5)}$$

since the subject knew that data sets contained at most six clouds (the upward bound is 1). Thus, our (adjusted) cohesiveness measure, which ranges between 0 and 1, is:

$$a_i = (\ddot{a}_{ii} - c_i)/(1 - c_i). \qquad \textbf{(EQ 6)}$$

Distinctiveness of a pair of clouds **i** and **j**, denoted b_{ij}, is defined as a function of the (i,j)'th off-diagonal element of the matrix $(M^{-1}CC'M^{-1})$ introduced above. In general, the off-diagonal elements measure the degree to which points from different clouds are **incorrectly** paired with one another, summed over all subsets: They are an index of the degree to which elements from two clouds are "confused" with one another. However, these indices are confounded with the measure of cohesiveness defined above. Thus, we define the matrix $\ddot{M} = [diag(CC')]^{1/2}$. Then an index of confusability of a pair of clouds *corrected* for the cohesiveness of the pair of clouds is given by the off-diagonal elements of $\ddot{M}^{-1}CC'\ddot{M}^{-1}$. Since this confusability measure is inversely related to the distance between two data clouds, we define the **distinctiveness** b_{ij} (which is directly related to distance) as the off-diagonal elements of

$$B = 11' - \ddot{M}^{-1}CC'\ddot{M}^{-1}. \qquad \textbf{(EQ 7)}$$

Note that b_{ij} varies from 0, for complete perceptual confusion of point-clouds **i** and **j**, to 1, for complete perceptual distinction of the two clouds.

6.0 Results

As stated earlier, we expect that the success of the subject in the task posed by our experiment will depend on the separation of the point-clouds and on their compactness.

We expected that the "objective" compactness of data clouds, as measured by α_i, would be positively related to the "subjective" cohesiveness, as measured by a_i. However, the observed correlation between α_i and a_i was essentially zero, indicating that the size of the point-cloud had no effect on the accuracy with which the subject grouped points. This result may be due to the fact that all point-clouds were generated by sampling from populations possessing the same variance-covariance structure.

We also expected that the "objective" separation of pairs of data clouds, as quantified by the separation measure β_{ij}, would be positively related to the "subjective" distinctive-

ness, as measured by the distinctiveness values b_{ij}. The observed correlation between β_{ij} and b_{ij} was .57, indicating that generally, the farther point-clouds were positioned apart, the more accurately the subject grouped points from within them.The scatterplot of the relationship between distinctiveness and separation (Figure 2) reveals a nonlinear relation-

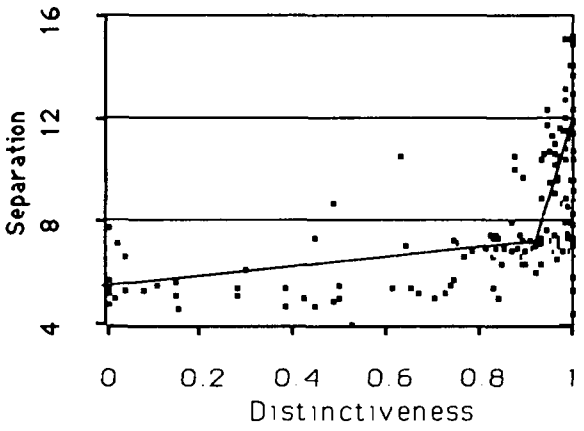


Figure 2

ship over the range of data we have considered. Closer study shows that this nonlinearity results from a trend in which clouds separated by distances greater than 12 were, essentially, perfectly distinguishable; clouds separated by 8 to 12 distance units show distinctivenesses between .85 and 1 (with 2 exceptions); while clouds separated by less than 8 units reveal a very mildly positive relationship. A three-piece linear spline has been drawn by eye for emphasis. (Note that a separation index normalized by compactness correlated .56 with distinctiveness and showed the same scaterplot shape).

Last, we expected that clouds containing many data points would be more accurately identified than clouds containing few points. Both subjective measures, cohesiveness and distinctiveness bear on the question of accuracy. Cohesiveness describes how well data points which belong together are kept together, while distinctiveness describes how well points that belonged apart are kept apart. The correlation between cohesiveness, a_i, and number of data points per cloud equaled .24, suggesting a very modest tendency for large clouds to have more of their observations correctly grouped together than small clouds. This result must be viewed as inconclusive, however, due to the potential correlation induced between cohesiveness and cloud size during the scaling of a_i onto the unit interval. Finally, no relation was seen between distinctiveness and cloud size (using the *average* number of points in the two clouds under consideration), indicating that the average size of pairs of data clouds did not affect the accuracy with which data points from the two clouds were placed into separate subsets.

We also considered the effect of cloud size on the relationship shown in Figure 2. We discovered average cloud sizes of all types (small, medium, and large) represented in all regions of the scatterplot. Thus, no matter how many points they contained, clouds which were close together were not distinguished as well as clouds which were far apart. Furthermore, among all clouds which were far apart, the subject's performance was excellent, across all cloud sizes.

7.0 Discussion

The results of the current investigation are "tantalizing, but preliminary". The intuitively appealing notion of a relationship between the distinctiveness and separation of clouds appears to have been borne out (albeit in a nonlinear fashion). No firm conclusions regarding the relationship between data cloud size and accuracy may be specified; although it would appear that cloud size, if it does have an effect, may exert it on a subject's ability to refrain form incorrectly fragmenting data clouds, rather than on his ability to differentiate between points belonging to two different clouds.

In the future, studies should vary the variance-covariance structure of the populations from which the cloud points are sampled in order to investigate effects on perceived cloud cohesiveness. Second, our results provide rough guidelines for redefining "interesting" ranges of distances to examine in more detail (at least for the variance-covariance structure we used). Third, future studies should contain more heterogeneous mixtures of cloud sizes than we used. Finally, our "subjective" cloud measures belong to a class of covariance-type measures; while our "objective" measures are distances. While comparison between these two classes of measures should correctly point out relationships between "subjective" and "objective" information where such relationships exist, they will not necessarily follow any "nice" functional form (c.f. Figure 2). Consequently, we are actively exploring ways to define distance-like "subjective" measures.

References

Buja, A. & Tukey, P. (1987) Dataviewer: A Program for Looking at Data in Several Dimensions. (Video) Bell Comm. Res., Morristown, NJ.

Tukey, J.W., Friedman, J.H. & Fisherkeller, M.A. (1973) PRIM-9. (Video) Stanford Linear Accelerator Center, Stanford Univ., Stanford, CA.

Young, F.W. & Rheingans, P. (1991) Visualizing Multivariate Data with VISUALS/Pxpl. (Video) Univ. N. Carolina, Psychometrics Lab., Chapel Hill, NC.

Direction and Motion Control in the Grand Tour

Di Cook[*‡], Andreas Buja[*], Javier Cabrera[*]
[*] Bellcore, 445 South St,Morristown, NJ 07962-1910
[‡] Dept of Statistics, Hill Cntr, Busch Campus, Rutgers University, New Brunswick, NJ 08904
dcook@stat.rutgers.edu

1 Abstract

Exploring multivariate data with the grand tour[1] is a visually exciting way to discover interesting structure. However, one criticism of this method is that as dimensionality increases the chances of quickly discovering views of interest diminish rapidly, because of the random nature of the grand tour, and the expanding volume of space.

To improve the chances of discovering interesting structure we propose a method for controlling the exploration by motion control and directing movement along the gradient of a projection pursuit function.

The benefits of this approach are two-fold. Firstly, it provides a fast, powerful exploratory data analysis tool, and secondly, it provides a vehicle for exploring and comparing projection pursuit functions.

2 Introduction

Suppose that we are in a two-dimensional world in a higher-dimensional universe, and suppose that despite this handicap we are interested in exploring our universe via sequences of two-dimensional views. Essentially this is the environment of the grand tour.

Our implementation of the grand tour, coined the random jump walk tour, is one in which a starting plane is fixed, an ending plane is generated randomly in the p-dimensional data space, and the jump walk tour path is the geodesic interpolation between the two. When the ending plane is reached it becomes the new starting plane, a new ending plane is randomly generated, and the tour progresses, essentially randomly walking on a Grassmann manifold in p-space[2].

To begin providing control in the tour we first look at controlling the jump size. We define the jump size to be the distance between the starting and ending planes, that is, the norm of the canonical angles. In a random jump walk tour this jump size fluctuates randomly, in an unrestricted manner. If we provide control over the jump size, by restricting it to be small, we keep the exploration local, whilst increasing the jump size allows more global movement over the space.

In addition we assign an index of *interest* to each two-dimensional view of the data. The meaning of interest varies somewhat in data analysis. For example, Friedman and Tukey[3] thought local clumpiness was interesting but later Friedman[4] associated interest with non-normality.

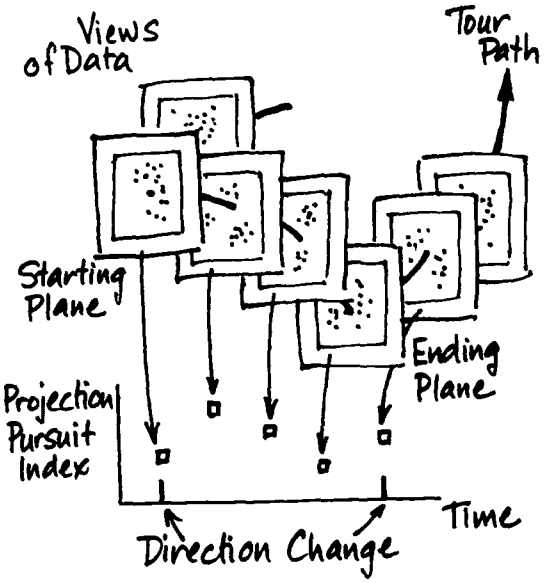


Figure 1: *Jump random walk tour with indices of interest plotted over time.*

A projection pursuit function is used to assign an index of interest to each two-dimensional projection of the data. By choosing a smooth function the derivative can be used to determine the new motion direction, as opposed to randomly generating directions in the unrestricted tour.

Combining this with jump size control means that we try to direct the tour towards views with higher indices of interest, and thus hopefully, views that expose the structure in the data.

3 Projection Pursuit Indices

There are four indices of two basic types which we have currently implemented; two indices based on expansions, and two indices based on density estimates. For two-dimensional projection pursuit it is usual to initially sphere the data, either by principal components

or using a robust variance-covariance matrix. Here the following notation is used:

z_i = data vector; $i = 1, ..., n$
α, β = projection vectors
$x_{1i} = \alpha' z_i, \quad x_{2i} = \beta' z_i$
$y_{1i} = 2\Phi(x_{1i}) - 1, \quad y_{2i} = 2\Phi(x_{2i}) - 1$

3.1 Polynomial Indices

Friedman's[4] index is the L_2-distance between the function g, obtained by inverting the empirical density through a standard normal cdf, and a bivariate uniform density on $[-1,1] \times [-1,1]$. The empirical density expanded in moments by Legendre polynomials as follows:

$$\begin{aligned} I(\alpha,\beta) &= \int_{-1}^{1}\int_{-1}^{1}\{g(y_1,y_2) - \frac{1}{2}\}^2 dy_1 dy_2 \\ &= \frac{1}{4}\sum_{j=0}^{\infty}\sum_{k=0}^{\infty}(2j+1)(2k+1) \times \\ &\qquad E^2\{P_j(y_1)P_k(y_2)\} - \frac{1}{4} \end{aligned}$$

where

$P_0(y) = 1, \quad P_1(y) = y,$
$P_j(y) = \frac{1}{j}[(2j-1)yP_{j-1}(y) - (j-1)P_{j-2}(y)]$

are the Legendre polynomials.

In response, Hall[5] suggested that Friedman's index is not useful for heavy tailed distributions, by showing that this index will be infinite if the tails of the distribution do not decrease at least as fast as $\exp\{-x^2/4\}$. As an alternative he proposed an index based on the L_2 distance of the empirical density, g, from a standard normal density, with the expansion of the empirical density obtained by Hermite polynomials. Our bivariate version of this approach is as follows:

$$\begin{aligned} I(\alpha,\beta) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\{g(x_1,x_2) - \phi(x_1,x_2)\}^2 dx_1 dx_2 \\ &= \sum_{j=0}^{\infty}\sum_{k=0}^{\infty}E^2\{h_j(x_1)h_k(x_2)\} \\ &\qquad -2\pi^{-\frac{1}{2}}E\{h_0(x_1)h_0(x_2)\} + \pi^{-1} \end{aligned}$$

where

$h_i(x) = (i!)^{-\frac{1}{2}}\pi^{\frac{1}{4}}H_i(x)\phi(x)$

and

$H_0(x) = 1, \quad H_1(x) = x$
$H_i(x) = xH_{i-1}(x) - (i-1)H_{i-2}(x)$

are the standardized Hermite polynomials[11].

Both of these indices are estimated by truncating the sum at some finite number (Friedman[4] suggests between 4 and 8 for his index), and estimating the expected values by sample means.

3.2 Density Estimate Indices

Friedman and Tukey[3] originally proposed an index based on a local scale measure multiplied by a local density estimate designed to search for clumpiness in the data. We have implemented a bivariate adaptation of this index based on the L_2-norm of a local density estimate. Because we initially sphered the data, we disregard the local scale measure, and have:

$$I(\alpha,\beta) = \int f^2(\mathbf{x})d\mathbf{x}$$

where f is estimated by a kernel density estimate

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^2}\sum_{i=1}^{n}K\{\frac{1}{h}(\mathbf{x} - \mathbf{x}_i)\}$$

with

$$K(\mathbf{x}) = \begin{cases} \frac{4}{\pi}(1 - \mathbf{x}'\mathbf{x})^3 & \text{if } \mathbf{x}'\mathbf{x}<1 \\ 0 & \text{otherwise} \end{cases}$$

The kernel is one that is proposed by Silverman[9] for bivariate density estimation, because of its differentiability properties. Normally in calculating a density estimate one would optimize the window width parameter, however this would be impractical to do for each two-dimensional projection, so we set a value and allow the user interactive control of this.

The fourth index we consider is negative entropy which is very similar to the Friedman-Tukey index, in that it is based on a local density estimate, as above:

$$I(\alpha,\beta) = -\int f(\mathbf{x})\log f(\mathbf{x})d\mathbf{x}$$

This index is discussed by Jones and Sibson[7], and Huber[6].

4 Implementation

The implementation of projection pursuit is embedded into XGobi, the dynamic graphics program under development by Swayne, Cook and Buja[10]. Figure 2 gives an indication of the setup.

An XGobi window is initiated with the data of interest. **Tour** mode is activated. The top plot window has the data dynamically touring. When **ProjPursuit** is selected the bottom window pops up. In this window the projection index is plotted over time. The current value of the index is also printed in a small window beside the projection pursuit button.

In addition the data is sphered by principal components, as indicated by the **PrnCmp Basis** button being highlighted, and the variable labels become PC1, PC2, etc.

Projection pursuit can be either **Active** or **Passive**. In active mode the direction of movement is determined by the derivatives of the projection pursuit function whilst in passive mode the tour reverts to the random jump walk but we still get the indices plotted over time.

Beside the **Active** button is one labelled **Bitmap**. Clicking on this generates a small picture in the bottom window of the view in the top window at the time. As

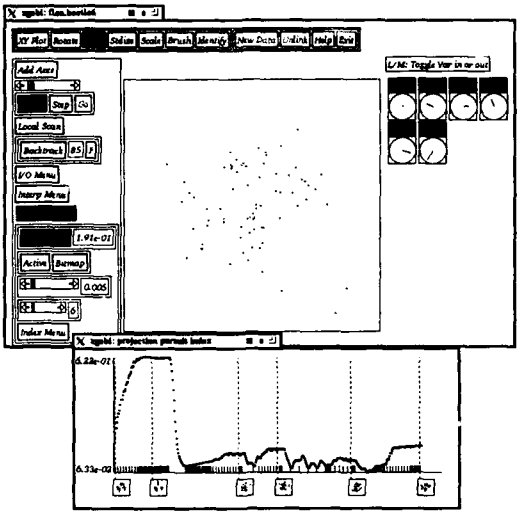


Figure 2: *Implementation of direction and motion control*

we can see the view corresponding to the first local maximum is one in which three distinct clusters within the data are separated.

The scrollbar below the active allows the user to control the tour jump size during active projection pursuit. The effects of this are seen in the bottom window. Every time a direction change is made a small vertical bar is drawn on the horizontal axis. When the jump size is small the tour sharpens up the nearest local maximum and when large the tour moves globally over the Grassmann manifold in p-space.

The next scrollbar controls the number of terms in the expansion of the polynomial indices, and also allows the window width to be adjusted for the density estimates.

Lastly there is a menu for selecting a projection pursuit index.

5 Examples

In this section we are looking at the uses of this methodology in exploring data, and comparing projection indices. For this purpose we show window dumps in figures 3, 4 and 5 of the bottom time series window of the progress of the tour guided by projection pursuit over time. Every time a new gradient is calculated a bar is drawn on the horizontal axis. Keep in mind that in the setup on a workstation this is happening dynamically, and we see the data touring simultaneously in the top window.

The data in figures 3 and 4 comes from Lubischew[8]. It consists of 6 measurements on 3 species of flea-beetles, with a total of 74 cases. The window dumps in figure 3 are brief sessions with Friedman's index and Hall's index, respectively. Figure 4 has window dumps of the Friedman-Tukey and entropy indices.

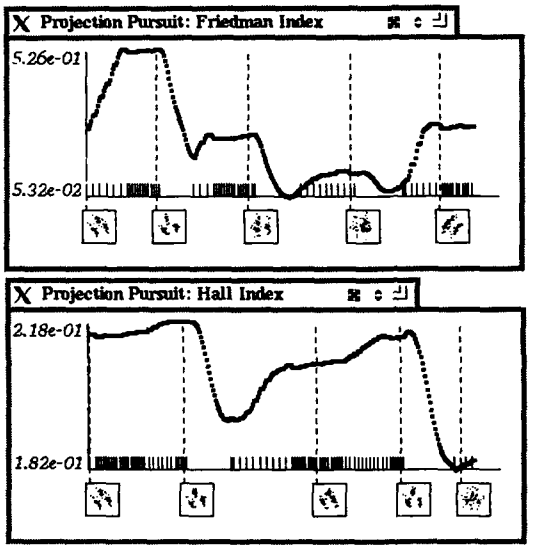


Figure 3: *Comparison of the polynomial indices on flea-beetle data.*

In each case the starting point is the view given by the first two principal components. We see that this is not a very discriminating view of the three species. However upon activating projection pursuit guidance we see that all the indices immediately move the tour into a view of a three group separation, so with very little work we have discovered a very informative picture of the data.

In figure 5, a comparison of the polynomial indices is illustrated on a nine-dimensional hypercube. The views that distinguish the hypercube based on two-dimensional projections are ones where the data collapse into the four vertices of a square. During an unrestrained random jump walk tour, most views of the hypercube appear close to being bivariate normal, aside from the interference patterns. It is virtually impossible to see a complete collapse into the four point view.

So it is interesting that starting at an arbitrary view, in this data, the projection pursuit directed tour very quickly finds a view of the data collapsed into a square, for both of the two polynomial indices. The one difference between the two indices is that Hall's index seems to need to do less work to find this four point view, whilst Friedman's index needs to wend its way through some lower level maxima. (The apparent planar non-equivariance of Hall's index is due to the truncation of the infinite sum.)

6 Conclusions

We have found that the polynomial indices show the most promise cue to their speed of computation. The computation of these indices is of order n, as opposed to order n^2 for the density estimate indices. In practice,

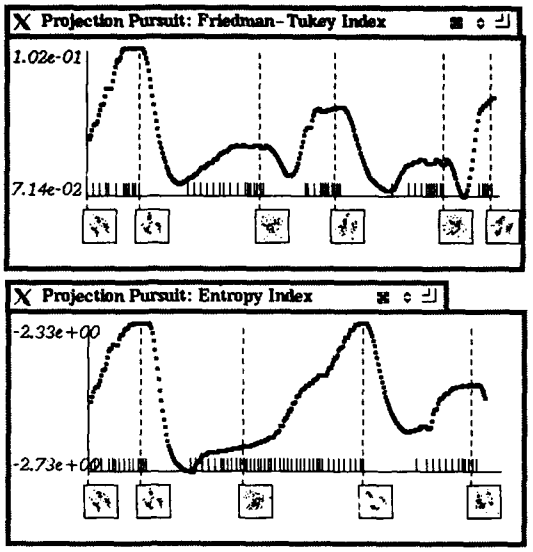


Figure 4: *Comparison of the density estimation indices on flea-beetle data.*

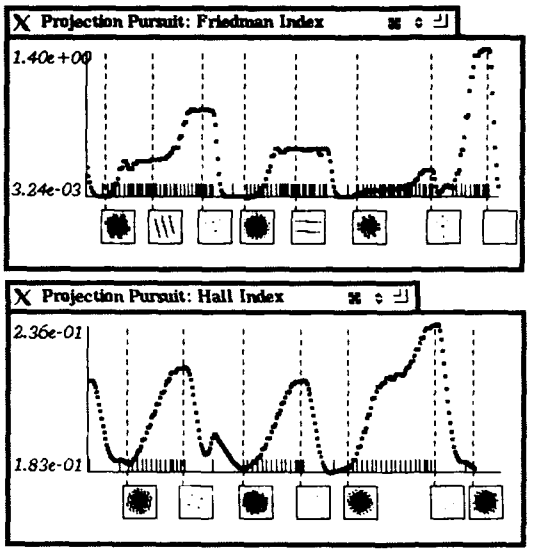


Figure 5: *Comparison of polynomial indices on nine-dimensional hypercube.*

Friedman's index doesn't appear to be overly sensitive to outliers and heavy tails as Hall suggested, but this doesn't mean that both indices behave identically, as we saw in the last section. However, other than non-normality it is difficult to quantify what it is that the indices are actually searching for.

Quite readily, beginning at the view given by the first two principal components, our guidance system finds interesting structure, but the inherent optimization problems with noisy functions arise. Using simple derivative-based direction control doesn't assist in finding tight local peaks and creates problems when the function consists of long trenches or ridges. These structures tend to be more common in as dimensionality increases. We counter some of the problems by switching to passive mode and allowing the random jump walk tour to move over the space before beginning active projection pursuit again.

Whilst it is simple to sphere the data by principal components, it is not ideal, and so our next question will be to explore these methods with robust sphering.

Despite the problems we have encountered, we have devised a tool which readily allows a comparison and development of indices, as well as providing direction and motion control in the grand tour to increase the chances of discovering structure when exploring data.

7 References

[1] Asimov, D. (1985) The Grand Tour: A tool for viewing multidimensional data. *SIAM J. Sci. and Statist. Comput.* **6** 128-143.

[2] Buja, A., Asimov, D. and Hurley, C. (1989) Methods for Subspace Interpolation in Dynamic Graphics. *Bellcore Technical Memorandum.*

[3] Friedman, J. H. and Tukey, J. W. (1974) A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Trans. Comput. C* **23** 881-889.

[4] Friedman, J. H. (1987) Exploratory Projection Pursuit. *J. Amer. Statist. Assoc.* **82** 249-266.

[5] Hall, P. (1989) Polynomial Projection Pursuit. *Ann. Statist.* **17** 589-605.

[6] Huber, P. J. (1985) Projection Pursuit (with discussion). *Ann. Statist.* **13** 435-525.

[7] Jones, M. C. and Sibson, R. (1987) What is projection pursuit? (with discussion) *J. Roy. Statist. Soc. Ser. A* **150** 1-36.

[8] Lubischew, A. A. (1962) On the Use of Discriminant Functions in Taxonomy. *Biometrics* **18** 455-477.

[9] Silverman, B. W. (1986) *Density Estimation.* Chapman and Hall, London.

[10] Swayne, D. F., Cook, D. (1990) XGobi: A Dynamic Graphics Program Implemented in X with a Link to S. *Proc. of the 22nd Symp. on the Interface between Comput. Sci. and Statist.*, Springer-Verlag, New York.

[11] Thisted, R. A. (1988) *Elements of Statistical Computing.* Chapman and Hall, New York.

8 Acknowledgements

We would like to thank our colleagues at Bellcore and Rutgers, particularly Debby Swayne and Martin Maechler, for many useful and informative discussions. We would also like to acknowledge unpublished work in the area done by Catherine Hurley.

A SYSTEM-INDEPENDENT GRAPHICAL USER INTERFACE FOR THE SCA STATISTICAL SYSTEM

Lon-Mu Liu, Alan Montgomery, Ki-Kan Chan
Department of Information and Decision Sciences
University of Illinois at Chicago (M/C 294)
Box 4348, Chicago, Illinois 60680

Abstract

In this paper we develop an approach for creating a graphical user interface (GUI) for an existing command driven mainframe program. As an example of this approach, we present an implementation using the SCA Statistical System. The windows front-end to the SCA System runs on a personal computer using Microsoft Windows. The front-end communicates with the SCA System running on a mainframe or workstation through a serial communication device. This implementation demonstrates the advantage of using such an approach.

1. Introduction

A general goal in the computer industry has been towards making computers and software more user friendly. A recent trend towards reaching this goal is through the use of a graphical user interface (GUI). A common problem with GUIs is that they either favor novice users over experts, or favor expert users over novices. In this paper, we present an approach for combining command and graphical user interfaces into a single user interface that benefits both novice and expert users. We will refer to the user interface presented in this paper as the composite user interface (CUI).

We have implemented this approach for the SCA Statistical System and later we will discuss certain features that have resulted from our design approach. The SCA Statistical System consists of three packages which provide capabilities for forecasting and time series analysis (Liu et al. 1986), quality and producitivity improvement using statistical methods (Liu et al. 1987), and general statistical analysis (Hudak et al. 1989). The graphical front-end to the SCA System runs on a personal computer using Microsoft Windows. The front-end communicates with the SCA System running on a host computer through a serial communication device. When the mainframe SCA System and the SCA Windows/Graphics Package (Liu et al. 1991) are used together, a complete windowing environment is created for the user without any modification to the existing SCA System.

2. Design and Philosophy of the Composite User Interface

A primary goal of our user interface design is to allow graphical and command user interfaces to coexist in the same application software. In addition, we want the user interface portion of the software to be as independent of the computational portion of the software as possible, and ideally the same user interface program is able to function with different versions of the computational portion of the program for different computers and operating systems. These are the key emphases of "system-independence" in our user interface design. In trying to fulfill these goals we have developed a hybrid user interface, that we refer to as the composite user interface. Below we outline the basic features that comprise the composite user interface presented in this paper:

1. The functionality of the program is independent from the user interface. To facilitate this separation, we use two separate programs: a computational program and a front-end program. We will refer to this separation of the front-end program from the computational program as the **segmented** feature of the interface.
2. The computational program provides a command user interface which is the same across different host operating systems.
3. The front-end program provides a graphical user interface which conforms to a native GUI environment. The user accesses the computational program through the use of dialog boxes, and other GUI devices. These graphical objects will generate syntactically correct commands for the computational program.
4. The front-end program should also provide a command window which preserves a command user interface to the computational program and bypass the graphical objects of the front-end

program. We will refer to the support of graphical and command user interfaces in one program as the **dual** feature of the interface.

5. The user should be able to customize the environment and use both graphical objects and commands interchangeably in the same session.

The primary purpose of the composite user interface is to support an interface in which graphical and command user interfaces can be integrated. Such a dual feature is very desirable. We envision that software development is an evolutionary process, and the extension of graphical user interfaces is part of this process. Since the current base of users have already made an extensive investment in the current command language, we see the need to preserve the command language as it currently exists.

The need for a segmented approach is also driven by the requirements of system-independent, modularity, and portability. The most plausible approach to implement this feature is to develop the front-end interface as a separate program.

The approach outlined above will result in more portable code, since the functionality of the program does not depend on the GUI. Also since command programs are usually less machine dependent, it should be fairly easy to move the command program to new environments. In addition, if the new environment does not support a GUI, the command program is still a viable program in its own right.

3. An Implementation Using the SCA System

In general, the approach outlined in Section 2 can be applied to any existing command program. In this section, we illustrate this approach using the SCA Statistical System. In this implementation, the front-end program runs on an IBM compatible personal computer running Microsoft Windows. We have tested the front-end program with the SCA System which runs on IBM/TSO, IBM/CMS, VAX/VMS, or UNIX operating systems. This software is currently available from SCA as the SCA Windows/Graphics Package (Liu et al. 1991), which we will also refer to as the SCAWIN program in this paper.

3.1 The SCA Statistical System

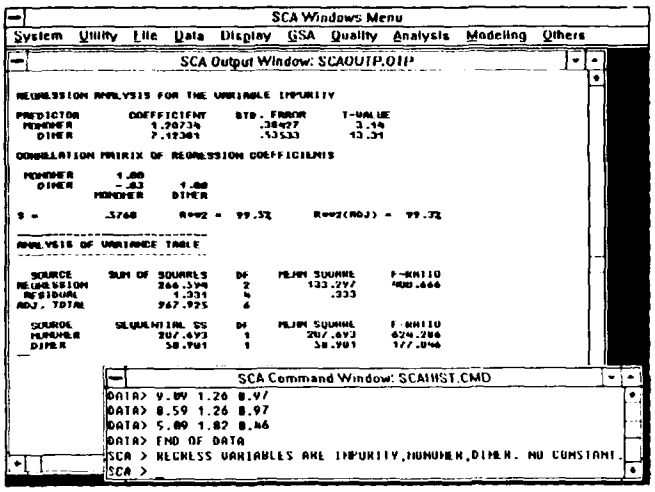
Even though the proposed approach theoretically can be employed in any command driven software, its effectiveness and implementation depends on the command structure of the software. If the software has a rather simple command structure, the GUI will be of less benefit. The SCA System has a fairly extensive syntactical structure, which includes modifiers to each command. This makes a GUI extremely beneficial to users who are not familiar with all the features of the SCA System. In this section, we employ a set of data from Box, Hunter, and Hunter (1978) to illustrate the command user interface of the SCA System. The data employed were the results of a chemical experiment. In this experiment, it was believed the initial rate of the formation of a chemical impurity causing a discoloration is linearly depedent on the concentrations of monomer and dimer. The rate is zero when both components are absent. The data are entered directly and stored in the SCA workspace in the variables IMPURITY, MONOMTER, and DIMER. A regression analysis with zero intercept (i.e. no constant term) is then performed. The SCA commands to perform the above analysis is listed below:

```
INPUT  VARIABLES ARE IMPURITY, MONOMER, DIMER.
5.75  0.34  0.73
4.79  0.34  0.73
5.44  0.58  0.69
9.09  1.26  0.97
8.59  1.26  0.97
5.09  1.82  0.46
END OF DATA
REGRESS  VARIABLES ARE IMPURITY, MONOMER, DIMER. @
         NO CONSTANT.
STOP
```

In the above SCA session, we have executed three SCA commands: INPUT, REGRESS, and STOP. The function and syntax of these commands are illustrative of SCA command syntax. An SCA command is also referred to as a **paragraph**. The first word of the command is called the **paragraph name**. In this example, INPUT, REGRESS, and STOP are paragraph names. The paragraph name is followed by various modifiers to the command, the modifiers are referred to as **sentences**. In the REGRESS paragraph there are two sentences: "VARIABLES ..." and "NO CONSTANT". Notice that sentences are separated by the delimiter period (".").

3.2 The SCA Windows/Graphics Program

Here we outline the features of the SCAWIN program. To start the SCAWIN program, the user must first login to the host computer using the terminal emulator window included with the SCAWIN program. After the user executes the SCA System, the user may enter the SCA commands and data (shown in Section 3.1) into the Command Window. Below is a sample screen for the analysis discussed in the above section.



The above display contains three windows. They are the **SCA Windows Menu**, the **SCA Output Window**, and the **SCA Command Window**.

(A) Command Window

The composite user interface described in this paper requires that a user be able to access a command program in its native language. The SCA Command Window allows the user to communicate with the SCA System in this manner. The Command Window also maintains a complete history of all SCA commands issued during an SCA session. The commands in this window can be edited and then executed.

(B) Output Window

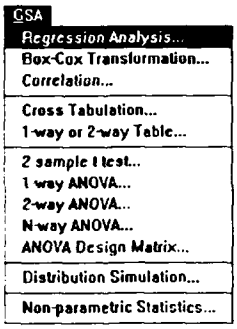
The Output Window contains all the output from an SCA session. All information (i.e. text) in the Output Window can be reviewed at any time.

(C) SCA Windows Menu

One of the most important features in the SCAWIN program is the ability to access the command language using graphical objects. To accomplish this, we have implemented pull-down menus to allow users to access all SCA paragraphs. The SCA Windows Menu, shown below, allows the user to create SCA commands through dialog boxes and also display help information for SCA paragraphs.

Ten menu items are displayed on the SCA menu bar. Each of these items activate a pull-down menu.

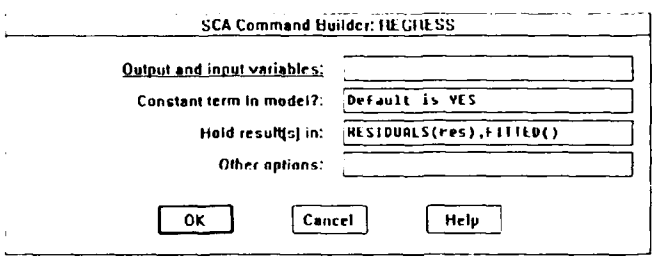
When the user selects a menu, a pull-down menu is generated. For example suppose a user wishes to perform a regression analysis, the user would select the "GSA" item on the SCA Windows Menu and then select the "Regression Analysis..." item.



By selecting this item, a dialog box is displayed to assist the user to create a command. In the SCAWIN program, this dialog box is referred to as the Command Builder. Below we describe the use of the Command Builder.

(D) Command Builder Window

The Command Builder Window is designed to facilitate the construction and entry of any SCA paragraphs. As an illustration, we show the short Command Builder Window for the REGRESS paragraph:

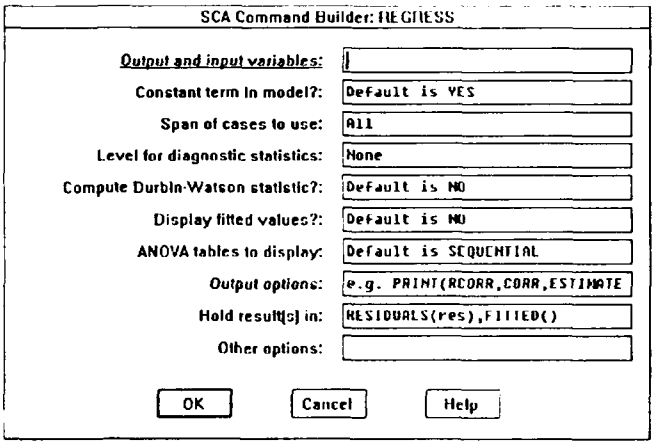


To create an SCA command, the user first enters information into one or more of the controls. (In this instance, a control refers to one of the text input boxes in the above dialog box.) Even though many controls may be displayed, the user does not need to enter information for all controls. Information only needs to be provided for those controls that correspond to required SCA sentences (required sentences were discussed in Section 3.1 and are signified by having the prompt underlined and in red color). When the user has completed entering the information, he may select the "OK" button or presses the **Enter** key. The command will then be created and

sent to the SCA System. The SCA command created is also displayed in the Command Window.

The Command Builder does not include instructions for every sentence when it creates an SCA command. Several keywords have been employed to indicate those sentences that need not be processed. These keywords are "None", "All", "Default", and "e.g.". If an argument appearing in the control begins with one of these keywords, then the Command Builder will use the SCA System's default argument for that sentence. The keywords defined above have been employed to provide the user with the default options or serve as sample information.

Another feature that has been implemented is the ability for the user to control the amount of options presented in the Command Builder Window. If the user requests a full Command Builder, then more optional sentences will be provided for the REGRESS paragraph as shown below:



(E) High Resolution Graphics

The mainframe SCA System does not have high resolution graphics capabilities of its own, this is due to the highly machine dependent requirements of high resolution graphics. Using the composite user interface, we are able to implement graphics without having to change the existing SCA program. This is achieved by capturing the data on the PC and then displaying the graphics in the Graphics Window.

4. Summary and Conclusion

In this paper we have outlined a composite user interface for creating a GUI for an existing command program, without having to modify the existing command program. The same front-end GUI program will work with the command program under different computers or operating systems. We have also stressed the need of the co-existence of command and graphical user interfaces. By using an interface with a dual feature, we have demonstrated an approach that will have benefits to different levels of users.

The approach presented in this paper not only provides benefits to users, but also to software developers. Software developers do not need to rework their existing programs, but instead can concentrate on the GUI front-end. By retaining the native command language and minimizing the changes to the existing code, costs for documentation and future software maintenance are reduced (Boehm and Papaccio 1988). The composite user interface has demonstrated to be a cost effective means for both users and software developers for migrating from character-oriented to graphical environments.

REFERENCES

Boehm, B.W. and P.N. Papaccio (1988). "Understanding and Controlling Software Costs", *IEEE Transactions on Software Engineering, 14(10)*, 1462-1477.

Box, G.E.P., W.G. Hunter, and J.S. Hunter (1978). *Statistic for Experimenters*. Wiley, New York.

Hudak, G.B., L.-M., Liu, G.E.P. Box, M.E. Muller, and G.C.Tiao (1989). *The SCA Statistical System: Reference Manual for General Statistical Analysis*. Scientific Computing Associates, P.O. Box 625, DeKalb, IL 60115.

Liu, L.-M., G.B. Hudak, G.E.P. Box, M.E. Muller, and G.C. Tiao (1986). *The SCA Statistical System: Reference Manual for Forecasting and Time Series Analysis*. Scientific Computing Associates, P.O. Box 625, DeKalb, IL 60115.

Liu, L.-M., G.B. Hudak, and G.E.P. Box (1987). *Quality and Productivity Improvement using the SCA Statistical System*. Scientific Computing Associates, P.O. Box 625, DeKalb, IL 60115.

Liu, L.-M., M.E. Muller, K.-K. Chan, A.L. Montgomery, W.-G. Zhang, and G.B. Hudak (1991). *SCA Windows/Graphics Package User's Guide: Windows and Graphics Capabilities of the SCA System for Mainframes and Workstations*. Scientific Computing Associates, P.O. Box 625, DeKalb, IL 60115.

A Bivariate, Nonstationary Time-Series Model for Global Fossil Fuel Production

Bert W. Rust and Frank J. Crosby
Center for Computing and Applied Mathematics
National Institute of Standards and Technology
Gaithersburg, MD 20899

Mankind is returning fossil fuel generated CO_2 to Earth's atmosphere at an exponential rate, causing concern about a greenhouse warming. Jones, et.al. (1986) derived the record of yearly average temperature changes plotted in Fig. 1. The least squares straight line has slope $0.38 \pm 0.04\,(^\circ C)(\text{century})^{-1}$, but the average slope since 1970 has been much greater and is thought by some to indicate the onset of the greenhouse.

In Fig. 2 the circles represent annual global totals of fossil fuel production for 1870-1986 [Boden (1988)]. The dashed curve is a nonlinear least squares fit of the model

$$\frac{dP}{dt} = \alpha P \ , \quad P(0) = P_0 \implies P(t) = P_0 \exp(\alpha t) \ .$$

The fitting parameters together with the sum of squared residuals (SSR) are given in the first row of the table on the next page.

Rust and Kirk (1982) showed that for 1870-1974, the exponential growth of fossil fuel production was modulated inversely by Northern Hemisphere temperature variations. If $P(t)$ is fossil fuel production at year t and $T(t)$ is temperature, then their model is written

$$\frac{dP}{dt} = \left(\alpha - \beta \frac{dT}{dt}\right) P \ , \quad P(0) = P_0 \ ,$$

where β is an additional fitting parameter. Using an earlier, cruder temperature record, they obtained the values given in the second row of the table.

A more realistic model, allowing for time lags between temperature changes and the corresponding reponses in production, can be written

$$\frac{dP}{dt} = \left\{\alpha - \beta \frac{d}{dt}\left[\int_{-\infty}^{t} \omega(t'-t,\tau) T(t') dt'\right]\right\} P \ ,$$

where $t'' \equiv t' - t$ is the time lag, $w(t'',\tau)$ is a *memory function* satisfying

$$w(t'',\tau) \geq 0 \ , \quad -\infty \leq t'' \leq 0 \ , \quad \int_{-\infty}^{0} w(t'',\tau) dt'' = 1 \ ,$$

and τ is a parameter measuring the rate at which $w(t'',\tau)$ tapers to zero for decreasing values of the time lag.

One way to specify $w(t'',\tau)$ is to assume a functional form in which τ becomes the fourth fitting parameter. We tried the following: the *boxcar function*,

$$w(t'',\tau) = \frac{1}{\tau} \ , \quad -\tau \leq t'' \leq 0 \ ,$$

the triangle function,

$$w(t'',\tau) = \frac{2}{\tau} + \left(\frac{2}{\tau^2}\right) t'' \ , \quad -\tau \leq t'' \leq 0 \ ,$$

and the half-Gaussian,

$$w(t'',\tau) = \frac{2}{\tau\sqrt{\pi}} \exp\left[-\left(\frac{t''}{\tau}\right)^2\right] \ , \quad -\infty \leq t'' \leq 0 \ .$$

We calculated the convolution integrals numerically, using for $T(t)$ a cubic interpolating spline representation of the temperature data (shown as the curve connecting the points in Fig. 1). The fitted parameter values are given in rows 3, 4 and 5 of the table, and the corresponding memory function estimates are plotted in Fig. 3. The half-Gaussian window gave the best fit, with an estimated $P(t)$ very similar to the solid curve in Fig. 2.

Another way to specify $w(t'';\tau)$ is to estimate it from the data. We did this by assuming that $w(t'';\tau) = 0$ for all lags with magnitude greater than $\tau = n + 1$, where n is a prespecified integer, and approximating the convolution integral by numerical quadrature, i.e.,

$$\frac{dP}{dt} = \left\{\alpha - \beta \frac{d}{dt}\left[\sum_{j=0}^{n} \omega_j w_j T(t-j)\right]\right\} P \ ,$$

where the ω_j are quadrature coefficients, and $w_j \equiv w(-j,\tau)$ are discrete values of the memory function to be estimated. The solution of this ODE can be written

$$P(t) = P_0 \exp\left\{\alpha t - \sum_{j=0}^{n} \beta_j \left[T(t-j) - T(-j)\right]\right\} \ ,$$

Model	P_0 [megatons]	α [yr^{-1}]	β [(°C)$^{-1}$]	τ [yr]	SSR
Simple Exponential	157	0.0313			64.2×10^5
Rust and Kirk (1982)	181	0.0320	1.20		
Boxcar Window	182	0.0332	1.13	10.1	8.52×10^5
Triangle Window	176	0.0334	1.22	16.3	7.82×10^5
Half-Gaussian Window	175	0.0335	1.23	9.95	7.53×10^5
Transfer Function	178	0.0334	1.18	14	7.39×10^5

where $\beta_j \equiv \beta\omega_j w_j$. This is a nonlinear *transfer function model* with $n+3$ fitting parameters $P_0, \alpha, \beta_0, \beta_1, \ldots, \beta_n$. The unit integral restriction on the memory function implies that $\beta = \sum \beta_j$, so, having calculated that value, the discrete memory function estimates can be obtained from the transfer coefficients by

$$w_j = \frac{\beta_j}{\beta\omega_j} \quad , \quad j = 0, 1, 2, \ldots, n \ .$$

Our strategy for determining τ was to make n as large as possible with $\beta_j > 0, j = 1, 2, \ldots, n$. The result was $n = 13$ ($\tau = 14$). The estimated memory function is plotted as connected circles in Fig. 3, and the other parameters are given in the last row of the table. The estimated $P(t)$, shown as a solid line in Figs. 2 and 6, tracks the measured data remarkably well.

Critics have claimed that 16 adjustable parameters should give a good fit using any time-series for $T(t)$. Therefore, we generated 300 artificial $T(t)$ records, using the least squares fit in Fig. 1 as a baseline and adding normally distributed random deviates with mean 0 and variance equal to that of the real temperatures about that baseline. Repeating the fit for each of those records, we obtained, in every case, one or more $\beta_j < 0$. We also obtained the SSR distribution shown in Fig. 4 where vertical lines mark the mean, -1σ, -2σ and -3σ points, and the black square marks the SSR for the real temperatures. Clearly, the probability of obtaining, by chance, such a low value of SSR, with all $\beta_j > 0$, is negligible. In fact, using measured data which averaged both the land and marine temperatures gave 4 negative β_j valuse and doubled the SSR. A good fit is obtained only with measured temperatures for the Northern Hemisphere land surface where most fossil fuel is consumed.

Lovelock (1979) propounded the *Gaia hypothesis* which postulates that life regulates and maintains the conditions needed to assure its survival. He noted that Earth's surface temperature has been nearly constant for the 3×10^9 year history of life, even though the Sun's luminosity has increased tenfold in that time. Surface temperature depends critically on the concentration of greenhouse gases in the atmosphere. According to the Gaia hypothesis, a warming caused by fossil fuel consumption should produce a feedback curtailing that consumption. The inverse modulation identified in the present study may represent just such a feedback.

Fossil fuel production is an indicator of economic vigor. It is not yet possible to predict future temperature variations, so the model described here can only make provisional predictions of future production by assuming various temperature scenarios. Three such scenarios are shown in Fig. 5, where circles represent measurements (1971-1988), and triangles, squares, and diamonds represent 20 years of increasing, stable, or decreasing temperatures, respectively. The corresponding provisional predictions are shown in Fig. 6 where circles represent measurements (1971-1986), the solid line represents model predictions for those years, and the triangles, squares, and diamonds are model predictions for the 3 future scenarios. The initial prediction in all cases is for 4 or 5 years of declining or static production. Thereafter, the cooling scenario predicts spectacular recovery, the stable temperatures predict 4 or 5 additional years of static production followed by recovery, and the continued warming scenario predicts 4 or 5 years of further declines followed by an almost static production . The production totals since 1986 are not yet available, but the current economic situation does not contradict the initial predictions.

References

[1] Boden, T.A. (1988) Numeric data package NDP001.REV, ORNL/CDIC-17, Carbon Dioxide Information Analysis Center, Oak Ridge, TN.

[2] Jones, P.D., Raper, S.C.B., Bradley, R.S., Diaz, H.F., Kelly, P.M., and Wigley, T.M.L. (1986) *Jour. of Climate and Appl. Meteor.* **25**, pp.161-179.

[3] Lovelock, J.E. (1979) *Gaia: A New Look at Life on Earth*, Oxford University Press, Oxford.

[4] Rust, B.W. and Kirk, B.L. (1982) *Enviroment International* **7**, pp. 419-422.

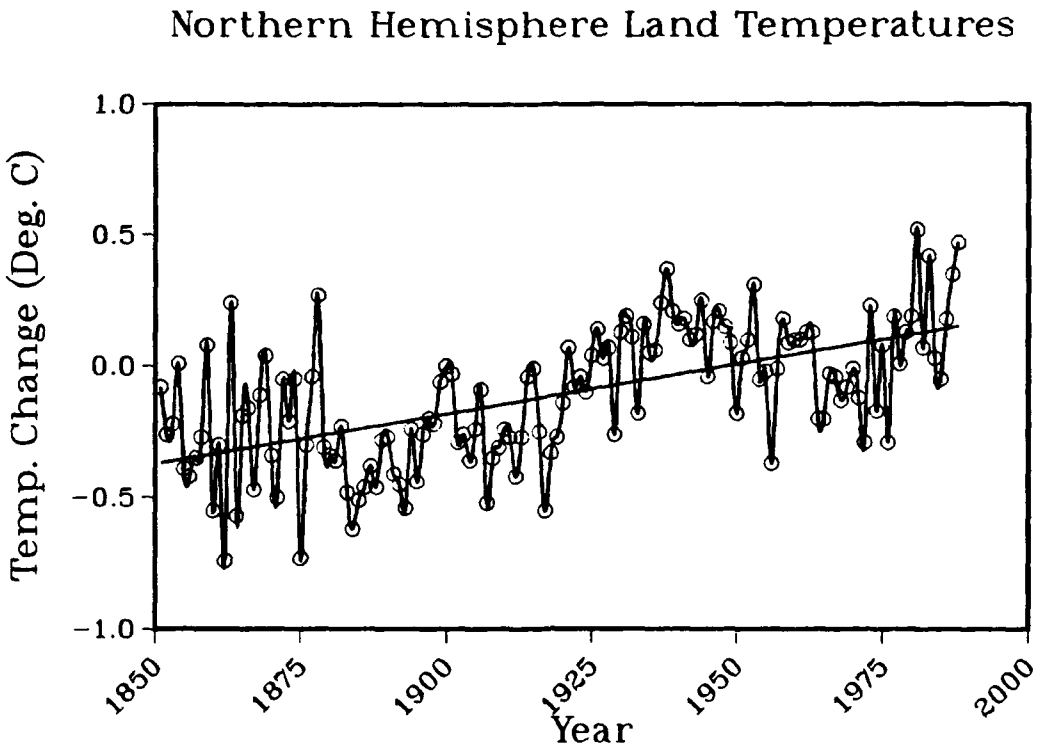


Figure 1

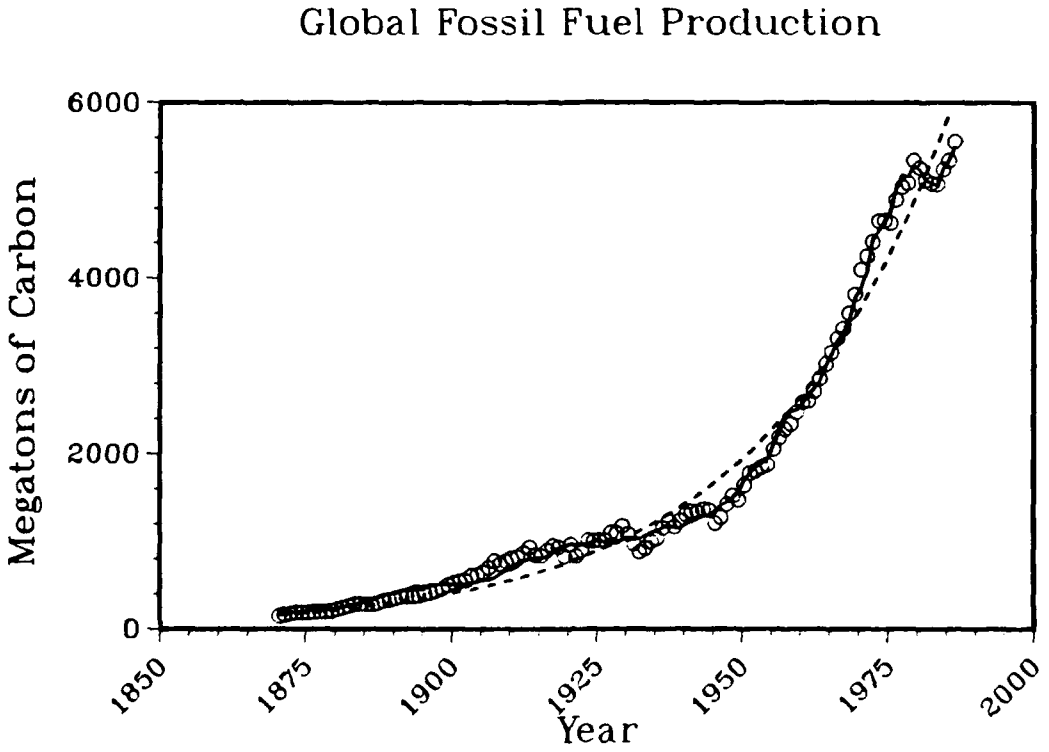


Figure 2

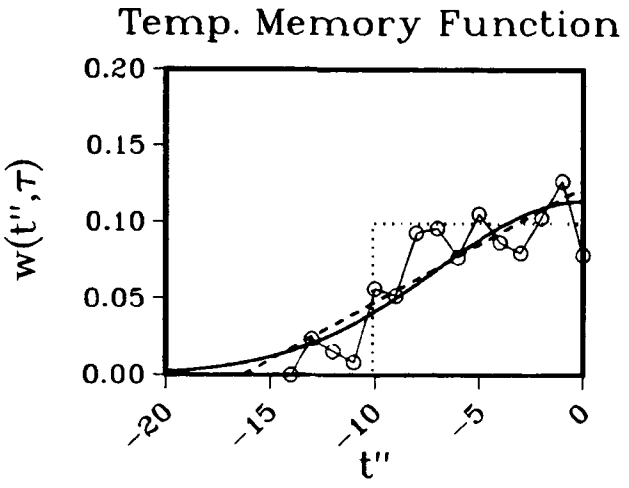


Figure 3

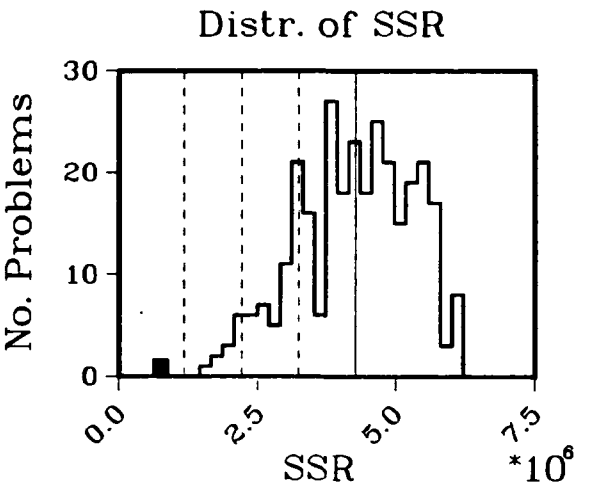


Figure 4

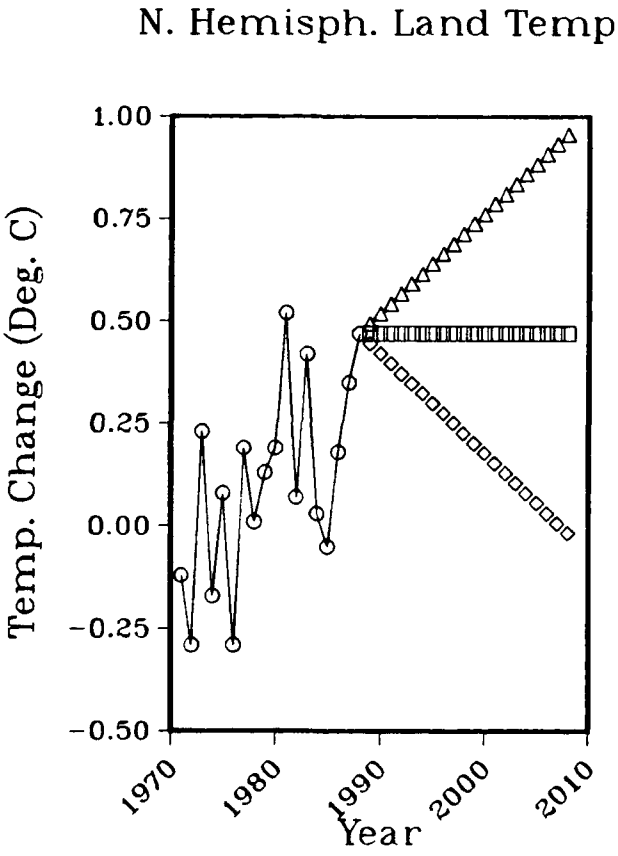


Figure 5

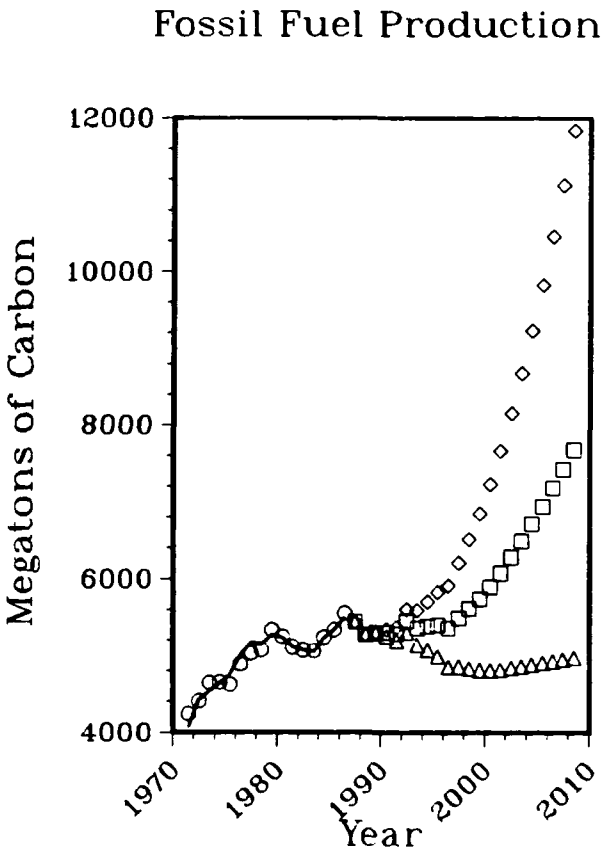


Figure 6

Influence on the Cross-Validated Smoothing Parameter in Spline Smoothing

William Thomas*
Division of Biostatistics
University of Minnesota
Minneapolis, Minnesota 55455

Abstract

We address the problem of influence in estimating the smoothing parameter when fitting a univariate smoothing spline. Using the local-influence methods of Cook (1986), a diagnostic is derived to identify observed responses that locally influence the choice of smoothing parameter by generalized cross-validation. The diagnostic motivates a discussion of an apparent sensitivity of generalized cross-validation.

1 Introduction

Consider scalar responses y_j generated according to the model $y_j = \mu(t_j) + \epsilon_j$, where μ is a "smooth" regression function, $a \le t_1 < \ldots < t_n \le b$, and the errors ϵ_j are uncorrelated, with zero mean and constant variance. We assume that μ is smooth enough to belong to the set $W_2^m[a, b]$ of functions g that, for some fixed m, have $m-1$ continuous derivatives and square-integrable mth derivative $g^{(m)}$ in $[a, b]$. The smoothing spline estimator of μ satisfies a penalized least squares criterion: it is the minimizer over $g \in W_2^m$ of

$$\frac{1}{n}\sum_{j=1}^{n}\{y_j - g(t_j)\}^2 + \lambda \int_a^b \left\{g^{(m)}(t)\right\}^2 dt, \quad \lambda > 0. \tag{1.1}$$

Here, the integral is a penalty for roughness in the spline. For a fixed value of $\lambda > 0$, the smoothing spline is a linear smoother, *i.e.*, there is a "hat" matrix H_λ, depending only on the design points $\{t_i\}$ and λ, that transforms the data vector $\mathbf{y}$ into the vector of smoothing spline fitted values: $H_\lambda \mathbf{y} = \hat{\mu}_\lambda$. Discussions of smoothing splines in statistics may be found in Wegman and Wright (1983), Silverman (1985), Eubank (1988), and Wahba (1990).

*Research supported in part by the National Institutes of Health (GM39015-01A1)

One chooses the *smoothing parameter* λ in (1.1) to balance the competing aims of smoothness and close fit to the data. Small values of λ produce rougher curves that follow the data more closely while large values of λ give smoother curves. A popular data-driven method for selecting λ is generalized cross-validation (GCV), introduced by Craven and Wahba (1979). The GCV choice $\hat{\lambda}$ minimizes over $\lambda > 0$

$$G(\lambda) = \frac{\|(I - H_\lambda)\mathbf{y}\|^2}{(n - \operatorname{tr} H_\lambda)^2} = \frac{\mathbf{e}_\lambda^{\mathrm{T}}\mathbf{e}_\lambda}{(n - \operatorname{tr} H_\lambda)^2}, \tag{1.2}$$

where I is the $n \times n$ identity matrix and $\mathbf{e}_\lambda = (I - H_\lambda)\mathbf{y}$ is the vector of residuals. For discussions of GCV in related smoothing problems, see Li (1985), Hall and Titterington (1987), and Härdle, Hall, and Marron (1988).

Analogues of familiar linear-regression diagnostics, based on case deletion, have been proposed for smoothing splines; see Wendelberger (1981), Eubank (1984, 1985), Silverman (1985), Eubank and Gunst (1986). However, no diagnostics for GCV have appeared in the literature. Although case-deletion diagnostics for the GCV choice $\hat{\lambda}$ are an obvious approach, they are computationally infeasible for large datasets. Further, as will be discussed in Section 3, the estimate $\hat{\lambda}$ is apparently sensitive to groups of observations acting together rather than single outlying points. Hence case-deletion diagnostics may not be very relevant.

2 A diagnostic for influential responses

To develop a diagnostic when influential groups of cases are a possibility, a natural approach is to perturb all observations simultaneously, rather than modifying or deleting single cases. To do this, we add a vector ω of small perturbations to produce $\mathbf{y}_\omega \equiv \mathbf{y} + \omega$. Through

their local influence (Cook, 1986), we can identify groups of responses that play a large role in the determination of the GCV estimator $\hat{\lambda}$.

For each ω, GCV applied to the perturbed data $\mathbf{y}_\omega$ selects $\hat{\lambda}(\omega)$. This defines a map $\omega \mapsto \hat{\lambda}(\omega)$ as ω ranges in an open set about $\omega = 0$, where $\hat{\lambda}(0) = \hat{\lambda}$ from the unperturbed data. We approximate the surface $\hat{\lambda}(\omega)$ with its tangent plane at $\omega = 0$ and find the direction of maximum slope $t_{\max}$ on this plane. It can be shown that $t_{\max} \propto \partial\hat{\lambda}(\omega)/\partial\omega^{\mathrm{T}}$, evaluated at $\omega = 0$. For the perturbation defined above, it is straightforward to calculate

$$t_{\max} \propto (cI - H)(I - H)^2\mathbf{y}, \qquad (2.1)$$

where $c = \mathrm{tr}\,\{H(I - H)\}/\mathrm{tr}\,(I - H)$ and $H = H_{\hat{\lambda}}$.

The essential idea is that a direction of large local change in the $\hat{\lambda}(\omega)$ surface at $\hat{\lambda}(0) = \hat{\lambda}$ corresponds to perturbation of influential responses, the vector $t_{\max}$ approximates this direction, and therefore large components of $t_{\max}$ flag locally influential observations. The diagnostic is a plot of $t_{\max}$ against case number, where cases with relatively large absolute components are jointly influential. The sign of a component indicates the direction in which to alter the response to produce a large (local) change in $\hat{\lambda}$.

3 Sensitivity of GCV

To illustrate the diagnostic, we consider the data shown in Figure 1, generated by adding independent Uniform$[-3, 3]$ errors to the sinusoidal mean function indicated by the dotted curve. The solid curve is the periodic cubic smoothing spline (defined below) fitted to the simulated data, using GCV to select $\hat{\lambda} = 7.6\times10^{-5} \approx 5\{(n-1)/2\}^{-4}$. An index plot of $t_{\max}$ (not shown) identifies five jointly influential cases, marked with filled circles: moving the responses for cases 17, 19, and 31 in one direction, and cases 4 and 18 in the opposite direction will produce a large local change in $\hat{\lambda}$. Note that the locally-influential responses are not "outlying" points.

Some experience with the diagnostic suggests that when $\hat{\lambda}$ is large, it is not particularly sensitive to small subsets of observations. However, when $\hat{\lambda}$ is very small, it seems to be sensitive to groups of observations which make it appear that the regression function has important high-frequency components. This can be made precise by examining what happens to the high-frequency components of $\mathbf{y}$ in GCV and the diagnostic. For simplicity, we consider the special case of periodic smoothing splines (Eubank, 1988, sec 6.3.1) where the mapping from the "time domain" (y) to the frequency domain is particularly transparent. However, the ideas extend in principle to the general case.

For periodic cubic splines ($m = 2$), we assume the model (1.1) and in addition that: (i) $t_1, \ldots, t_n$ are equally spaced in $[0, 1]$, (ii) μ is smoothly periodic in the sense that $\mu(0) = \mu(1)$ and $\mu^{(1)}(0) = \mu^{(1)}(1)$, and, for simplicity, (iii) n is odd. Write the Fourier transform of $\mathbf{y}$ as $\mathbf{f}(\mathbf{y}) = X\mathbf{y}/n$, where X is the $n \times n$ matrix with rows

$$x_r^{\mathrm{T}} = (1, \exp(2\pi ir/n), \ldots, \exp\{2\pi i(n-1)r/n\}),$$

in the order $r = -(n-1)/2, \ldots, (n-1)/2$, and where $i^2 = -1$. The Fourier coefficient of $\mathbf{y}$ for the frequency r/n is the rth component of $\mathbf{f}(\mathbf{y})$,

$$f_r(\mathbf{y}) = \frac{1}{n}\sum_{k=1}^{n} y_k \exp\{2\pi i(k-1)r/n\},$$

so that the f_r for large $|r|$ correspond to high frequencies. Then $|f_r(\mathbf{y})|^2$ is the power of the signal $\mathbf{y}$ at frequency r/n.

The cubic periodic spline estimate of μ for fixed $\lambda > 0$ is, to a high order of approximation, $\hat{\mu}_\lambda \equiv X^{\mathrm{H}}WX\mathbf{y}/n = X^{\mathrm{H}}W\mathbf{f}(\mathbf{y})$, where X^{H} is the hermitian, or conjugate transpose, of X, and $W = W(\lambda)$ is a diagonal matrix with diagonal elements $w_r(\lambda) = (1 + \lambda r^4)^{-1}$, for $r = -(n-1)/2, \ldots, (n-1)/2$. Since the weights $w_r(\lambda)$ decrease with increasing frequency $|r/n|$, W acts as a low-pass filter in the frequency domain which smooths the data by damping high-frequency components of $\mathbf{y}$. The amount of damping depends on λ: small values produce less damping, large values more.

To examine influence on GCV, we rewrite $t_{\max}$ in (2.1) as a function of the Fourier coefficients of the data $\mathbf{f}(\mathbf{y})$:

$$t_{\max} \propto X^{\mathrm{H}}(cI - W)(I - W)^2\mathbf{f}(\mathbf{y}),$$

where c is defined below (2.1). The filter $(cI - W)(I - W)^2$ is increasing in $|r/n|$ for all values of $\hat{\lambda}$ and so acts as a high-pass filter, increasing the output power at high frequencies. Thus, $t_{\max}$ has large absolute components corresponding to groups of responses which make large contributions to the high-frequency components of the data $\mathbf{y}$.

Finally, the GCV criterion (1.2) can be expressed in terms of the power $|f_r(\mathbf{y})|^2$ at various frequencies as

$$G(\lambda) = \sum_{|r|\le(n-1)/2} n\theta_r(\lambda)|f_r(\mathbf{y})|^2,$$

where

$$\theta_r(\lambda) = \left(\frac{\lambda r^4}{1+\lambda r^4}\right)^2 \Bigg/ \left(\sum_{|j|\le(n-1)/2} \frac{\lambda j^4}{1+\lambda j^4}\right)^2$$

Note that, in contrast to the weights $w_r(\lambda)$ for the periodic spline, the GCV weights $\theta_r(\lambda)$ are strictly increasing with $|r/n|$, so that high-frequency components of the data may have a larger role in determining $\hat{\lambda}$. The amount by which high frequencies outweigh low frequencies depends critically on the value of λ. Figure 2 shows several sequences of GCV weights $\theta_r(\lambda)$ with $n = 51$, for λ equal to $\lambda_0 \equiv \{(n-1)/2\}^{-4}$, $10\lambda_0$, $10^2\lambda_0$, and $10^4\lambda_0$. When $\hat{\lambda}$ is near $\{(n-1)/2\}^{-4}$, high frequencies receive substantially greater weight. Thus, when GCV is minimized at a very small $\hat{\lambda}$, it may be driven by small groups of cases which contribute to the power of $\mathbf{y}$ at high frequencies. When GCV is minimized at a large $\hat{\lambda}$, it is relatively insensitive, since higher and lower frequences have nearly equal weight.

Acknowledgement

I am grateful to Randy Eubank for many helpful discussions, and to John Adams, Steve Marron, and Gary Oehlert for helpful suggestions.

References

Cook, R.D. (1986), "Assessment of Local Influence," (with discussion), *Journal of the Royal Statistical Society*, Ser. B, 48, 133–169.

Craven, P. and Wahba, G. (1979), "Smoothing Noisy Data with Spline Functions: Estimating the Correct Degree of Smoothing by the Method of Generalized Cross-Validation," *Numerische Mathematik*, 31, 377–403.

Eubank, R.L. (1984), "The Hat Matrix for Smoothing Splines," *Statistics and Probability Letters*, 2, 9–14.

Eubank, R.L. (1985), "Diagnostics for Smoothing Splines," *Journal of the Royal Statistical Society*, Ser. B, 47, 332-341.

Eubank, R.L. (1988), *Spline Smoothing and Nonparametric Regression*. New York: Marcel Dekker.

Eubank, R.L. and Gunst, R.F. (1986), "Diagnostics for Penalized Least-Squares Estimators," *Statistics and Probability Letters*, 4, 265–272.

Hall, P. and Titterington, D.M. (1987), "Common Structure of Techniques for Choosing Smoothing Parameters in Regression Problems," *Journal of the Royal Statistical Society*, Ser. B, 49, 184–198.

Härdle, W., Hall, P., and Marron, J.S. (1988), "How Far Are Automatically Chosen Regression Parameters from Their Optimum?" (with discussion), *Journal of the American Statistical Association*, 83, 86–101.

Li, K.C. (1985), "From Stein's Unbiased Risk Estimates to the Method of Generalized Cross-Validation," *The Annals of Statistics*, 13, 1352–77.

Silverman, B.W. (1985), "Some Aspects of the Spline Smoothing Approach to Non-Parametric Regression Curve Fitting (with discussion)," *Journal of the Royal Statistical Society*, Ser. B, 47, 1–52.

Wahba, G. (1990), *Spline Models in Statistics*. Philadelphia: SIAM.

Wegman, E.J. and Wright, I.W. (1983), "Splines in Statistics," *Journal of the American Statistical Association*, 78, 351–365.

Wendelberger, J.G. (1981), "The Computation of Laplacian Smoothing Splines with Examples," Technical Report 648, Department of Statistics, University of Wisconsin—Madison.

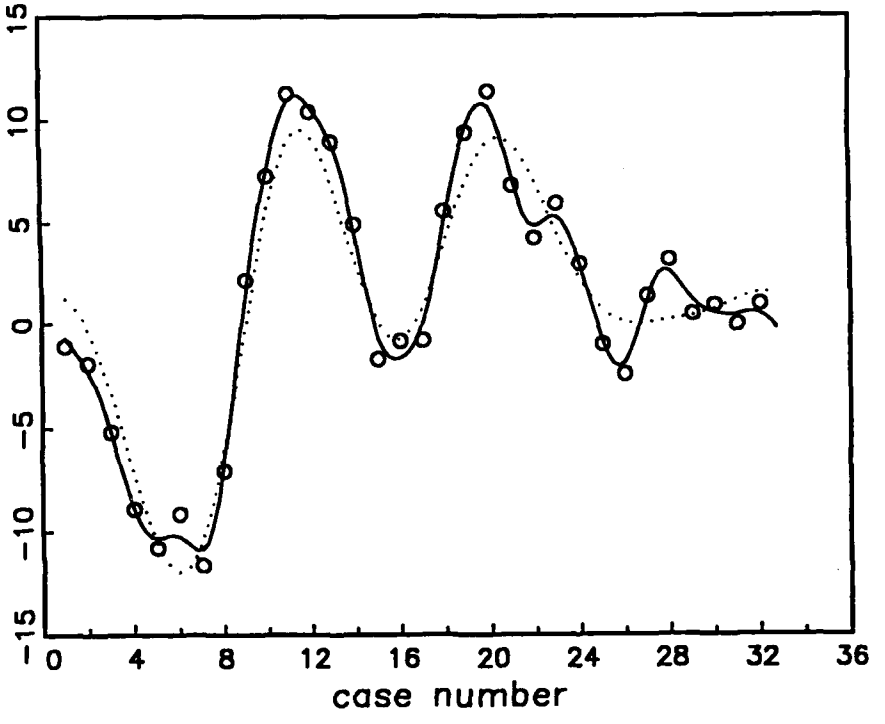


Figure 1. Simulated data based on a periodic regression function (dotted curve) with a cubic spline fit (solid curve). Data is plotted against case number rather than t.

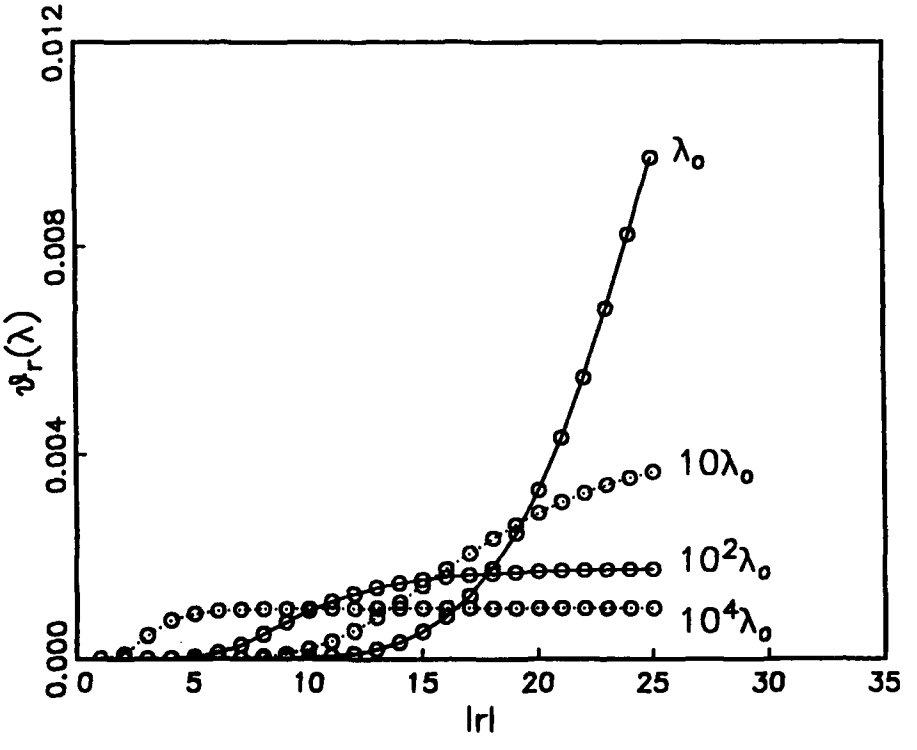


Figure 2. GCV coefficients $\theta_r(\lambda)$ vs. $|r|$, for several values of λ, given as multiples of $\lambda_0 = \{(n-1)/2\}^{-4}$.

On "Fit the Short Curve" Principle for Smoothing Nonparametric Estimators

Andrzej S. Kozek[1] and Eugene F. Schuster
Department of Mathematical Sciences
The University of Texas at El Paso
El Paso, Texas 79968

1 Introduction

Let (X,Y) be a bivariate random vector with $E|Y| < \infty$. The nonparametric regression problem is to estimate the regression function

$$r(x) = E(Y|X = x) \qquad (1)$$

based on a random sample (X_i, Y_i), i=1,$\cdots$,n from (X,Y).

The Nadaraya-Watson (NW), the Nearest Neighbor (NN), and the Optimal Quantile (OQ) kernel type estimators of $r(x)$ defined in (2)-(4) depend on smoothing parameters h, k and p, respectively. The asymptotic optimal form of these smoothing parameters is known, see Collomb (1977) and Mack (1981). This information, however, is not sufficient in practical applications and data driven (DD) methods for choosing smoothing parameters have been developed, see Hall (1984), Rice (1984), Härdle and Marron (1985), Marron and Härdle (1986), Bhattacharya and Mack (1987), Härdle, Hall and Marron (1988) and Kozek and Schuster (1990). One popular DD method of choosing smoothing parameters, the so called leave-out-one-at-a-time cross-validation principle (CVP) , chooses the smoothing parameter, say v, to minimize $CV(v)$ of equation (6). The CVP measures fit of the estimator to the data on the set $\{X_1, \cdots, X_n\}$. This criterion imposes no condition on the behavior of the curve between the points X_i. It is not surprising then that we frequently observe excellent fit, but simultaneously nonregular behavior elsewhere. It is well-known (and has been our experience) that the CVP tends to choose an estimator which overfits the data. This lack of smoothness is often visible in small or moderate sample sizes, say 10-50. In fact, for sample sizes this small, the CV function is often degenerate in the sense that it is not defined for small values of the smoothing parameter and it increases on the interval where it is well defined (see Figure 1).

What we desire in practice is a uniform behavior of the regression estimator and its derivatives. In this context, many interesting ideas have appeared in spline estimation for the case of nonrandom X and special experimental designs, see Wahba (1990) and Eubank (1990). The key of the success here seems to be in the formulation of minimization criterion :

- minimize a simple expression which penalizes both for the lack of fit and for the lack of smoothness.

In Section 2 we propose our criterion for the adaptive choice of the smoothing parameter for kernel type estimators in case of random X. As in spline theory, we penalize both for lack of fit and for lack of smoothness.

2 Fit the Short Curve Principle

We restrict our consideration to the following three kernel type estimators of the regression function

$$\hat{r}_h(x) = \frac{\sum_{i=1}^n Y_i K\left(\frac{x-X_i}{h}\right)}{\sum_{i=1}^n K\left(\frac{x-X_i}{h}\right)}, \qquad (2)$$

$$\hat{r}_k(x) = \frac{\sum_{i=1}^n Y_i K\left(\frac{x-X_i}{d_k(x)}\right)}{\sum_{i=1}^n K\left(\frac{x-X_i}{d_k(x)}\right)}, \qquad (3)$$

$$\hat{r}_p(x) = \frac{\sum_{i=1}^n Y_i K\left(\frac{x-X_i}{q_p(x)}\right)}{\sum_{i=1}^n K\left(\frac{x-X_i}{q_p(x)}\right)}, \qquad (4)$$

where K is a nonnegative kernel and h, $d_k(x)$, and $q_p(x)$ are window bandwidths corresponding to the Nadaraya-Watson (Nadaraya (1964), Watson (1964)) , k-th nearest neighbor (Cover(1968), Collomb (1980)) and p-th optimal quantile (Kozek and Schuster (1990)) estimators, respectively. Here $d_k(x)$ is the distance from x to its k-th nearest neighbor in the sample $X_1, \cdots, X_n$ and $q_p(x)$ is the p-th quantile corresponding to $\bar{Q}_n^x(\cdot)$, a continuous linearly smoothed version of the empirical distribution function $Q_n^x(\cdot)$ based on $|x-X_1|, \cdots, |x-X_n|$. $\bar{Q}_n^x(\cdot)$ is given by

$$\bar{Q}_n^x(t) = \begin{cases} 0 & \text{if } t < d_1(x) \\ [k-1+\frac{t-d_k(x)}{d_{k+1}(x)-d_k(x)}]/(n-1) & \text{if } t \in I_k(x) \\ 1 & \text{if } t \geq d_n(x) \end{cases}$$

where $d_1(x) \leq \cdots \leq d_n(x)$ are ordered quantities $|x-X_1|, \cdots, |x-X_n|$, and $I_k(x) = [d_k(x), d_{k+1}(x))$. Whenever any of the estimators (2)-(4) is not well defined, i.e. its denominator equals zero, we assign a large

[1] A. S. Kozek is from the Institute of Computer Science, U. of Wroclaw, Poland, and is visiting thru academic year 1990-1991 at U.T. El Paso.

constant for its value, say 10^{10}. Such a convention is useful from a numerical point of view and has been implied in the *Fit Short* (FS)[1] package. The simulations via FS have been made for a variety of long and short tailed kernels including the Gaussian kernel and the continuously differentiable (compact support) quartic kernel

$$K(x) = \begin{cases} \frac{15}{16}(1-x^2)^2 & \text{for } x \in [-1,1] \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

The estimators $\hat{r}_h(x), \hat{r}_k(x)$ and $\hat{r}_p(x)$ depend on the smoothing (window width) parameters h, k, and p. Let $\hat{r}_v(x)$ stand for the estimator corresponding to a parameter $v \in \{h,k,p\}$.

To penalize for lack of fit we use the $CV(v)$ function

$$CV(v) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{r}_v^{(-i)}(X_i))^2 \quad (6)$$

where $\hat{r}_v^{(-i)}(x)$ is given by one of (2)–(4), but is based on all data pairs except for leaving out (X_i, Y_i). Let $L(v)$ be a function to penalize for roughness. Since overfitting the data would tend to produce estimators $r_v(x)$ whose derivatives were large in magnitude midway between adjacent observations on the independent variable, we looked for natural functionals $L(v)$ which penalize for large values of $|r_v'(x)|$ at the midpoints of the order statistics, i.e. at points $(X_{(i-1)} + X_{(i)})/2$.

Penalty functionals $L(v)$ we considered were the length, the total variation, and the curvature of the estimator $\hat{r}_v(x)$. A natural criterion to impose on a penalizing procedure is that it produces a smoothing parameter which is invariant under linear transformations on the dependent variable Y. The total variation (the integral of the absolute value of the first derivative) and the global measure of curvature (the integral of the square of the second derivative) possess this invariance property. The length (the integral of the square root of 1 plus the square of the first derivative) does not. The curvature functional possesses desirable theoretical and computational properties as a penalty function in cubic spline estimation which are not present in the present problem. When the derivative is large, the length and variation functions produce essentially the same values. Moreover, the length criterion is more easily understood by practitioners and seemed to work somewhat better than the variation functional in our experimentation. For these reasons, we have chosen to present our penalizing criterion for the length functional. Our general approach applies to any of the DD criterions discussed in Härdle et al. (1988) and any penalizing functional. For the sake of simplicity, however, we restrict our attention to the CVP and we penalize for estimators with excessive length. We use the following Riemann sum approximation to the length of $\hat{r}_v(x)$ (and $r(x)$) on an interval [A,B]:

$$L(v) = \sum \Delta X_i \sqrt{1 + (\hat{r}_v'(X_i^*))^2} \quad (7)$$

where $\Delta X_i = X_{(i)} - X_{(i-1)}$, $X_i^* = (X_{(i-1)} + X_{(i)})/2$, $\hat{r}_v'(x)$ is an approximation to the derivative of the estimator $\hat{r}_v(x)$ at x and the sum $\sum$ is over order statistics $X_{(i)}$ in an interval $[A, B]$. Most of our simulations were run with A and B corresponding to symmetric trimming of 0-10% of the data pairs corresponding to the smallest and largest $X_{(i)}$'s. The FS package simplifies computations of the derivatives of the NN and OQ estimators, by treating $d_k(x)$ and $q_p(x)$ as constants.

Our **Fit the Short Curve Principle (FSCP)** can now be described as follows:

- find the smoothing parameter v_0 minimizing the cross validation term $CV(v)$ in (6),
- choose the smoothing parameter v which minimizes

$$FSC(v) = \frac{CV(v)}{CV(v_0)} + \frac{L(v)}{L(v_0)}. \quad (8)$$

We prefer (8) to any gauge function of $CV(v)$ and $L(v)$ we have tried. $CV(v_0)$ will tend to underestimate the mean square error of the regression estimator and $L(v_0)$ will tend to overestimate length. Thus the FSC function of (8) will tend to weigh the fit criterion more heavily and should be near 2 when properly gauged. We did iterate this procedure. However, there was little improvement in the estimators produced in the second iteration.

In the next section we shall see that one can use any DD criterion including FSCP to select the bandwidth parameter in some specified envelope and retain the strong consistency of $\hat{r}_v(x)$. Computer simulations using FSCP, examples, and conclusions are discussed in Section 4.

3 The strong consistency

In this section we show that if some asymptotic restrictions on the bandwidth parameter sequence are imposed and some mild regularity conditions are satisfied, then any data driven choice of the bandwidth leads to a pointwise strongly consistent sequence of estimators. Hence we can conclude that estimators (2)–(4), with window width parameter selected from an envelope by the FSCP, converge pointwise with probability 1 to the true regression function. In this direction let

$$C(n) \searrow 1 \quad as \quad n \longrightarrow \infty, \quad (9)$$

$$h(n) = Dn^{-1/5}, \quad (10)$$

$$h_1(n) = h(n)/C(n), \quad (11)$$

$$h_2(n) = C(n)h(n), \quad (12)$$

$$c_1 H(|x|) \leq K(x) \leq c_2 H(|x|) \ 0 < c_1 < c_2, \quad (13)$$

$$K(x) > c > 0 \quad if \quad |x| < r \ some \ c, \ r > 0, \quad (14)$$

$$K(\lambda x) \quad is \quad nonincreasing \ in \lambda, \lambda > 0, \quad (15)$$

[1] FS was developed in 1991 at U.T. El Paso with the assistance of Krzysztof Kozek.

where K is a Borel kernel and H is a nonnegative, decreasing, bounded, Lebesgue integrable function on $x > 0$. From Theorem 1 in Kozek and Schuster (1991) we infer the convergence of general DD kernel-type estimators.

Theorem 3.1 *Let $(X,Y),(X_1,Y_1),\ldots,(X_n,Y_n)$ be independent, identically distributed bivarate random variables such that the probability distribution μ of X has no singular component and*

$$E|Y|(log(1+|Y|))^{1+\delta} < \infty \text{ for some } \delta > 0. \quad (16)$$

Let the kernel K and the bandwidth h_1 and h_2 satisfy (9)...(15). If $h_1 \leq h \leq h_2$ and $h = h(n,x,X_1,Y_1,\ldots,X_n,Y_n)$, then the NW estimator (2) with the bandwidth h is for μ-a.e. x, convergent with probability one to $r(x)$.

Corollary 3.2 *If the minimization of FSC in (8) is over $v = h \in [Dn^{-1/5}/C(n), Dn^{-1/5}C(n)]$, then the NW estimator (2) with the bandwidth h chosen by the FSCP is for μ-a.e. x, convergent with probability one to $r(x)$.*

From Theorem 3.1 it follows that the strong consistency does not impose any severe restrictions upon the choice of the sequence $C(n)$ in (9). It allows adjustments suitable to local predilections.

From Theorems 2 and 3 of Kozek and Schuster (1991) it follows that if X has an almost everywhere continuous Lebesgue density, then:

Corollary 3.3 *If the minimization of FSC in (8) is over $v = k \in [Dn^{4/5}/C(n), Dn^{4/5}C(n)]$, then the NN estimator (3) with the bandwidth parameter k chosen by the FSCP is for μ-a.e. x, convergent with probability one to $r(x)$.*

Corollary 3.4 *If the minimization of FSC in (8) is over $v = p \in [Dn^{-1/5}/C(n), Dn^{-1/5}C(n)]$, then the OQ estimator (4) with the bandwidth parameter p chosen by the FSCP is for μ-a.e. x, convergent with probability one to $r(x)$.*

4 Conclusions and Examples

We summarize our simulation experience with the estimators (2)-(4) using FSCP as implemented in the statistical package FS running on IBM XT, AT, PS/2 or on IBM compatible personal computers. Frequently, for samples of sizes 10-100, the CV function is strictly increasing on the interval where it is well defined. Since the length functional decreases rapidly as window width increases one can heuristically argue that the FSCP has the desired effect we observed in all simulations, i.e. FSCP chooses a larger window than that given by the CVP alone. As a result the estimators obtained by the FSCP are smoother than those obtained by the CVP. Typically, the FSCP, in contrast to the CVP, has an objective function $FSC(v)$ with a well determined minimum (see Figure 1) occuring at a point which seems to reasonably balance fit with smoothness.

Poor results were obtained for kernels which can assume negative values. Kernels of this type possess optimal properties in the fixed design case of nonrandom X which do not seem to be present for the ratio type kernel estimators (2)-(4) in the bivariate case of random X. The quartic kernel in (5) is a computationally simple, symmetric, unimodal, and continuously differentiable type kernel desired in the FSCP. Overall it worked well for NW estimators with little difference between the CVP and the FSCP criterion in cases where there were no large spacings among the X_i's. NW estimators using the CVP with Guassian or Student's t kernels were occasionly quite rough. The FSCP based estimator frequently showed substantial improvement in these cases.

Our experience leads us to believe that there are inherent limitations with kernel estimators of the NW type of (2) which utilize a constant window width. Estimators NN and OQ of (3)-(4) allow for varying window width and adapt to the local density of the X variable. The NN estimator is quite rough, particularly for small samples, and is computationally awkward and time consuming to analyze using CVP or FSCP type criteria. The OQ estimator is a smoothed version of the NN estimator studied by Kozek and Schuster (1990) which moderates these difficulties and seemed to perform reasonably well over a variety of regression models. It seems much less sensitive to both the choice of kernel and the presence of large spacings in the X_i's. In cases where the CVP worked well there was often no significant difference between the CVP and the FSCP OQ estimates.

When properly normalized, the denominators of the regression estimators (2)-(4) estimate the density of X in the absolutely continuous case. The FS package includes an option to compare these density estimates with the true density in simulations. Our experience indicates strong links between values of smoothing parameters corresponding to good regression estimators and good estimates of the density of the random variable X.

To illustrate points raised in our discussions we have used the FS package to take a random sample of 20 pairs from the regression model $Y = X^2 + \epsilon$ where X is standard normal and ϵ is independent of X and normally distributed with mean 0 and standard deviation 0.1. Figure 1 contains the graphs of the CV and FSC functions using the Gaussian kernel. Figure 2 contains the data pairs, the true regression function, the NW estimator with smoothing parameter selected by both CVP and FSCP, and the OQ estimator selected by FSCP. Ten percent trimming was used in (7) for these examples.

References

[1] BHATTACHARYA, P.K. AND MACK, Y.P. (1987). Weak convergence of k-NN density and regression estimators with varying k and applications. *Ann. Statist.* **15**, 976-994.

[2] COLLOMB, G. (1977). Quelques proprietes de la méthode du noyau pour l'estimation nonparamétrique de la régression en un point fixé. *Comptes Rendus de l'Acad. des Sci. de Paris* **285 A**, 289-292.

[3] COLLOMB, G. (1980). Estimation de la regression par la methode des k points les plus proches avec noyau: quelques propriétés de convergence ponctuelle. *Lect. Notes Math.* **821**, 159-175.

[4] COVER, T. M. (1968). Estimation by the nearest neighbor rule. *IEEE Trans. Information Theory* **14**, 50-55.

[5] EUBANK, R.L. (1988). *Spline Smoothing and Nonparametric Regression.* New York: Marcell Dekker, Inc.

[6] HALL, P. (1984). Asymptotic properties of integrated square error and cross-validation for kernel estimation of a regression function. *Z. Wahrsch. Verw. Gebiete* **67**, 175-196.

[7] HÄRDLE, W., HALL, P. AND MARRON, J. S. (1988). How far are automatically chosen regression smoothing parameters from their optimum? *JASA* **83**, 401, 86-95.

[8] HÄRLDE, W. AND MARRON, J. S. (1985). Optimal bandwidth selection in nonparametric regression function estimation. *Ann. Statist.* **13**, 1465-1481.

[9] KOZEK, A. S. AND SCHUSTER, E. F. (1990). Optimal quantile principle for selecting variable bandwidth in regression estimators. *To appear in: Proceedings of the Computer Science and Statistics: 22nd Symposium on the Interface.*

[10] KOZEK, A.S. AND SCHUSTER, E. F. (1991). Strong consistency of nonparametric regression estimates with data driven bandwidths *submitted for publication.*

[11] MACK, Y. P. (1981). Local properties of k-NN regression estimates. *SIAM J. Alg. Disc. Meth.* **2**, 311-323.

[12] MARRON, J. S., AND HÄRDLE, W. (1986). Random approximations to some measures of accuracy in nonparametric curve estimation. *J. Multivariate Anal.* **20**, 91-113.

[13] NADARAYA, E. A. (1964). On estimating regression *Theory Probab. Appl.* **9**, 157-159.

[14] RICE, J.(1984). Bandwidth choice for nonparametric regression. *Ann. Statist.* **12**, 1215-1230.

[15] WAHBA, G. (1990). *Spline Models for Observational Data.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

[16] WATSON, G. S. (1964). Smooth regression analysis. *Sankhya Ser.* **A 26**, 359-372.

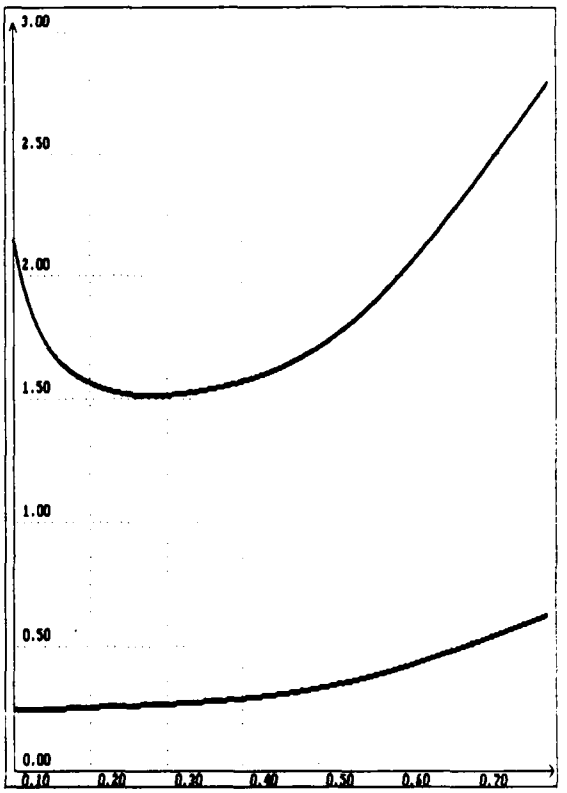


Figure 1. *FSC* (top curve) and *CV* functions.

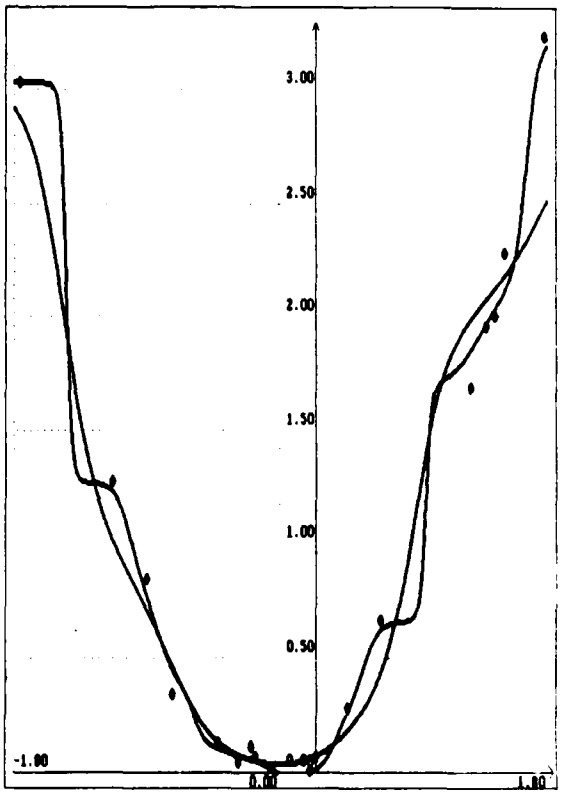


Figure 2. NW CVP (rough curve) and NW FSCP estimators

Exact Stratified Linear Rank Tests for Binary Data

Cyrus R. Mehta† Nitin Patel‡ Pralay Senchaudhuri†

† Department of Biostatistics, Harvard School of Public Health
and
‡ Indian Institute of Management, Ahmadebad, India

Abstract

We present an efficient network algorithm for generating exact permutational distributions for linear rank tests defined on stratified $2 \times c$ contingency tables. The algorithm can evaluate exact one and two sided p-values, and compute exact confidence intervals for trend parameters arising from certain loglinear and logistic models embedded in these contingency tables. It is especially efficient for highly imbalanced categorical data, a situation where the asymptotic theory is unreliable. Part of the algorithm can be adapted to evaluating the conditional maximum likelihood and its derivatives for the logistic regression model, with grouped data. We illustrate the techniques with an analysis of two data sets; the leukemia data on the Hiroshima atomic bomb survivors, and data from a clinical trial of bone marrow transplant.

1 Introduction

Linear rank tests play a major role in nonparametric inference. The Chernoff-Savage theorem (1958) ensures the asymptotic normality of these tests, and indeed, for continuous data the asymptotic results work very well. By the time the sample size is around 30, there is very little difference between the asymptotic distribution of a linear rank test statistic and its exact permutational distribution. However this is not the case for categorical data. Here the rate of convergence to asymptotic normality depends on more than just sample size. The number of ties in each category, the group imbalance, and the choice of rank scores, all affect the shape of the permutation distribution in complicated ways, making it difficult to predict a priori whether the asymptotic results for a given data set are reliable. It is important therefore to develop efficient numerical algorithms to supplement existing asymptotic results for the categorical case. These algorithms serve both the data analyst concerned about the validity of the inference in small, sparse, or imbalanced data sets, and the theoretical statistician developing new asymptotic methods and wishing to confirm that the theory is accurate.

This paper develops a very fast algorithm for generating exact permutation distributions for linear rank tests defined on stratified $2 \times c$ contingency tables. The permutational problem is formulated very precisely in Section 2. A network algorithm for solving the problem is presented in Section 3. A major strength of the algorithm is that its limits of computational feasibility increase with the degree of imbalance between the groups being compared. This is precisely where it is needed most, since the reliability of asymptotic results decrease as the imbalance increases. In another paper we analyze some case-control data in which the total sample size is 99,960. Yet, because of the severe imbalance between cases and controls, the asymptotic results differ from the exact ones. The algorithm developed here performs exact permutational inference on the data set with no difficulty whatsoever, despite its enormous sample size.

The inference techniques discussed in this paper are conditional. This is true both for the exact as well as the asymptotic inference. Exact methods for parameter estimation naturally require strong numerical algorithms. But it is not generally recognized that conditional inference places a heavy computational burden on the maximum likelihood estimation as well. A by-product of the algorithmic development in Section 3 is its applicability to the problem of estimating model parameters by maximizing a conditional likelihood function and evaluating its first two derivatives. Without our algorithm, evaluating the conditional likelihood, even though it only yields asymptotic estimates, would be almost as difficult as the

exact inference.

2 Statistical Formulation

In this section we formulate a general permutation problem whose solution will make exact statistical inference possible for a rich class of linear rank tests, defined on ordered categorical or binary data. The computational difficulties encountered with the permutation problem are discussed, setting the stage for the development of an efficient numerical algorithm, in Section 3.

2.1 Tabular Representation of the Data

The data can be represented as a collection of s $2 \times c$ contingency table consisting of 2 rows, c columns, and s strata. A specific collection, or three way table, of this type, denoted by $\underline{\mathbf{x}} \equiv (\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_s)$, is displayed below:

$\mathbf{x}_1 =$

Stratum 1					
Rows	Col_1	Col_2	...	Col_c	Row-Total
Row_1	x_{11}	x_{21}	...	x_{c1}	m_1
Row_2	x'_{11}	x'_{21}	...	x'_{c1}	m'_1
Col-Total	n_{11}	n_{21}	...	n_{c1}	N_1
Col-Score	w_1	w_2	...	w_c	

$\mathbf{x}_2 =$

Stratum 2					
Rows	Col_1	Col_2	...	Col_c	Row-Total
Row_1	x_{12}	x_{22}	...	x_{c2}	m_2
Row_2	x'_{12}	x'_{22}	...	x'_{c2}	m'_2
Col-Total	n_{12}	n_{22}	...	n_{c2}	N_2
Col-Score	w_1	w_2	...	w_c	

$\vdots$

$\mathbf{x}_s =$

Stratum s					
Rows	Col_1	Col_2	...	Col_c	Row-Total
Row_1	x_{1s}	x_{2s}	...	x_{cs}	m_s
Row_2	x'_{1s}	x'_{2s}	...	x'_{cs}	m'_s
Col-Total	n_{1s}	n_{2s}	...	n_{cs}	N_s
Col-Score	w_1	w_2	...	w_c	

The above tabular representation accommodates both the comparison of two multinomial populations and the comparison of k binomial populations. In either case we may adjust for possible covariate effects by stratification. Unstratified data may be regarded as a special case with $s = 1$.

Two Multinomial Populations The two rows of stratum k represent two independent multinomial populations. Each observation falls into one of c ordinal response categories. Thus x_{jk} is the number of stratum k observations, out of a total of m_k, falling into ordered category j for population 1, and x'_{jk} is the number of stratum k observations, out of a total of m'_k, falling into ordered category j for population 2. The stratum invariant scores, $w_1, w_2, \ldots w_c$, are numerical values assigned to the c ordered multinomial response categories.

Several Binomial Populations The c columns of stratum k represent c independent binomial populations with row 1 representing successes and row 2 representing failures. For population j and stratum k there are x_{jk} successes and x'_{jk} failures in n_{jk} independent Bernoulli trials. The stratum invariant scores, $w_1, w_2, \ldots w_c$ typically represent doses, or levels of exposure, affecting the success rates of the c binomial populations.

2.2 Exact Conditional Inference

Define the reference set for the kth stratum, Γ_k, as all possible $2 \times c$ contingency tables whose row and column margins are fixed at the corresponding values of the observed $2 \times c$ table, $\mathbf{x}_k$:

$$\Gamma_k = \{\mathbf{y}_k\text{: } \mathbf{y}_k \text{ is } 2 \times c;\ y_{jk} + y'_{jk} = n_{jk}, \forall j;$$

$$\sum_{j=1}^{c} y_{jk} = m_k, \sum_{j=1}^{c} y'_{jk} = m'_k\} \ .$$

Define the full reference set as the cartesian product of the reference sets across all s strata:

$$\Theta = \Gamma_1 \times \Gamma_2 \times \ldots \times \Gamma_s = \{\underline{\mathbf{y}}\text{: } \mathbf{y}_k \in \Gamma_k, k = 1, 2, \ldots s\ \} \ .$$

The test statistic, T, is defined as a sum of linear rank statistics over the s strata:

$$T = T_1 + T_2 + \ldots + T_s \ ,$$

where each T_k can only take on the values t_k of the form

$$t_k = \sum_{j=1}^{c} w_j y_{jk} \ ,$$

for some $\mathbf{y}_k \in \Gamma_k$, and a fixed set of scores, $w_1, w_2, \ldots w_c$. By a suitable choice of scores one can obtain a very rich class of linear rank tests. The distribution of the test

statistic, T, is derived by limiting the sample space to $\underline{\mathbf{y}} \in \Theta$.

Under the null hypothesis of no row and column interaction the conditional probability distribution of T_k given $\mathbf{y}_k \in \Gamma_k$ is

$$f_k(t_k) = \frac{\sum_{\mathbf{y}_k \in \Gamma_{k,t_k}} \prod_{j=1}^{c} \binom{n_{jk}}{y_{jk}}}{\binom{N_k}{m_k}}, \qquad (2.1)$$

where

$$\Gamma_{k,t_k} = \{\mathbf{y}_k \in \Gamma_k : \sum_{j=1}^{c} w_j y_{jk} = t_k\} .$$

Then by convolution, the conditional probability distribution of T, given $\underline{\mathbf{y}} \in \Theta$, is

$$f(t) = \frac{\sum_{\underline{\mathbf{y}} \in \Theta_t} \prod_{k=1}^{s} \prod_{j=1}^{c} \binom{n_{jk}}{y_{jk}}}{\prod_{k=1}^{s} \binom{N_k}{m_k}}, \qquad (2.2)$$

where

$$\Theta_t = \{\underline{\mathbf{y}} \in \Theta : \sum_{k=1}^{s} \sum_{j=1}^{c} w_j y_{jk} = t\} .$$

Notice that (2.2) is a sum of generalized hypergeometric probabilities and is free of all unknown parameters. This enables us to compute exact p-values for all the linear rank tests listed above. We can also compute the first two moments of T and thereby perform asymptotic inference by appealing to the Chernoff-Savage theorem.

2.3 Parameter Estimation

For data arising from two multinomial distributions or c binomial distributions, we can specify loglinear and logistic models, respectively, for the data generating process. Let π_{jk} be the probability that a subject from stratum k is classified as falling into row 1 and column j. Let π'_{jk} be the probability that a subject from stratum k is classified as falling into row 2 and column j. If the two rows of each stratum represent data from two multinomial populations, the above probabilities must satisfy the constraints

$$\sum_{j=1}^{c} \pi_{jk} = \sum_{j=1}^{c} \pi'_{jk} = 1 ,$$

for $k = 1, 2, \ldots s$. If the c columns of each stratum represent data from c binomial populations, the above probabilities must satisfy the constraints

$$\pi_{jk} + \pi'_{jk} = 1 ,$$

for $j = 1, 2, \ldots c$, and $k = 1, 2, \ldots s$. In either case we assume that there is no three-factor interaction so that the $c - 1$ odds ratios

$$\Psi_j = \frac{\pi_{jk}\pi'_{1k}}{\pi_{1k}\pi'_{jk}} ,$$

$j = 2, 3, \ldots c$, do not depend on k. Next we model these odds ratios as a function of the scores. If the data have been generated from two stratified multinomial populations, it is natural to derive the odds ratios from a loglinear model with a linear by linear row times column association (Agresti, 1990, page 275, equation (8.11)). In the present context the linear by linear model specifies the following expected cell counts on the logarithmic scale:

$$\log(m_k \pi_{jk}) = \alpha_{jk} + \beta w_j$$

for row 1, and

$$\log(m'_k \pi'_{jk}) = \alpha_{jk}$$

for row 2.

If the data have been generated from c stratified binomial populations it is natural to derive the odds ratios from a logistic regression model (Cox, 1970):

$$\log \frac{\pi_{jk}}{\pi'_{jk}} = \alpha_k + \beta w_j .$$

Both models yield the relationship

$$\log \Psi_j = \beta(w_j - w_1) , \qquad (2.3)$$

where β is an unknown parameter to be estimated from the data. It can be shown that T is a sufficient statistic for β under both the linear by linear association model and the logistic regression model. Moreover, the conditional distribution of T, given $(\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_s) \in \Theta$, depends only on β, other (nuisance) parameters being eliminated by the conditioning. This conditional distribution is given by

$$f(t|\beta) = \frac{f(t)\exp(\beta t)}{\sum_u f(u)\exp(\beta u)} , \qquad (2.4)$$

where the denominator of equation (2.4) is simply the normalizing constant obtained by summing over all possible values of T. When $\beta = 0$ we obtain the null distribution (2.2).

The conditional maximum likelihood estimate (cmle) of β is obtained by finding the value of $\hat{\beta}$ that maximizes the conditional probability (2.4) at the observed value $T = a_0$. To obtain the variance of the cmle we need the second derivative of the log likelihood, evaluated at the cmle. Both the cmle and its variance may be rapidly evaluated by repeated backward induction on a network, as discussed in detail in Section 3. We can then use these estimates to perform asymptotic hypothesis tests or compute asymptotic confidence intervals for β.

To obtain an exact confidence interval for β we need the coefficients $f(t)$ for all values of T in the tails its distribution. A network algorithm for this computation is described in Section 3. Once these coefficients have been computed, the conditional tail probabilities, $T \geq a_0$, or $T \leq a_0$, for any value of β, may be derived from equation (2.4). Exact confidence bounds for β are then obtained by inverting corresponding UMP unbiased tests for β, as shown in Cox (1970). For example, a 100(1 - α)% lower confidence bound for β, say $\beta(a_0)$, would be obtained as the solution to

$$\sum_{t=a_0}^{t_{max}} f(t|\beta(a_0)) = \alpha \ . \qquad (2.5)$$

The solution to equation (2.5) may be rapidly evaluated by a simple binary search because, as shown in (2.4), $f(t)$ and β are separable in the expression for $f(t|\beta)$.

2.4 Computational Issues

From the above discussion it is clear that a broad class of exact linear rank tests and parameter estimates can be obtained if we are able to compute truncated distributions of the form

$$\Omega = \{(t, f(t)): \ t \geq a_0\} \ . \qquad (2.6)$$

Exhaustive enumeration of all the tables in Θ for generating Ω would be computationally explosive. Consider the simple case of a single stratum, no ties, and $m = m' = N/2$. The number of tables in the reference set Θ for various values of N is

Sample Size (N)	Tables in Reference Set (Γ)
20	1.8×10^5
30	1.5×10^8
40	1.4×10^{11}
50	1.3×10^{14}
100	1.0×10^{29}

If there were s strata, the size of the corresponding reference set would be raised to the sth power. It is clear that even in the very powerful computing environment available today, explicit enumeration of all the tables in the reference set Θ rapidly becomes computationally infeasible. However much recent research, for example, Mehta et. al. (1984) (1985) (1988), Pagano and Tritchler (1983), Tritchler (1984), Streitberg and Rohmel (1986), and Hollander and Pena (1988), has focused on implicit enumeration of the tables in Θ, thereby considerably extending the size of problem for which exact inference is possible.

Mehta, Patel and Tsiatis (1984), and Mehta, Patel and Wei (1988), developed a network algorithm for implicit enumeration of all the $2 \times c$ contingency tables in the reference set Γ, defined for a single stratum ($s = 1$). Mehta, Patel and Gray (1985) developed a network algorithm for implicit enumeration of s 2×2 contingency tables (where $s > 1$). The present paper generalizes the earlier work to s independent $2 \times c$ contingency tables, a considerably more difficult problem. An alternative method would be to treat the s $2 \times c$ problem as a special case of conditional logistic regression and directly use the exact algorithm of Hirji, Mehta and Patel (1988). However that would not exploit the special structure of the problem in the way that the present algorithm does. We conjecture that the algorithm presented here is the fastest one currently available for categorical data, with unequally spaced w_j scores. In another paper we perform exact inference on some rather large data sets, to illustrate how powerful the algorithm is, and to set up a benchmark against which competing algorithms may be evaluated.

A second contribution of this paper is to provide an efficient numerical algorithm for computing the cmle for β (equation 2.3) and its standard error. A previous algorithm for this problem, in the more general conditional logistic regression setting, was developed by Gail, Lubin, and Rubenstein (1981). Our algorithm is equivalent to theirs for data with no ties, but is considerably more efficient for categorical data. In another paper, we show that the Gail et. al., algorithm, as implemented in the EGRET (1988) software package, is unable to compute conditional maximum likelihood estimates for a large heavily tied data set, whereas our algorithm, obtains the required estimates very rapidly.

3 Numerical Algorithms

We provide numerical algorithms for two problems; generating the truncated permutation distribution Ω, defined by (2.6), and computing the cmle for β, say $\hat{\beta}$, along with its standard error, $\hat{\sigma}$. Both problems are solved within one unified framework wherein the reference set Θ is represented as a network. We will see that processing the network in the forward direction yields Ω, while processing the same network in the backward direction yields $\hat{\beta}$ and its standard error.

3.1 Generating an Overall Truncated Permutation Distribution

Our goal is to generate the truncated permutation distribution Ω for T, the sum of linear rank statistics across all the strata. Our strategy will be to generate s independent stratum specific truncated permutation distributions of the form

$$\Omega_k = \{(t_k, f_k(t_k)) : \ t_k \geq a_k\} \ ,$$

at the cut-off points

$$a_k = a_0 - \sum_{i \neq k} t_{i,max} \ ,$$

for, $k = 1, 2, \ldots s$. Here $t_{k,max}$ is the maximum value of the random variable T_k, and is easily evaluated as part of the backward induction step discussed below. We will perform pairwise convolutions on these stratum specific distributions until the overall distribution is obtained. Thus there are two steps to be performed repeatedly; a distribution generation step, and a convolution step. These steps are described next in separate subsections.

3.1.1 Generating Stratum Specific Truncated Permutation Distributions

Suppose we wish to generate the truncated permutation distribution Ω_k, for the kth stratum. In principle this involves enumerating all the $2 \times c$ contingency tables $\mathbf{y}_k \in \Gamma_k$, computing the value of $t_k = \sum_{j=1}^{c} w_j y_{jk}$ for each one, and summing the hypergeometric probabilities of all the tables $\mathbf{y}_k \in \Gamma_{k,t_k}$, as shown in (2.2). We do this enumeration implicitly rather than explicitly, by representing the reference set Γ_k as a network of nodes and arcs, and then processing the network in a recursive stage-wise fashion.

Network Representation of Γ_k

The network representation of the reference set, Γ_k, is constructed in $c+1$ stages labelled $0, 1, \ldots c$, where stage j corresponds to the jth column of a typical $2 \times c$ table in Γ_k. At stage j there exist a set of nodes of the form (j, m_{jk}), where each $m_{jk} = \sum_{l=1}^{j} y_{lk}$ corresponds to one distinct partial sum of the first j columns of the tables $\mathbf{y}_k \in \Gamma_k$. Arcs emanate from each node (j, m_{jk}) and connect it to successor nodes of the form $(j+1, m_{j+1,k})$. These successor nodes may be specified explicitly as the set

$$\mathbf{R}(j, m_{jk}) = \{(j+1, m_{j+1,k}): \ \max(m_{jk}, m_k - \sum_{l=j+2}^{c} n_{lk})$$

$$\leq m_{j+1,k} \leq \min(m_{jk} + n_{j+1,k}, m_k)\} \ . \qquad (3.7)$$

Starting at stage 0 with initial node $(0,0)$, and applying (3.7) successively to the nodes at stages $1, 2, \ldots c-1$, we automatically end up with the unique terminal node (c, m_k). In this construction each path, or sequence of connected arcs of the form

$$(0,0) \rightarrow (1, m_{1k}) \rightarrow \cdots \rightarrow (c, m_k) \qquad (3.8)$$

corresponds to one and only one table $\mathbf{y}_k \in \Gamma_k$, with $y_{jk} = m_{jk} - m_{j-1,k}$, for $j = 1, 2, \ldots c$. Thus the tables in Γ_k are in one-to-one correspondence with the paths through the network.

To complete the network representation we assign to each arc

$$(j-1, m_{j-1,k}) \rightarrow (j, m_{jk})$$

a **rank length**

$$r_{jk} = w_j(m_{jk} - m_{j-1,k})$$

and a **probability length**

$$p_{jk} = \binom{n_{jk}}{m_{jk} - m_{j-1,k}} \exp(\beta r_{jk}) \ . \qquad (3.9)$$

The rank length of a complete path of the form (3.8) connecting the initial node to the terminal node is defined as the sum of rank lengths of the individual arcs constituting that path. Its probability length is the product of probability lengths of the individual arcs constituting that path. The distribution of T_k is then the same as the distribution of rank lengths of all the paths in Γ_k.

Backward Induction on Γ_k

We can obtain much useful information about the distribution of T_k very quickly, by a single backward pass through the network Γ_k. At any node (j, m_{jk}) define

the sub-network, $\Gamma_k(j, m_{jk})$, to be the set of all possible paths from (j, m_{jk}) to the terminal node (c, m_k). In other words $\Gamma_k(j, m_{jk})$ consists of all possible values of the entries in columns $(j+1, j+2, \ldots, c)$ of the $2 \times c$ contingency tables in Γ_k whose first j columns sum to m_{jk}. Now define the length of the longest path in $\Gamma_k(j, m_{jk})$ by

$$LP(j, m_{jk}) = \max_{\Gamma_k(j,m_{jk})} \{ \sum_{l=j+1}^{c} r_{lk} \} , \tag{3.10}$$

the length of the shortest path in $\Gamma_k(j, m_{jk})$ by

$$SP(j, m_{jk}) = \min_{\Gamma_k(j,m_{jk})} \{ \sum_{l=j+1}^{c} r_{lk} \} , \tag{3.11}$$

and the sum of probability lengths of all the paths in $\Gamma_k(j, m_{jk})$ by

$$TP(j, m_{jk}) = \sum_{\Gamma_k(j,m_{jk})} \prod_{l=j+1}^{c} p_{lk} . \tag{3.12}$$

The values of LP, SP, and TP can be rapidly obtained by backward induction. We illustrate how this is done for LP. Set $LP(c, m_k) = 0$. Now suppose that $LP(j+1, m_{j+1,k})$ is known for every node at stage $j+1$. Move backwards to stage j, select a node $(j, m_{j,k})$, and compute

$$LP(j, m_{jk}) = \max_{\mathbf{R}(j,m_{jk})} \{ r_{j+1,k} + LP(j+1, m_{j+1,k}) \} . \tag{3.13}$$

Repeat this process for every node at stage j and then move back one more stage. Proceeding in this manner we reach stage $(0, 0)$ having evaluated the LP values for all the nodes of the network. The other nodal quantities may be obtained similarly.

Processing Γ_k in the Forward Direction

Starting with the initial node (0,0), we process the network in the forward direction, stage by stage, in such a way that by the time we reach the terminal node, (c, m_k), we will have generated the desired truncated distribution Ω_k. First we introduce some notation. At any node (j, m_{jk}) define the sub-network, $\Upsilon_k(j, m_{jk})$, to be the set of all possible paths from the starting node (0,0) to (j, m_{jk}). In other words, $\Upsilon_k(j, m_{jk})$ consists of all possible values of the entries in columns $(1, 2, \ldots j)$ of the $2 \times c$ contingency tables in Γ_k whose first j columns sum to m_{jk}. (Notice that this set differs from $\Gamma_k(j, m_{jk})$, which specifies the last $c-j+1$ columns of these tables.) Denote a generic path,

$$(0,0) \rightarrow (1, m_{1k}) \rightarrow \cdots \rightarrow (j, m_{jk})$$

in $\Upsilon_k(j, m_{jk})$ by τ. The rank length of τ is

$$r(\tau) = \sum_{l=1}^{j} r_{lk} ,$$

and its probability length is

$$p(\tau) = \prod_{l=1}^{j} p_{lk} .$$

There will typically be several paths, $\tau \in \Upsilon_k(j, m_{jk})$, each having the same rank length, $r(\tau) = u$. Let $c(u)$ be the sum of probability lengths of all these paths. That is,

$$c(u) = \sum_{\{\tau \in \Upsilon_k(j,m_{jk}):\ r(\tau)=u\}} p(\tau) .$$

We now provide a recursive procedure for processing the network in the forward direction. Suppose we have reached stage j of the network in such a way that at each of its nodes, (j, m_{jk}), we are carrying a set of records

$$\Lambda(j, m_{jk}) = \{(u, c(u)):\ u = r(\tau), u + LP(j, m_{j,k}) \geq a_k,$$

$$\tau \in \Upsilon_k(j, m_{jk})\} .$$

The following five-step algorithm is used to update these sets and thereby move forward to stage $j+1$.

Step 1: Select a record $(u, c(u)) \in \Lambda(j, m_{jk})$.

Step 2: Transmit a copy of this record to each successor node $(j+1, m_{j+1,k})$, where the successors are identified by (3.7).

Step 3: At each successor node, $(j+1, m_{j+1,k})$, transform the transmitted record to (u^*, c^*), where $u^* = u + r_{j+1,k}$, and $c^* = c(u) p_{j+1,k}$.

Step 4: Insert (u^*, c^*) into $\Lambda(j+1, m_{j+1,k})$ as follows:

1. If $u^* + LP(j+1, m_{j+1,k}) < a_k$, drop this record from further consideration, and go to Step 5. Otherwise continue with the insertion as described below. (The value of LP is available from the backward induction on Γ_k.)
2. If there already exists a record $(u, c(u)) \in \Lambda(j+1, m_{j+1,k})$ such that $u = u^*$, then merge the two records by replacing $(u, c(u))$ with $(u, c(u) + c^*) \in \Lambda(j+1, m_{j+1,k})$.
3. If no record currently in $\Lambda(j+1, m_{j+1,k})$ has $u = u^*$, then augment $\Lambda(j+1, m_{j+1,k})$ by adding $(u, c(u))$ to it, as a new record.

The technique of hashing (Sedgewick 1983, page 201) is used to search for matches and either merge or augment records in $\Lambda(j+1, m_{j+1,k})$. This ensures an optimum trade-off between efficient use of available memory and fast search.

Step 5: Return to Step 1.

The above 5-step algorithm continues until every record in $\Lambda(j, m_{jk})$ has been processed. Then another node at stage j is selected, and all its records are processed in accordance with the above 5 steps. When all nodes at stage j have been exhausted, repeat Steps 1 through 5 for stage $j+1$. Starting with $\Lambda(0,0) = \{(0,1)\}$ and moving through stages $0, 1, \ldots c-1$ by repeatedly carrying out Steps 1 through 5, we process the entire Γ_k network, ending up at its terminal node with the set of records $\Lambda(c, m_k)$. These records are really the same as the desired truncated probability distribution Ω_k, except that the probability lengths, $c(u)$, have to be normalized by dividing by their sum. That is,

$$f_k(t_k) = \frac{c(t_k)}{\sum_u c(u)} .$$

3.1.2 Pairwise Convolution of the Stratum Specific Truncated Distributions

We restrict our discussion to the convolution of Ω_1 with Ω_2. The resultant distribution may be convolved with Ω_3 in exactly the same manner. We can go on with this pairwise convolution until we obtain Ω.

First sort the records of Ω_1 in ascending order of t_1, and the records of Ω_2 in descending order of t_2. Set $i = 1$, $j = 1$. Now proceed with the following 3-step algorithm:

Step 1: Select record i from Ω_1. Denote it by $(t_1^i, f_1(t_1^i))$. Select record j from Ω_2. Denote it by $(t_2^j, f_2(t_2^j))$.

Step 2: If

$$t_1^i + t_2^j + \sum_{k=3}^{s} t_{k,max} \geq a_0 ,$$

set $j = j+1$, and return to Step 1. But if

$$t_1^i + t_2^j + \sum_{k=3}^{s} t_{k,max} < a_0 , \qquad (3.14)$$

convolve record i from Ω_1 with each of the first $j-1$ records from Ω_2.

Step 3: Set $i = i+1$, and return to Step 1.

There are many ways to perform the convolution at Step 2, if the inequality (3.14) holds. We use hashing to club records having the same value of $t_1 + t_2$. The details are similar to Step 4.2 of the 5-step algorithm for forward processing of Γ_k. A considerable efficiency gain is achieved because we need not consider records from Ω_2 located at positions j or below. The inequality (3.14) ensures that they can never contribute to the final set of records in Ω, since the maximum to which they could be augmented is less than a_0. This is analogous to the record elimination achieved at Step 4.1 of the 5-step algorithm for forward processing of Γ_k.

3.2 Evaluating $\hat{\beta}$ and its Variance

To obtain $\hat{\beta}$, the cmle for β, we must maximize the logarithm of the likelihood (2.4). Then the second derivative of the log likelihood, evaluated at $\hat{\beta}$, yields the desired variance. But direct evaluation of the log likelihood is not an easy task, given the complicated expression for the denominator of (2.4). In fact if one attempted to evaluate this denominator directly, it would require the enumeration of all the s $2 \times c$ tables in Θ. This would make the asymptotic inference as computationally complex as the exact inference. Fortunately there is an easier approach that works well up to extremely large sample sizes. Notice that the denominator of (2.4) is the same as $TP(0,0)$, summed over all the strata. We can easily set up recursions like (3.13) for TP, its first derivative, TP', and its second derivative, TP'', and rapidly evaluate all three quantities during the backward induction

of Γ_k. For example,

$$TP'(j, m_{jk}) = \sum_{\mathbf{R}(j,m_{jk})} p_{j+1,k}[TP(j+1, m_{j+1,k}) + TP'(j+1, m_{j+1,k})]$$

It is easy to show by successive differentiation of the logarithm of (2.4) that the second derivative of the contribution to the log likelihood of the kth stratum is

$$[TP(0,0)]^{-2}[TP'(0,0)]^2 - [TP(0,0)]^{-1}[TP''(0,0)] \qquad (3.15)$$

Evaluating (3.15) at the cmle of β, summing across strata, and equating the resultant second derivative to zero, yields the desired asymptotic variance.

4 Concluding Remarks

The following technical features of the network algorithm were responsible for its extraordinary success:

- The network representation takes advantage of the categorical nature of the data by requiring only as many stages as there are discrete categories.
- The number of nodes in the Γ_k network is determined $\min(m_k, m'_k)$. Thus the greater the imbalance between the two row sums, the smaller the network, and the easier the processing.
- The preliminary backward induction pass through the network provides valuable information about the 'future' for each stage of the forward processing. This enables us to generate a truncated permutation distribution directly at the forward pass, rather than generating the full permutation distribution and then truncating it as needed. In effect, substantially fewer records are carried along at each stage of the forward pass, as records not satisfying the LP criterion get eliminated.
- The network representation enables us to generate the distribution of each T_k recursively in a stagewise forward pass through the network. During this forward pass paths having the same rank length up to some node are 'clubbed' together. We thus deal only with paths having distinct rank lengths up to each node, rather than all the paths up to that particular node.
- The backward induction step enables us to rapidly evaluate the denominator of (2.4), and its first and second derivatives. This greatly facilitates the conditional maximum likelihood inference.

References

Agresti A (1990). *Categorical Data Analysis*, Wiley & Sons, New York.

Chernoff H, Savage IR (1958). Asymptotic normality and efficiency of certain nonparametric test statistics. *Ann Math Stat* 29:972-994.

Cox DR (1970). *The Analysis of Binary Data.* Methuen, London.

Egret (1988). *State of the Art Epidemiological Computing.* Statistics and Epidemiological Research Corporation, Seattle, WA.

Gail MH, Lubin JH, Rubinstein LV (1981). Likelihood calculations for matched case-control studies and survival studies with tied death times. *Biometrika* 68:703-707.

Gail MH, Mantel N (1977). Counting the number of $r \times c$ contingency tables with fixed margins. *JASA* 72:859-862.

Hirji KF, Mehta CR, Patel NR (1988). Exact inference for matched case-control studies. *Biometrics* 44:803-814.

Hollander M, Pena D (1988). Nonparametric tests under restricted treatment assignment rules. *JASA* 83(404):1144-1151.

Landis R, Heyman ER, Koch GG (1978). Average partial association in three-way contingency tables: a review and discussion of alternative tests. *Int. Stat. Rev.* 46:237-254.

Mehta CR, Patel NR, Tsiatis AA (1984). Exact significance testing to establish treatment equivalence for ordered categorical data. *Biometrics* 40:819-825.

Mehta CR, Patel NR, Gray R (1985). On computing an exact confidence interval for the common odds ratio in several 2×2 contingency tables. *JASA* 80(392):969-973.

Mehta CR, Patel NR, Wei LJ (1988). Computing exact significance tests with restricted randomization rules. *Biometrika* 75(2):295-302.

Pagano M, Tritchler D (1983). On obtaining permutation distributions in polynomial time. *JASA* 78:435-441.

Sedgewick R (1983). *Algorithms.* Addison-Wesley, Reading, MA.

StatXact (1991). *A Statistical Package for Exact Nonparametric Inference: Version 2.* Cytel Software Corporation, Cambridge, MA.

Streitberg B, Rohmel R (1986). Exact distributions for permutation and rank tests. *Statistical Software Newsletter* 12:10-17.

Tritchler D (1984). An algorithm for exact logistic regression. *JASA* 79:709-711.

Model Checking for Logistic Regression: A Conditional Approach

BY EDWARD J. BEDRICK
Department of Mathematics and Statistics, University of New Mexico
Albuquerque, New Mexico 87131, U.S.A
AND JOE R. HILL
EDS Research, 5951 Jefferson St. NE, Albuquerque, NM 87109, U.S.A

Abstract

We review the use of exact methods for checking logistic regression models. We focus on global model checks, outlier detection, and goodness-of-link checks. We discuss approximations to exact conditional methods whenever available. We also contrast exact conditional methods and standard unconditional methods based on asymptotic approximations. The techniques are applied to two examples.

1 Introduction

The generalized linear model (McCullagh and Nelder, 1989) provides a unified framework for analyzing binomial response data. An attractive feature of this approach is that a common collection of data analytic and inferential techniques can be used for logistic, probit, complementary log-log, and other possible link functions. Maximum likelihood is usually used to fit generalized linear models (GLIMs). For binomial response models fit within the GLIM paradigm, inferences and model assessments are based on large sample approximations. For example, chi-squared approximations to the deviance and Pearson statistics are used to assess the global fit of the model, while normal approximations to the deviance and Pearson residuals are used to check for outliers.

The logistic regression model has a special place within the class of binomial response models because it is a linear exponential family model. Hence, exact conditional methods for inference are available in contrast to unconditional methods based on maximum likelihood.

The use of conditional distributions for exact inference on logistic parameters dates to Cox (1958, 1970). Hirji, Mehta and Patel (1987) gave an efficient algorithm for computing exact tests of logistic regression parameters. Davison (1988) discussed saddlepoint expansions for approximate conditional inference in logistic regression.

Exact conditional methods can also be used to check the logistic model. The distribution of the data given the observed value of the sufficient statistic for the logistic model serves as the reference distribution for model checks. Once the reference distribution is generated, specific features can be assessed by an appropriate choice of a test statistic. For example, the global fit of the model can be based on a conditional assessment of the deviance or Pearson statistic.

There are compelling reasons for basing model assessments on the distribution of the data given the observed value of the sufficient statistic. First, for any test of model adequacy the conditional distribution is functionally independent of the model parameters. In addition, the conditional approach uses the exact discrete distribution of the data in contrast to methods that assume continuous approximations to this distribution. For small samples or sparse data sets this exactness can be critical. McCullagh (1985, 1986) provided further support for this position.

Bedrick and Hill (1990) developed an algorithm to enumerate the reference distribution for checking logistic models. The enumeration can be computationally intensive, but is feasible for modestly sized data sets.

This paper reviews the use of exact methods for checking logistic models. We focus on global model checks, outlier detection, and goodness-of-link checks. We discuss approximations to exact methods when they are available. We also discuss the corresponding unconditional methods. The rest of this paper is organized as follows. Section 2 introduces notation and develops our approach to model checking. Section 3 discusses exact methods for the three areas of interest. Examples are given in Section 4. Section 5 suggests directions for future research and offers concluding remarks.

2 Background

2.1 Notation

Assume that $Y = (Y_1, ..., Y_n)'$ is a vector of independent binomial random variables with sample sizes $(m_1, ..., m_n)'$ and probability vector $\pi = (\pi_1, ..., \pi_n)'$. The mean vector and covariance matrix of Y are given by $\mu = (\mu_1, ..., \mu_n)'$ and $V = \text{diag}(v_1, ..., v_n)$, respectively, where $\mu_i = m_i\pi_i$ and $v_i = m_i\pi_i(1-\pi_i)$, $i = 1, ..., n$. The logistic regression model can be expressed as

$$\text{logit}(\pi_i) = \log\{\pi_i/(1-\pi_i)\} = z_i'\beta$$

$i = 1, ..., n$, where z_i' is a known $1 \times p$ vector of covariates, and β is a $p \times 1$ vector of unknown regression parameters. Using matrix notation,

$$\text{logit}(\pi) = Z\beta \tag{1}$$

where Z is an $n \times p$ full rank design matrix with i-th row z_i'. Under model (1), $S = Z'Y$ is sufficient for β. Let $\hat{\beta}$ be the maximum likelihood estimator (MLE) of β under model (1). Similar notation is used for other MLEs under model (1); for example $\hat{\mu}$ is the MLE of the mean vector. Finally, set $h_i = \hat{v}_i z_i'(Z'\hat{V}Z)^{-1}z_i$, $i = 1, ..., n$.

The Pearson and deviance statistics on $n-p$ degrees of freedom are given by $X^2 = \sum x_i^2$ and $D = \sum d_i^2$, where $x_i^2 = (Y_i - \hat{\mu}_i)^2/\hat{v}_i$ and

$$d_i^2 = 2[Y_i\log(Y_i/\hat{\mu}_i) + (m_i - Y_i)\log\{(m_i - Y_i)/(m_i - \hat{\mu}_i)\}].$$

2.2 General comments on model checking

The distribution of the data $\text{pr}(Y;\beta)$, indexed by β, can be factored into the marginal distribution of the sufficient statistic S, and the conditional distribution of the data given the sufficient statistic:

$$\text{pr}(Y;\beta) = \text{pr}(Y \mid S)\ \text{pr}(S;\beta).$$

Taking a Fisherian approach (Fisher, 1950), inferences about β are based on $\text{pr}(S;\beta)$, while model checks use $\text{pr}(Y \mid S)$. Letting $s_{obs} = Z'y_{obs}$ be the observed value of the sufficient statistic for the logistic model, the reference distribution for model checking is

$$\text{pr}(Y = y \mid S = s_{obs}) = C_{obs}\prod_{i=1}^{n}\binom{m_i}{y_i} \tag{2}$$

where

$$C_{obs}^{-1} = \sum_{(y_1^*, ..., y_n^*)' \in \mathcal{Y}_{obs}}\ \prod_{i=1}^{n}\binom{m_i}{y_i^*},$$

$$\mathcal{Y}_{obs} = \{y^* = (y_1^*, ..., y_n^*)',\ y_i^* \text{ an integer}:$$
$$0 \le y_i^* \le m_i \text{ and } Z'y^* = s_{obs}\}.$$

Note that $\mathcal{Y}_{obs}$ is the set of response vectors that give the same value of the sufficient statistic as the observed data.

Specific features of interest are assessed by the appropriate choice of a test statistic, say $t(Y)$. Assuming for the moment that large values of $t(Y)$ call into question the adequacy of the model, the significance level associated with the observed value of the test statistic, $t_{obs} = t(y_{obs})$, is given by

$$p(t_{obs}) = \text{pr}\{t(Y) \ge t_{obs} \mid S = s_{obs}\}.$$

The choice of a test statistic need not imply that a particular alternative model is of interest. Indeed, although the statistic created to assess a feature of the model might be motivated by consideration of a particular alternative model, the conclusions drawn from such a test are provisional and do not require acceptance of that alternative. In this regard, the evaluation of $p(t_{obs})$ is a pure significance test. We are not testing formal hypotheses. The computational aspects are, however, identical to those used for exact conditional tests.

Although the reference distribution (2) looks seductively simple, the elements of $\mathcal{Y}_{obs}$ usually must be enumerated to check the model. Depending on the number of samples n and the configuration of covariates, this enumeration can be computationally intensive (Bedrick and Hill, 1990). Once $\mathcal{Y}_{obs}$ is generated, however, implementation of many model checks is routine.

3 Conditional Methods for Model Checking

3.1 Global model checks

A global evaluation of the model can be based on the conditional probability of the data, $q(y) = \text{pr}(Y = y \mid S = s_{obs})$. The p-value for $q(y)$ is the sum of the conditional probabilities for y-vectors that are at least as rare as the observed vector y_{obs}, that is, $p(q_{obs}) = \text{pr}\{q(Y) \le q(y_{obs}) \mid S = s_{obs}\}$. Alternative tests are based on conditional assessments of the deviance and Pearson statistics. P-values for these statistics are $p(D_{obs})$ and $p(X_{obs}^2)$; for example, $p(D_{obs}) = \text{pr}(D \ge D_{obs} \mid S = s_{obs})$. Each vector in $\mathcal{Y}_{obs}$ has the same fitted values for the logistic model (1). Thus, once $\mathcal{Y}_{obs}$ is stored, these three p-values are easily calculated.

McCullagh (1985, 1986) developed Edgeworth approximations to the conditional significance levels for D and

X^2. He derived both a normal approximation and a second-order skewness correction to $p(D_{obs})$ and $p(X^2_{obs})$. The approximations are relatively easy to program. We refer the interested reader to McCullagh's papers for the corresponding formulae. The Edgeworth approximations assume that the number of samples n is large, but they do not require that the sample sizes m_i are large. These approximations are ideal for studies involving many small binomial samples because exact evaluations are often infeasible and the standard unconditional chi-squared approximations to D and X^2 assume that each sample size is large.

McCullagh (1985) conducted a small empirical study of the approximations to $p(X^2_{obs})$ for sparse data problems. He concluded that the normal approximation was inadequate and that the skewness correction gave better results. To the best of our knowledge, there have been no studies of the accuracy of the Edgeworth approximations to $p(D_{obs})$.

For non-replicated binary data (i.e. $m_i = 1$ for all i) the deviance is identical for each observation in $\mathcal{Y}_{obs}$ (McCullagh, 1986). Moreover, each observation in this set has the same probability. Consequently, there is little information in non-replicated binary data concerning the global fit of the model. The diagnostic power of global tests is also likely to be limited when all the sample sizes are very small. In these situations, specific model checks need to be formulated.

3.2 Single degree of freedom checks

Many model checks can be formulated as an exact test on a single logistic regression parameter. To develop these model checks, we consider testing $\gamma = 0$ in the model

$$\text{logit}(\pi) = Z\beta + w\gamma, \tag{3}$$

where w is a known $n \times 1$ vector. The sufficient statistics under model (3) are $R = w'Y$ and $S = Z'Y$. One-sided significance levels for alternatives $\gamma > 0$ and $\gamma < 0$ are given by $p_U(r_{obs}) = \text{pr}(R \geq r_{obs} \mid S = s_{obs})$ and $p_L(r_{obs}) = \text{pr}(R \leq r_{obs} \mid S = s_{obs})$, respectively. A two-sided significance level is often defined to be $2\min\{p_L(r_{obs}), p_U(r_{obs})\}$ (Cox, 1970). Hirji *et al.*'s (1987) algorithm can be used to evaluate these tail probabilities.

To emphasize our earlier comments on significance testing versus hypothesis testing, consider the problem of testing for an increasing trend in the probabilities π_i, $i = 1, ..., n$. The usual small sample test (Gart et al., 1986; p. 85) uses the conditional distribution of $R = \sum_j w_j Y_j$ given $S = \sum_j Y_j$, where $w_1 < \cdots < w_n$ are a somewhat arbitrarily preassigned set of scores. Large values of R suggest an increasing trend. This test is formally equivalent to an upper one-sided exact test of zero slope in the model $\text{logit}(\pi_i) = \alpha + w_i\gamma$, $i = 1, ..., n$. One would likely not accept this as an alternative model if the data suggested that the null model of equal probabilities was implausible.

Davison (1988) derived double saddlepoint approximations to the tail probabilities $p_U(r_{obs})$ and $p_L(r_{obs})$. For simplicity, we will consider the upper tail approximation. Let $Z_w = [Z, w]$ be the full model (3) design matrix, and define $\hat{\gamma}$, D_w, and $\hat{V}_w$ to be, respectively, the MLE of γ, the deviance, and the estimated covariance matrix of Y under this model. The double saddlepoint approximation of $p_U(r_{obs})$ is

$$p_U(r_{obs}) \approx 1 - \Phi(x^*) + \phi(x^*)(1/c^* - 1/x^*), \tag{4}$$

where $x^* = \text{sign}(\hat{\gamma})(D - D_w)^{.5}$ and $c^* = \{1 - \exp(-\hat{\gamma})\}\{\det(Z_w'\hat{V}_w Z_w)/\det(Z'\hat{V}Z)\}^{-.5}$. Here $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard normal distribution function and density function, respectively. The lead term in (4) is the normal approximation to the signed square root of the drop in deviance. This is often used for a large sample test of $\gamma = 0$. Davison also discussed the use of a continuity correction.

Bedrick and Hill (1991) evaluated the accuracy of double saddlepoint approximations to the tail probabilities $p_U(r_{obs})$ and $p_L(r_{obs})$ for several well-known conditional tests. The saddlepoint approximations were extremely accurate, except when the data were sparse or the design matrix was highly unbalanced. Moreover, these approximations were superior to Edgeworth approximations. In our experience, the saddlepoint approximations are sufficient for most assessments.

3.3 Local deviations: Outlier detection

Pregibon (1981, 1982) developed unconditional methods to detect outliers in logistic regression data, assuming a mean slippage model. This outlier model allows a separate mean for a potentially outlying observation. For the moment, we consider methods to detect a single outlier at a designated observation, say j. Letting e_j be the $n \times 1$ indicator variable with j^{th} element equal to one and all other elements equal to zero, the outlier model is

$$\text{logit}(\pi) = Z\beta + e_j\gamma. \tag{5}$$

A test that the jth observation is an outlier is found by testing $\gamma = 0$.

To test $\gamma = 0$, Pregibon suggested the score statistic $t_j^2 = x_j^2/(1 - h_j)$ and the drop in deviance $\Delta_j = D - D_j$, where D_j is the deviance from the outlier model (5). He found a one-step approximation to Δ_j that does

not require estimated probabilities for the outlier model, $\tilde{\Delta}_j = d_j^2/(1-h_j)$. Note that t_j^2 and $\tilde{\Delta}_j$ are, respectively, standardized squared Pearson and deviance residuals.

The usual χ_1^2 approximations for these statistics are questionable for small sample sizes. The inappropriateness of the χ_1^2 approximation for binary data is clear because the residuals have two point distributions (Jennings, 1986).

Bedrick and Hill (1990) discussed several exact conditional tests for a single outlier. Following the development in section (3.2), the statistics Y_j and S are sufficient under the outlier model (5). Thus, significance levels for one-sided alternatives are given by $p_U(y_{j,obs}) = \text{pr}(Y_j \geq y_{j,obs} \mid S = s_{obs})$ and $p_L(y_{j,obs}) = \text{pr}(Y_j \leq y_{j,obs} \mid S = s_{obs})$, while a two-sided significance level is $2\min\{p_L(y_{j,obs}), p_U(y_{j,obs})\}$.

Another test is based on small values of the conditional probability $q_j(y_j) = \text{pr}(Y_j = y_j \mid S = s_{obs})$. The corresponding p-value is $p(q_{j,obs}) = \text{pr}(q_j(Y_j) \leq q_j(y_{j,obs}) \mid S = s_{obs})$. Alternatively, we can use conditional assessments of t_j^2, Δ_j, or $\tilde{\Delta}_j$.

All of the test statistics considered here depend on Y only through Y_j and S. Thus, the reference distribution for each of the tests is

$$\text{pr}(Y_j = y_j \mid S = s_{obs}), \tag{6}$$

which can be derived from the reference distribution (2).

Conflicting inferences from the different statistics are possible because the statistics measure extremeness in Y_j differently. This makes the choice of test statistic an important issue which Bedrick and Hill (1990) addressed in detail. For example, they recommended recentering the statistics t_j^2 and $\tilde{\Delta}_j$ at the conditional mean of Y_j when the reference distribution (6) is multimodal or extremely skewed. In such cases, the recentered statistics, which approximate tests based on the conditional likelihood for γ, behave like the test based on the conditional probability q_j.

We view the outlier test as a check on whether $y_{j,obs}$ is inconsistent with the model, without reference to an alternative. Consequently, we prefer the probability test statistic q_j to the other statistics.

The saddlepoint approximation (4) based on fitting the outlier model (5) gives estimates of $p_U(y_j)$ and $p_L(y_j)$ for all y_j. These estimates can be used to approximate the reference distibution (6), and the exact distribution of each of the test statistics.

When the location of the outlier is unknown Bedrick and Hill (1990) recommended that the minimum p-value (for a given statistic) be used to indicate which case might be an outlier. Evaluating the p-value of this extreme p-value statistic requires the entire conditional distribution (2). In addition, they suggested using a plot of the ordered p-values versus their conditional expected values together with upper and lower bounds. The p-value plot is a natural analog to standard Q-Q plots. They also discussed the problem of detecting multiple outliers.

3.4 Goodness-of-link tests

The appropriateness of the logistic link function can be assessed in several ways. For simplicity, suppose that we are interested in checking whether a specific alternative link function, say the probit, provides a better fit to the data. Assume that the same covariates are used with both links. Let $\hat{\mu}^A$ and D_A be the estimated mean vector and the deviance under the alternative link. The discrepency between the two fitted models can be measured by the difference in deviances: $\Delta_A = D - D_A$. This test statistic is minus twice the (unconditional) log-likelihood ratio statistic for comparing non-nested models. Large positive values of Δ_A suggest a departure from the logistic model in the direction of the alternative link. A conditional assessment of Δ_A requires that $\hat{\mu}^A$ be computed for each vector in $\mathcal{Y}_{obs}$.

The comparison of non-nested models was initially studied by Cox (1961), who developed large sample unconditional tests based on the likelihood ratio. Wahrendorf, Becher, and Brown (1987) proposed the difference in deviances for comparing non-nested generalized linear models. They assessed the significance of the difference in deviances unconditionally, using nonparametric bootstrap samples.

An alternative approach is to imbed the logistic model within a parametric family of link functions. Davison (1988) showed how the saddlepoint approximation could be used to assess the adequacy of the logistic link within this framework. We refer the interested reader to his paper for details.

4 Examples

The first example examines the accuracy of approximations to exact conditional methods. The second example illustrates a goodness-of-link check. We used Bedrick and Hill's (1990) algorithm to generate $\mathcal{Y}_{obs}$ for these examples.

4.1 Nodal involvement data

Brown (1980) discussed an experiment where 53 prostate cancer patients underwent surgery to examine their lymph nodes for evidence of cancer. The data were used

to develop a model for predicting nodal involvement (1 = evidence of cancer, 0 = no evidence) from 5 preoperative binary prognostic variables. The data are given in Table 1; see Bedrick and Hill (1990) for a description of the covariates. All of the 23 samples are small, so asymptotic theory does not apply to residuals or outlier tests.

Table 1: Nodal involvement data with designated case test statistics and conditional p-values. The covariates $z_{j1}, ..., z_{j5}$ are given as a binary string of length 5.

j	y_j/m_j	z_{jk}	t_j^2	$p(t_j^2)$	$\tilde{\Delta}_j$	$p(\tilde{\Delta}_j)$
1	5/6	01111	1.54	0.39	1.11	1.00
2	1/6	00001	0.08	1.00	0.08	1.00
3	0/4	11100	2.81	0.39	4.86	0.21
4	2/4	11001	0.22	1.00	0.22	1.00
5	0/4	00000	0.22	1.00	0.43	1.00
6	2/3	01101	0.03	1.00	0.03	1.00
7	1/3	11000	1.73	0.35	1.24	0.35
8	0/3	10001	0.75	1.00	1.37	0.61
9	0/3	10000	0.11	1.00	0.22	1.00
10	0/2	10010	0.58	1.00	1.06	1.00
11	1/2	01001	0.00	1.00	0.00	1.00
12	1/2	00100	4.44	0.18	2.54	0.18
13	1/1	11111	0.10	1.00	0.20	1.00
14	1/1	11011	0.27	1.00	0.48	1.00
15	1/1	10111	0.53	1.00	0.90	1.00
16	1/1	10011	1.15	1.00	1.61	1.00
17	0/1	10100	0.09	1.00	0.17	1.00
18	1/1	01110	0.47	1.00	0.80	1.00
19	0/1	01100	0.52	1.00	0.86	1.00
20	1/1	01010	1.36	1.00	1.94	1.00
21	1/1	00101	2.14	0.35	2.51	0.35
22	0/1	00011	1.83	0.41	2.24	0.41
23	0/1	00010	0.34	1.00	0.60	1.00

Brown proposed a main effects model for the log-odds of nodal involvement:

$$\text{logit}(\pi_i) = \beta_0 + \beta_1 z_{i1} + \beta_2 z_{i2} + \beta_3 z_{i3} + \beta_4 z_{i4} + \beta_5 z_{i5},$$

$i = 1, ..., 23$. A FORTRAN version of our algorithm generated the 6034 response vectors belonging to $\mathcal{Y}_{obs}$ in 13 seconds on a SUN SPARC station IPC computer.

The deviance and Pearson statistics for this model are $D_{obs} = 18.07$ and $X^2_{obs} = 15.46$ on 17 degrees of freedom. The exact conditional p-values for $q(y_{obs})$, D_{obs}, and X^2_{obs} are 0.758, 0.803, and 0.792, respectively, suggesting that the model provides an adequate global fit to the data. The Edgeworth approximations to the conditional p-values for D_{obs} and X^2_{obs} lead to the same conclusions. The first-order normal approximations to $p(D_{obs})$ and $p(X^2_{obs})$ are 0.937 and 0.912, while the second-order skewness adjusted approximations are 0.846 and 0.828. The second-order approximations are reasonably accurate.

Table 1 gives the score statistic t_j^2, and its exact conditional p-value $p(t_j^2)$, for each observation. Summaries for $\tilde{\Delta}_j$ are also provided. We note that $p(t_j^2) = p(q_j)$ for each observation. None of the observations appear to be unusual. Note the consistency in the p-values across statistics, even when the magnitudes of t_j^2 and $\tilde{\Delta}_j$ are very different. Re-centering t_j^2 and $\tilde{\Delta}_j$ at the conditional expectation of Y_j had little effect on the conditional p-values.

The saddlepoint approximation to the marginal conditional distributions $\text{pr}(Y_j = y_j \mid S = s_{obs})$ were very accurate. The relative error in the continuity corrected estimates averaged 3% across the samples. Moreover, the approximations to the significance levels for t_j^2, $\tilde{\Delta}_j$, and q_j were always within 0.01 of the exact values given in Table 1.

The exact conditional and approximate conditional assessments indicate that the logistic model provides an adequate global and local fit to the data.

4.2 A dose-response experiment

Table 2 gives data from an experiment designed to examine the toxicity of a pesticide to a species of Chrysanthemum aphis (Finney, 1947; p. 69). We initially considered a logistic model with log-dose as the predictor. The deviance and Pearson statistics are $D_{obs} = 5.96$ and $X^2_{obs} = 5.88$ on 4 degrees of freedom. The asymptotic p-values for these statistics based on a χ^2_4 approximation are both about 0.20.

Table 2: Dose-response data (Finney, 1947; p. 69) with expected counts for logistic ($\hat{\mu}$) and complementary log-log ($\hat{\mu}^A$) links.

j	m_j	y_j	Log-dose	$\hat{\mu}_j$	$\hat{\mu}_j^A$
1	47	7	0.40	7.18	8.64
2	46	22	0.71	18.47	17.36
3	46	27	1.00	32.02	29.85
4	48	38	1.18	39.88	39.33
5	46	43	1.31	41.17	41.99
6	50	48	1.40	46.29	47.80

The expected cell counts under the logistic model are given in Table 2. Although the observed count at the

third dose level appears to be inconsistent with the fitted model, the discrepency is not significant.

Morgan (1985) suggested several alternative models for these data. We fit several models, of which the complemetary log-log link provided the best fit. The deviance and Pearson statistics for this model are 3.66 and 3.69 with asymptotic p-values 0.455 and 0.449, respectively. The expected cell counts under this model are given in Table 2. None of the observations is poorly fit. In comparison, the complementary log-log provides a better fit than the logistic at large doses, but a somewhat poorer fit at the low doses.

Although the sample sizes are large, an exact analysis of the logistic model is feasible. Our FORTRAN routine generated the 1496 response vectors belonging to $\mathcal{Y}_{obs}$ in 39 seconds on the IPC. The exact p-values for D_{obs}, X^2_{obs}, and $q(y_{obs})$ are 0.383, 0.343, and 0.447, respectively. The exact p-values for D_{obs} and X^2_{obs} are approximately twice their unconditional p-values. A conditional assessment indicates that each observation is adequately fit by the model.

To illustrate the goodness-of-link check, we evaluated the exact distribution of Δ_A using the complementary log-log as the alternative link function. The difference between the observed logistic and complementary log-log deviances was 2.30, which corresponds to the 91.6^{th} percentile of this distribution. Thus, the data provide some indication of a departure from the logistic model in the direction of the complementary log-log. As noted, the two models give different fits at the extreme doses. Such differences are an important consideration in model selection when the extreme percentiles of a tolerance distribution are the primary interest. The data suggest that this issue should be explored more completely before selecting the logistic model for inference.

5 Discussion: Potential for extending current methods

The model checks we described in this paper are but a small subset of the methods for which exact analyses are theoretically possible. For example, specific methods are needed for non-replicated binary data because of the extreme discreteness of the response. Landwehr, Pregibon, and Shoemaker (1984) and Fowlkes (1987) developed diagnostic tools for binary data. Landwehr *et al.*'s local mean deviance plot assesses the local fit of the logistic model to observations with similar covariate values. Discussants of this article suggested variations of the local mean deviance which either group observations with similar predicted probabilities or use analogs of linear regression lack-of-fit tests. Regardless of how the data are grouped, this is a fruitful approach because the deviance provides no information about the global fit of the model to non-replicated binary data. Fowlkes's diagnostics are based on smoothing the binary responses to examine the underlying structure. In theory, all of the variations of the local mean deviance plot and Fowlkes's "smoothed$-\chi^2$" components can be calibrated conditionally. Unfortunately, the size of the problem for which their methods are most effective are beyond the capability of our current algorithms.

The present infeasiblity of enumerating $\mathcal{Y}_{obs}$ for large data sets is not the only limiting factor with exact methods. We generated $\mathcal{Y}_{obs}$ for a study with 29 samples of size two and four binary covariates, only to find that $\mathcal{Y}_{obs}$ contained over 285 million response vectors. Given the size of $\mathcal{Y}_{obs}$, certain exact evaluations were infeasible so we based our assessments on a sample of responses from $\mathcal{Y}_{obs}$.

The implementation of conditional methods for model checking would be greatly enhanced by the development of an efficient algorithm to simulate from the reference distribution (2). In the problem discussed just above, we had to generate $\mathcal{Y}_{obs}$ *prior to* sampling. Several algorithms are available for randomly sampling $2 \times n$ contingency tables with fixed margins, without first generating the population of tables. Note that the set of $2 \times n$ contingency tables with fixed margins is equivalent to the reference set for checking a logistic model with an intercept term only. The introduction of covariates to the model imposes additional constraints on the tables. This added structure makes it difficult to project whether responses from $\mathcal{Y}_{obs}$ can be randomly generated without first enumerating this set. The problem merits serious consideration.

To close, we believe that exact methods should play an important role in future analyses of logistic regression models. We optimistically project that advances in this area will continue due to an increased interest in computationally intensive methods coupled with the continual development of more powerful computing algorithms and environments.

6 REFERENCES

Bedrick, E. J. and Hill, J. R. (1990). Outlier tests for logistic regression: A conditional approach. *Biometrika*, 77, 815 - 827.

Bedrick, E. J. and Hill, J. R. (1991). An empirical assessment of saddlepoint approximations for testing a logistic regression parameter. To appear in *Biometrics*.

Brown, B. W. (1980). Prediction analysis for binary data. In *Biostatistics Casebook* (eds. R. G. Miller, Jr., B. Efron, B. W. Brown, Jr., and L. E. Moses), 3 - 18. John Wiley and Sons, New York.

Cox, D. R. (1958). The regression analysis of binary sequences (with discussion). *J. R. Statist. Soc.* B, 20, 215 - 232.

Cox, D. R. (1961). Tests of separate families of hypotheses. *Proc. Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 105 - 123.

Cox, D. R. (1970). *Analysis of Binary Data.* Methuen, London.

Davison, A. C. (1988). Approximate conditional inference in generalized linear models. *J. R. Statist. Soc.* B, 50, 445 - 461.

Finney, D. J. (1947). *Probit Analysis, first edition.* Cambridge University Press, Cambridge.

Fisher, R. A. (1950). The significance of deviations from expectation in a Poisson series. *Biometrics*, 6, 17 - 24.

Fowlkes, E. B. (1987). Some diagnostics for binary logistic regression via smoothing. *Biometrika*, 74, 503 - 516.

Gart, J. J., Krewski, D., Lee, P. N., Tarone, R. E. and Wahrendorf, J. (1986). *Statistical Methods in Cancer Research. Volume III. The Design and Analysis of Long-Term Animal Experiments. IARC Scientific Publication No. 79.* International Agency for Research on Cancer, Lyon.

Hirji, K. F., Mehta, C. R., and Patel, N. R. (1987). Computing distributions for exact logistic regression. *J. Am. Statist. Assoc.*, 82, 1110 - 1117.

Jennings, D. E. (1986). Outliers and residual distributions in logistic regression. *J. Am. Statist. Assoc.*, 81, 987 - 990.

Landwehr, J. M., Pregibon, D., and Shoemaker, A. C. (1984). Graphical methods for assessing logistic regression models (with discussion). *J. Am. Statist. Assoc.*, 79, 61 - 83.

McCullagh, P. (1985). On the asymptotic distribution of Pearson's statistic in linear exponential family models. *Inter. Statist. Review*, 53, 61 - 67.

McCullagh, P. (1986). The conditional distribution of goodness-of-fit statistics for discrete data. *J. Am. Statist. Assoc.*, 81, 104 - 107.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models, second edition.* Chapman and Hall, New York.

Morgan, B. J. T. (1985). The cubic logistic model for quantal assay data. *Appl. Statist.*, 34, 105 - 113.

Pregibon, D. (1981). Logistic regression diagnostics. *Ann. Statist*, 9, 705 - 724.

Pregibon, D. (1982). Score tests in GLIM with applications. In *Lecture Notes in Statistics, No. 14, GLIM 82: Proceedings of the International Conference on Generalized Linear Models* (ed. R. Gilchrist), Springer-Verlag, New York.

Wahrendorf, J., Becher, H., and Brown, C. C. (1987). Bootstrap comparison of non-nested generalized linear models: applications in survival analysis and epidemiology. *Applied Statistics*, 36, 72 - 81.

Using Gibbs Sampling for Bayesian Inference in Multidimensional Contingency Tables

Leonardo D. Epstein
Department of Biostatistics
The Johns Hopkins University
Baltimore, Md. 21205

Stephen E. Fienberg
Department of Statistics
Carnegie Mellon University
Pittsburgh, Pa. 15213

Abstract

This paper discusses a method suggested by Epstein and Fienberg (1991) for the Bayesian analysis of multidimensional contingency tables in connection with the Gibbs sampler to calculate posterior densities.

The method consists of a two-stage hierarchical prior. The first stage is a Dirichlet distribution with a loglinear reparametrization for its means. The second stage is a multivariate normal distribution on the loglinear parameters. However, other distributions can be used if the Dirichlet-normal combination is not flexible enough to accomodate one's prior beliefs.

These prior distributions are useful when one believes, with uncertainty, in a given loglinear structure for the cell probabilities.

Key words: Contingency tables; Bayesian estimation; Dirichlet prior distribution; Gibbs sampler; Loglinear model; Maximum likelihood estimation of Dirichlet distributions.

1 Introduction

A new Bayesian method for the analysis of multidimensional contingency tables was recently proposed by Epstein and Fienberg (1988) and Epstein (1990). As with many other Bayesian methods, ours uses the posterior means of the cell probabilities to estimate these parameters. The focus on posterior means is in part due to the importance of point estimation and in part due to computational difficulties in drawing further inferences from the posterior. The purpose of this article is to illustrate with an example how to use the Gibbs sampler to compute estimates of the posterior densities that arise from our method. These density estimates are readily integrable to compute posterior probabilities and moments.

We introduce the esentials of the method via a simple example. Suppose our interest is on inferences about the array of cell probabilities $\boldsymbol{\theta} = \{\theta_{ij}\}$ of a 2×2 contingency table and suppose also that given $\boldsymbol{\theta}$, the observed counts $\boldsymbol{x} = \{x_{ij}\}$ follow a multinomial distribution $M(N, \boldsymbol{\theta})$. We label the two factors by 1 and 2.

When the data follow a multinomial distribution to model prior beliefs it is common to use the conjugate Dirichlet prior $D(K, \boldsymbol{\eta})$ with density

$$[\boldsymbol{\theta}|K, \boldsymbol{\eta}] = \beta \prod_{ij} \theta_{ij}^{K\eta_{ij}-1},$$

where $\beta = \Gamma(K)/\prod_{ij}\Gamma(K\eta_{ij})$, and $\boldsymbol{\eta} = \{\eta_{ij}\}$.

Before the observation of $\boldsymbol{x}$ we might believe with some uncertainty that the two factors are independent. That is, we might believe that $\boldsymbol{\theta}$ satisfies $\theta_{ij} = \theta_{i+}\theta_{+j}, i = 1,2, j = 1,2$, with some degree of uncertainty.

The condition $\theta_{ij} = \theta_{i+}\theta_{+j}$ is equivalent to

$$\log \theta_{ij} = u + u_{1(ij)} + u_{2(ij)}. \tag{1}$$

with the restriction that the term u in this equation is

$$u = -\log(\sum_{i,j} \exp(u_{1(ij)} + u_{2(ij)}), \tag{2}$$

so that $\sum_{i,j}\theta_{ij} = 1$. This normalization leads to an equivalent parametrization that uses the multivariate logits, i.e., if

$$\theta_{ij} = \frac{e^{\gamma_{ij}}}{\sum_{ij} e^{\gamma_{ij}}},$$

then the γ_{ij} are the multivariate logits (see Leonard and Novick, 1976). The parametrization (1) and the normalizing condition on u are equivalent to reparametrizing $\{\gamma_{ij}\}$ using

$$\gamma_{ij} = u_{1(ij)} + u_{2(ij)}.$$

Unless neccessary, the remainder of this paper omits explicit reference to the normalizing role of u. Thus, we will simply speak of the loglinear parametrization $\log \theta_{ij} = u + u_{1(ij)} + u_{2(ij)}$.

To see that the parametrization (1) is equivalent to independence, substitute the value of u back in equation (1) to get

$$\theta_{ij} = \frac{e^{u_{1(ij)}}}{\sum_i e^{u_{1(ij)}}} \times \frac{e^{u_{2(ij)}}}{\sum_i e^{u_{2(ij)}}}.$$

Hence

$$\theta_{i+} = \frac{e^{u_{1(ij)}}}{\sum_i e^{u_{1(ij)}}} \text{ and } \theta_{j+} = \frac{e^{u_{2(ij)}}}{\sum_j e^{u_{1(ij)}}}.$$

To incorporate in the prior our uncertain belief in independence, Epstein (1990) and Epstein and Fienberg (1991) proposed using a loglinear parametrization on the Dirichlet means. That is, to reflect the plausibility that the cell probabilities satisfy (1) they suggest using

$$\log \eta_{ij} = u + u_{1(ij)} + u_{2(ij)}, \tag{3}$$

with $u = -\log(\sum_{i,j} \exp(u_{1(ij)} + u_{2(ij)})$.

The index "1" in $u_{1(ij)}$ indicates that this u-term depends only on the index i. It is more common to omit the indices on which the u-terms do not depend. Thus often we write $u_{1(i)}$ and $u_{2(j)}$ instead of $u_{1(ij)}$ and $u_{2(ij)}$, but the fact that $\{u_{1(ij)}\}$ and $\{u_{2(ij)}\}$ are arrays of the same dimensions as $\{\eta_{ij}\}$ simplifies many formulas.

To establish the connection with the multidimensional case, we note that parametrization (3) maps an array $\{\gamma_{ij}\}$ belonging to the linear subspace $M = \{\{\gamma_{ij}\} : \gamma_{ij} = u_{1(i)} + u_{2(j)}\}$ into the array $\{\exp(\gamma_{ij})/\sum_{ij} \exp(\gamma_{ij})\}$.

The parametrization (3) implies that

$$\eta_{i+} = \frac{e^{u_{1(ij)}}}{\sum_i e^{u_{1(ij)}}} \text{ and } \eta_{+j} = \frac{e^{u_{2(ij)}}}{\sum_j e^{u_{1(ij)}}}. \tag{4}$$

Thus, $\{u_{1(ij)}\}$ and $\{u_{2(ij)}\}$ parametrize the marginal arrays $\{\eta_{i+}\}$ and $\{\eta_{+j}\}$, respectively.

We follow the notation of Andersen (1974) to represent marginal tables, and the definition will be recalled in section 2 more formally. This notation represents the marginal array with entries η_{i+} by $\eta^Y = \{\eta^Y_{ij}\}$, where $Y = \{1\}$. The set of factor labels Y indicates that η^Y depends only on the index corresponding to factor 1, namely i, and that η was collapsed over the indices corresponding to the factors not in Y, namely j. We will also use products of arrays. Thus, for example, the product of $\eta^{\{1\}}$ and $\eta^{\{2\}}$, denoted by $\eta^{\{1\}}\eta^{\{2\}}$, is the array whose (i,j) entry is $\eta^{\{1\}}_{ij}\eta^{\{2\}}_{ij}$, or, in the usual notation, $\eta_{i+}\eta_{+j}$.

The parametrization $\log \eta_{ij} = u + u_{1(ij)} + u_{2(ij)}$ is equivalent to $\eta_{ij} = \eta^{\{1\}}_{ij}\eta^{\{2\}}_{ij}$ with $0 \le \eta^{\{1\}}_{ij} \le 1$, $0 \le \eta^{\{2\}}_{ij} \le 1$, $\eta^{\{1\}}_{1j} + \eta^{\{1\}}_{2j} = 1$, and $\eta^{\{2\}}_{i1} + \eta^{\{2\}}_{i2} = 1$ (see Albert and Gupta, 1982). However, the loglinear parametrization on the Dirichlet means allowed Epstein and Fienberg (1991) and Epstein (1990) to extend the method to multidimensional tables.

If we feel we cannot specify a value for $\eta^{\{1\}}_{ij}$ and $\eta^{\{2\}}_{ij}$, or, equivalently, for $u_{1(ij)}$ and $u_{2(ij)}$, then the Dirichlet distribution cannot adequately represent our prior beliefs. However, as Albert and Gupta (1982) point out, the Dirichlet distribution may still be used as the first stage of a two-stage prior. With a loglinear parametrization for the Dirichlet means there are two equivalent alternative ways to complete the two-stage prior. One may use distributions on the u-terms or one may prefer to specify distributions on $\eta^{\{1\}}_{ij}$ and $\eta^{\{2\}}_{ij}$ directly.

The loglinear parametrization will be more useful when analyzing tables of higher dimensions where one may consider more complex loglinear structures. As the next section explains, with loglinear parametrizations for n-way tables one can also specify the second-stage in two alternative ways, but to use the second one must determine the generating class of the loglinear parametrization and use the margins of η given by the generator as parameters of the Dirichlet distribution.

The parameter K governs the concentration of the prior distribution about the independence surface

$$\begin{aligned} \mathcal{S} \;=\; & \{\{\theta_{ij}\} | \theta_{ij} = \theta^{\{1\}}_{ij}\theta^{\{2\}}_{ij}, \\ & 0 \le \theta^{\{1\}}_{ij} \le 1, 0 \le \theta^{\{2\}}_{ij} \le 1, \\ & \theta^{\{1\}}_{11} + \theta^{\{1\}}_{12} = 1, \theta^{\{2\}}_{21} + \theta^{\{2\}}_{22} = 1\}. \end{aligned}$$

In the limit, as $K \to \infty$, the prior, and therefore the posterior, concentrate all of their mass on $\mathcal{S}$.

When we use a two-stage prior we obtain the posterior means

$$\varepsilon(\theta_{ij}|\boldsymbol{x}) = \frac{N}{N+K}\frac{x_{ij}}{N} + \frac{K}{N+K}\varepsilon(\eta^{\{1\}}_{ij}\eta^{\{2\}}_{ij}|\boldsymbol{x}), \tag{5}$$

which we use to estimate θ. The expectation $\varepsilon(\eta^{\{1\}}_{ij}\eta^{\{2\}}_{ij}|\boldsymbol{x})$ is with respect to the distribution induced on $\eta^{\{1\}}$ and $\eta^{\{2\}}$ through equations (4).

In most practical situations, when $K \to 0$ the posterior means $\varepsilon(\theta_{ij}|\boldsymbol{x})$ converge to the observed proportions x_{ij}/N. When $K \to \infty$ not only the posterior distribution concentrates the all of its mass on $\mathcal{S}$, but the posterior mean $\varepsilon(\theta|\boldsymbol{x})$ itself belongs to $\mathcal{S}$. This property translates into

$$\lim_{K\to 0} \varepsilon(\theta_{ij}|\boldsymbol{x}) = \lim_{K\to 0} \varepsilon(\eta^{\{1\}}_{ij}|\boldsymbol{x}) \times \lim_{K\to 0} \varepsilon(\eta^{\{2\}}_{ij}|\boldsymbol{x}).$$

It shows that the estimates corresponding to increasing values of K reflect an increasingly strong prior belief in the plausibility of independence of the two factors by

compromising between estimates obtained under a saturated model and estimates obtained under an independence model. Epstein (1990) showed that this property holds for general loglinear parametrizations.

With this introductory example it is now easy to see how our approach extends to tables of higher dimension. If we believe, with uncertainty, in a given loglinear structure for the cell probabilities, we use a two-stage prior. In the first stage use a Dirichlet distribution with means having the same loglinear structure. In the second stage use distributions, Gaussian for example, on the u-terms of the loglinear parametrization.

In the introductory example we speak of independence being a plausible structure for the cell probabilities to indicate that we believe in independence only to a certain degree. In general, we will speak of a plausible loglinear structure to indicate that we believe in that structure only to a certain degree.

In the multidimensional case, as $K \to \infty$, the prior and therefore the posterior concentrate all their mass in the subset of arrays η defined by the loglinear parametrization. Epstein (1990) studied properties of the posterior means as estimators when the loglinear parametrization on η is hierarchical.

The next section reviews the extension of the method for multidimensional tables and the basic elements of loglinear parametrizations.

Section 3 presents our implementation of the Gibbs sampler. The implementation requires finding maximum likelihood estimates for Dirichlet means under a loglinear parametrization. Subsection 3.1 describes the use of the projection gradient method to compute these maximum likelihood estimates. Additionally, section 3 discusses a rejection-acceptance scheme to draw deviates from a posterior distribution that does not require the marginal (predictive) distribution. Section 4 illustrates the implementation of the Gibbs sampler and the method of Epstein and Fienberg (1991) with simple sociological example concerning student politics and family structure.

2 A Bayesian Method for Multidimensional Tables

In this section we review the method proposed by Epstein (1990) and Epstein and Fienberg (1991) for multidimensional tables. We refer the reader to Epstein (1990) for proofs and a detailed discussion of this section's results.

Following the notation of Andersen (1974), consider n factors or treatments labeled $1, 2, \ldots, n$, with factor i having r_i levels. Define $\bar{r}_i = \{1, \ldots, r_i\}$ and call it the set of levels of factor i. The set $I = \bar{r}_1 \times \cdots \times \bar{r}_n$, is usually refered to as the index set or the set of cells.

A selection of levels $\iota = (i_1, i_2, \ldots, i_n)$, a generic element in I, is often referred to as the $(i_1, i_2, \ldots, i_n)$-cell. One obtains a $r_1 \times \cdots \times r_n$ contingency table $\boldsymbol{x} = \{x_\iota, \iota \in I\}$ when N individuals are examined and cross-classified according to the levels of each of the factors.

We shall assume that $\boldsymbol{x} = \{x_\iota, \iota \in I\}$ has a multinomial $M(N, \{\theta_\iota\})$ distribution, where θ_ι is the probability of an individual being classified in cell ι. However, the method easily adapts to other sampling distributions, such as Poisson and product multinomial (Bishop, Fienberg, and Holland, 1975).

In the first stage use a Dirichlet $D(K, \eta)$ distribution with density

$$[\theta|K, \eta] = \beta \prod_{\iota \in I} \theta_\iota^{K\eta_\iota - 1}, \tag{6}$$

indexed by $\eta = \{\eta_\iota, \iota \in I\}$, and where $\beta = \frac{\Gamma(K)}{\prod_{\iota \in I} \Gamma(K\eta_\iota)}$. The parameter $K > 0$ is prespecified. Thus, $\varepsilon(\theta_\iota|K, \eta) = \eta_\iota$ for $\iota \in I$.

Let $w \subset \bar{n}$, i.e., w is a set of factor labels. We shall denote $\boldsymbol{u}_w$ the interaction parameter among the factors in w. More specifically, the interaction $\boldsymbol{u}_w$ is the $r_1 \times \ldots \times r_n$ array

$$\boldsymbol{u}_w = \{u_{w(i_1, \ldots, i_n)}\},$$

where the entries $u_{w(i_1, \ldots, i_n)}$ of $\boldsymbol{u}_w$ depend only upon the indices i_j with $j \in w$. Often the interactions are taken to satisfy the usual ANOVA constraints, i.e., the sum of the entries $u_{w(i_1, \ldots, i_n)}$ over the levels of any factor $j \notin w$ is zero. These constraints achieve identifiability of the parametrization. The Bayesian approach does not require identifiable parametrizations and therefore we need not use constraints. Their use, however, is not precluded. One should use them whenever they facilitate producing a prior distribution reflecting one's beliefs.

Loglinear parametrizations are usually used for the multinomial parameters. The model defined by

$$\log \boldsymbol{\theta} = \sum_{w \subset \bar{n}} \boldsymbol{u}_w, \tag{7}$$

is the saturated or unrestricted model. Whenever a vector, $\boldsymbol{x}$ say, appears as the argument of a real function of one variable, f say, then $f(\boldsymbol{x})$ shall stand for the vector $(f(x_1), \ldots, f(x_l))^t$.

The entries of the array $\boldsymbol{u}_\emptyset$, where $\emptyset$ is the empty set, are all the same. The term $\boldsymbol{u}_\emptyset$ is usually referred to as the constant term.

In the general case we can use the multivariate logits γ_ι by writing:

$$\theta_\iota = \frac{e^{\gamma_\iota}}{\sum_{\iota \in I} e^{\gamma_\iota}}.$$

The parametrization (7) is equivalent to

$$\gamma = \sum_{w \subset \bar{n}, w \neq \emptyset} u_w.$$

We obtain submodels by including only some interactions in the formula above. To specify which interactions we include in a submodel, we use a class of subsets of $\bar{n}$ which we call $\mathcal{A}$. For example, we write

$$\log \theta = \sum_{w \in \mathcal{A}} u_w.$$

to specify a parametrization which only includes the interactions among factors in w, with $w \in \mathcal{A}$.

We are concerned with making inferences when we feel it is plausible that

$$\log \theta = \sum_{w \in \mathcal{A}} u_w, \tag{8}$$

where $\mathcal{A}$ is a strict subset of $\bar{n}$. To incorporate this belief into the prior we suggest that instead of using the loglinear parametrization on θ we use it on the Dirichlet means, that is,

$$\log \eta = \sum_{w \in \mathcal{A}} u_w. \tag{9}$$

This restricts $\log \eta$ to lie in a linear subspace M of $\Re^I$. To ensure that the parametrization is such that $\sum_{\iota \in I} \eta_\iota = 1$, it is necessary to assume that M contains the array **1** whose entries are all 1. In the introductory example the class $\mathcal{A}$ is $\{\{1\}, \{2\}\}$ and therefore equation (9) becomes

$$\log \eta_{ij} = u + u_{1(ij)} + u_{2(ij)}.$$

In the parametrization (9) the term $u_\emptyset = \{u\}$ must satisfy

$$u = -\log\Big(\sum_{\iota \in I} \exp\Big(\sum_{w \in \mathcal{A}, w \neq \emptyset} u_{w(\iota)}\Big)\Big),$$

so that $\sum_{\iota \in I} \eta_\iota = 1$. The term $u_\emptyset$ in (8) must satisfy this restriction as well. The restriction on $u_\emptyset$ will remain implicit whenever we refer to parametrizations such as those in (8) and (9).

In summary, we suggest a two-stage hierarchical prior. The first stage consists of setting $\theta \sim D(K, \eta)$ where η is parametrized using (9). The second stage consists of setting distributions on the u-terms in (9).

As a consequence of using the parametrization (9) one can specify a value for η by specifying values for some margins of η. For example, if $\log \eta_{ij} = u + u_{1(ij)} + u_{2(ij)}$, then $\eta_{ij} = \eta_{i+}\eta_{+j}$. In this fashion we specify the rc values η_{ij} by specifying values for η_{i+} and η_{+j}, a total of only $r + c$ values.

This result extends to the general case. When the loglinear parametrization (9) is hierarchical then η is totally specified by the value of the margins $\eta^{Y_1}, \ldots, \eta^{Y_T}$, where $\{Y_1, \ldots, Y_T\}$ is the generating class of the loglinear parametrization. Therefore, we can implement the second stage either by using distributions on the u-terms or by using distributions on the margins $\eta^{Y_1}, \ldots, \eta^{Y_T}$. For $Y \subset \bar{n}$ the Y-margin η^Y is defined as being the array whose entries are

$$\eta^Y_{i_1 \cdots i_n} = \sum_{\bar{n} \setminus Y} \eta_{i_1 \cdots i_n} = \sum_{i_j \in \bar{r}_j, j \in \bar{n} \setminus Y} \eta_{i_1 \cdots i_n}.$$

3 Implementation of the Gibbs Sampler

This section describes the specifics of the implementation of the Gibbs sampler for calculating the posterior densities of the cell probabilities.

We start with a brief review of the Gibbs sampler and refer the reader to Gelfand *et al.* (1990) and Gelfand and Smith (1990) for a detailed description of the use of Gibbs sampling in Bayesian inference.

Suppose that one wishes to estimate the density $[X]$ of the random variable X assuming it is possible to draw deviates from the conditional densities $[X|Y]$ and $[Y|X]$, where Y is another random variable.

The algorithm consists of iteratively repeating a two-step cycle. Before starting one draws a deviate $X^{(0)}$ from an arbitrary density $[X]_0$. Step one of the cycle is to draw a deviate $Y^{(1)}$ from $[Y|X^{(0)}]$. Step two is to draw $X^{(1)}$ from $[X|Y^{(1)}]$. Then one first replaces $X^{(0)}$ by $X^{(1)}$ and proceeds with the second cycle. A succession of cycles produces a sequence $(X^{(1)}, Y^{(1)}), (X^{(2)}, Y^{(2)}), \ldots, (X^{(i)}, Y^{(i)}), \ldots$. The sequence $X^{(i)}$ converges in distribution to $X \sim [X]$ and $Y^{(i)}$ converges in distribution to $Y \sim [Y]$.

Gelfand and Smith (1990) suggest building an estimate of the density $[X]$ as follows. Using the Gibbs sampler, obtain m independent replicates $(X_1^{(t)}, Y_1^{(t)}), \ldots, (X_m^{(t)}, Y_m^{(t)})$. With these deviates obtain the density estimate

$$\widehat{[X]} = m^{-1} \sum_{j=1}^{m} [X|Y_j^{(t)}]. \tag{10}$$

We use Gibbs sampling to estimate the posterior distribution $[\boldsymbol{\theta}|\boldsymbol{x}]$. For the two-stage prior $\boldsymbol{\theta} \sim [\boldsymbol{\theta}|\boldsymbol{\eta}]$ and $\boldsymbol{\eta} \sim [\boldsymbol{\eta}]$. We identify X with $\boldsymbol{\theta}$ and Y with $\boldsymbol{\eta}$ and use the following Gibbs scheme to draw deviates from the posterior $[\boldsymbol{\theta}|\boldsymbol{x}]$:

```
do j = 1, m
   Set θ^(0)
   do i = 1, t
      Step 1: draw η from [η|θ^(0)]
      Step 2: draw θ^(1) from [θ|η, x]
      θ^(0) ← θ^(1)
   end do
   η_j^(t) ← η
   θ_j^(t) ← θ^(0).
end do
```

On exit, this process has generated m independent deviates $\boldsymbol{\eta}_j^{(t)} \sim [\boldsymbol{\eta}^{(t)}], j = 1, \ldots, m$ and m independent deviates $\boldsymbol{\theta}_j^{(t)} \sim [\boldsymbol{\theta}^{(t)}], j = 1, \ldots, m$.

With these deviates, the density estimate in equation (10) is a finite mixture of Dirichlet densities,

$$\widehat{[\boldsymbol{\theta}|\boldsymbol{x}]} = m^{-1} \sum_{j=1}^{m} [\boldsymbol{\theta}|\boldsymbol{\eta}_j^{(t)}, \boldsymbol{x}].$$

It is particularly simple to evaluate the marginal density estimate of a cell probability. For example, in the situation of the introductory example,

$$\widehat{[\theta_{11}|\boldsymbol{x}]} = m^{-1} \sum_{j=1}^{m} [\theta_{11}|\boldsymbol{\eta}_j^{(t)}, \boldsymbol{x}], \tag{11}$$

where $[\theta_{11}|\boldsymbol{\eta}, \boldsymbol{x}]$ is the beta$(K^*\eta_{11}^*, K^*(1-\eta_{11}^*))$ density with $K^* = N + K$, $\eta_{11}^* = \alpha\eta_{11} + (1-\alpha)x_{11}/N$, and $\alpha = K/(N+K)$.

Automatically monitoring convergence is still an open issue; at present the best one can do is to prespecify the total number of iterations t, say.

The distribution $[\boldsymbol{\theta}|\boldsymbol{\eta}, \boldsymbol{x}]$, which is used in step two of a Gibbs sampler cycle, is Dirichlet with concentration parameter $K^* = (N + K)$ and cell means $\eta_\iota^* = K/(N+K)\eta_\iota + 1/(N+K)x_\iota$. Drawing deviates from the distribution $[\boldsymbol{\theta}|\boldsymbol{\eta}, \boldsymbol{x}]$ is straightforward. We chose to generate these deviates by independently generating $\gamma_\iota \sim$ Gamma$(p_\iota, 1)$, $\iota \in I$ with, $p_\iota = K^*\eta_\iota^*$, and then setting $\theta_\iota = \gamma_\iota / \sum_{\iota' \in I} \gamma_{\iota'}$. The joint distribution of $\{\theta_\iota\}$ is Dirichlet $D(K^*, \{q_\iota\})$ with $q_\iota = p_\iota/K^*$.

However, drawing deviates from the distribution $[\boldsymbol{\eta}|\boldsymbol{\theta}^{(0)}] = [\boldsymbol{\theta}^{(0)}|\boldsymbol{\eta}]\ [\boldsymbol{\eta}]/[\boldsymbol{\theta}^{(0)}]$, used in step one of a cycle, is not straightforward.

We suggest the following adaptation of the rejection method to sample from $[\boldsymbol{\eta}|\boldsymbol{\theta}^{(0)}]$. This adaptation uses deviates $\boldsymbol{\eta}$ from $[\boldsymbol{\eta}]$, which are easy to generate, to obtain deviates from $[\boldsymbol{\eta}|\boldsymbol{\theta}^{(0)}] = [\boldsymbol{\theta}^{(0)}|\boldsymbol{\eta}]\ [\boldsymbol{\eta}]/[\boldsymbol{\theta}^{(0)}]$.

```
accept ← false
do while ( not( accept ) )
   generate a deviate η from [η]
   generate a deviate v from U[0, B]
   if ( v ≤ [θ^(0)|η] ) then accept ← true
end do
```

Above, B is such that $B \geq [\boldsymbol{\theta}^{(0)}|\boldsymbol{\eta}]$ for all $\boldsymbol{\eta}$ in its domain. It is simple to show that an accepted $\boldsymbol{\eta}$ is a deviate from $[\boldsymbol{\eta}|\boldsymbol{\theta}^{(0)}]$. An important feature of this approach is that it does not require the calculation of $[\boldsymbol{\theta}^{(0)}]$, or an estimate of it, as is sometimes necessary in some implementations of the Gibbs sampler (see for example Gelfand and Smith, 1990).

A geneneralized rejection method that uses an enveloping function $B(\boldsymbol{\eta})$ for $\boldsymbol{\eta} \rightarrow [\boldsymbol{\eta}|\boldsymbol{\theta}]$ may increase the speed of this algorithm. At present, however, we will content ourselves with a boxed envelop, the main advantage being the ease of programming. Obtaining a good value for B is crucial for a good performance of the rejection method. The ideal choice is to find $\hat{\boldsymbol{\eta}}$ such that

$$[\boldsymbol{\theta}^{(0)}|\hat{\boldsymbol{\eta}}] = \max\{[\boldsymbol{\theta}^{(0)}|\boldsymbol{\eta}] : \log \boldsymbol{\eta} = \sum_{w \in \mathcal{A}} \boldsymbol{u}_w\},$$

and then take $B = [\boldsymbol{\theta}^{(0)}|\hat{\boldsymbol{\eta}}]$. Observe that $\boldsymbol{\eta} \rightarrow [\boldsymbol{\theta}^{(0)}|\boldsymbol{\eta}]$, is the Dirichlet likelihood function given the data $\boldsymbol{\theta}^{(0)}$.

The next subsection introduces a maximization procedure to find B. The procedure appears to be fast enough to use it in combination with the Gibbs sampler.

Observe that under the loglinear parametrization $\boldsymbol{\gamma} = \log \boldsymbol{\eta} = \sum_{w \in \mathcal{A}} \boldsymbol{u}_w$, $\hat{\boldsymbol{\gamma}} = \log \hat{\boldsymbol{\eta}}$ is the maximum likelihood estimate of $\boldsymbol{\gamma}$.

3.1 Maximizing the Dirichlet Likelihood

In this section we briefly describe the "gradient projection method" and apply it to maximize the Dirichlet loglikelihood. In addition to being easy to implement, various features of the Dirichlet likelihood and loglinear parametrizations make the gradient projection method preferable to other methods. We discuss the advantages of the gradient projection method after introducing additional definitions.

Recall that a loglinear parametrization for $\boldsymbol{\eta}$ restricts $\log \boldsymbol{\eta}$ to lie in a linear subspace M. The usual form of writing a loglinear parametrization with u-terms expresses $\boldsymbol{\gamma} \in M$ in terms of a basis matrix of M, i.e., a matrix B whose columns form a basis for M. When expressing a vector $\boldsymbol{\gamma} \in M$ in terms of the unique $\boldsymbol{u}$ such that $\boldsymbol{\gamma} = B\boldsymbol{u}$, the coordinates of $\boldsymbol{u}$ are the u-terms.

To avoid technical complications that the restriction $\sum_{\iota \in I} \eta_\iota = 1$ introduces, we redefine some functions of $\boldsymbol{\eta}$ as functions of $\eta_\iota, \iota \neq (r_1, \ldots, r_n)$ only. To this effect, for $\boldsymbol{\eta}$ given we define $\bar{\boldsymbol{\eta}}$ as $\bar{\eta}_\iota = \eta_\iota$, $\iota \in \bar{I}$ with $\bar{I} = I - \{(r_1, \ldots, r_n)\}$ as the index set for the vectors $\bar{\boldsymbol{\eta}}$.

Maximizing the Dirichlet likelihood $l(\boldsymbol{\eta}|\boldsymbol{\theta})$ is equivalent to maximizing

$$E(\bar{\boldsymbol{\eta}}) = K < \boldsymbol{\eta}, \boldsymbol{\lambda} > - \sum_{\iota \in I} \log \Gamma(K\eta_\iota),$$

where $\boldsymbol{\lambda} = \log \boldsymbol{\theta}$ and $\boldsymbol{\eta}$ is given by $\eta_\iota = \bar{\eta}_\iota, \iota \in \bar{I}$ and $\eta_{(r_1, \ldots, r_n)} = 1 - \sum_{\iota \in \bar{I}} \bar{\eta}_\iota$. Except for an additive constant, $E(\boldsymbol{\eta})$ is $\log l(\boldsymbol{\eta}|\boldsymbol{\theta})$.

The Dirichlet means corresponding to the multivariate logits $\boldsymbol{\gamma}$ are given by $\bar{\boldsymbol{\eta}} = H(\boldsymbol{\gamma})$ with $\bar{\eta}_\iota = \exp(\gamma_\iota)/\sum_{\iota' \in I} \exp(\gamma_{\iota'}), \iota \in \bar{I}$. To use the parametrization with the multivariate logits it is convenient to define

$$G(\boldsymbol{\gamma}) = E(H(\boldsymbol{\gamma})). \tag{12}$$

Observe that if we use the parametrization with the u-terms, then we may find $\boldsymbol{u}$, the m.l.e. of $\boldsymbol{u}$, by maximizing $U(\boldsymbol{u}) = G(B\boldsymbol{u})$.

Roughly speaking, there are three classes of alternative methods to maximize U. One possibilty is to solve the equation $JU(\boldsymbol{u}) = 0$, where JU stands for the array of partial derivatives of U. Typically, iterative procedures to solve this equation require updating an estimate of the Hessian of U after some iterations. On the one hand, it is difficult to obtain formulas for the second derivatives of U and on the other, computing second derivatives numerically is in general expensive and roundoff errors are difficult to control. Since $U(\boldsymbol{u}) = G(B\boldsymbol{u})$, this approach poses the additional difficulty of explicitly requiring a basis matrix for M.

An alternative is to use a steepest ascent method where at each step there is a unidimensional search along the direction $JU(\boldsymbol{u})$. This alternative also requires a basis matrix. In fact, any method that uses $\boldsymbol{u}$ as the variable of the objective function, will require a basis matrix.

The gradient projection method is preferable to these alternatives because it does not require estimating Hessians or a basis matrix of M. Moreover, the gradient projection method allows us to take advantage of the ANOVA-type parametrization for $\boldsymbol{\gamma}$ to perform certain computations more efficiently.

To use the gradient projection method we view the problem of maximizing the Dirichlet likelihood as the problem of finding $\hat{\boldsymbol{\gamma}} \in M$ such that

$$G(\hat{\boldsymbol{\gamma}}) = \max\{G(\boldsymbol{\gamma}), \boldsymbol{\gamma} \in M\},$$

which is a constrained maximization problem. A point $\boldsymbol{\gamma} \in M$ is refered to as a "feasible point". The gradient projection method projects the gradient of the objective function onto M to increase the value of $G(\boldsymbol{\gamma})$ and to maintain feasibility at the same time.

The following is a summary of the gradient projection to solve the above maximization problem:

Step 1 Initialization: Choose $\boldsymbol{\gamma}_0 \in M$
Let $v_0 = G(\boldsymbol{\gamma}_0)$
Step 2 Compute $\boldsymbol{d}_0 = JG(\boldsymbol{\gamma}_0)$
Step 3 Compute $\boldsymbol{\delta}_0 = P_M \boldsymbol{d}_0$
Step 4 Unidimensional maximization:
Find $\hat{\alpha} > 0$ such that
$G(\boldsymbol{\gamma}_0 + \hat{\alpha}\boldsymbol{\delta}_0)) = \max_{\alpha > 0} G(\boldsymbol{\gamma}_0 + \alpha\boldsymbol{\delta}_0))$
Set $\boldsymbol{\gamma}_1 = \boldsymbol{\gamma}_0 + \hat{\alpha}\boldsymbol{\delta}_0$
Step 5 Convergence test:
Let $v_1 = G(\boldsymbol{\gamma}_1)$
If $(v_1 - v_0)/v_0 < \epsilon$ then stop
else $\boldsymbol{\gamma}_0 \leftarrow \boldsymbol{\gamma}_1$
$v_0 \leftarrow v_1$
go to Step 2.

On exit, $\boldsymbol{\gamma}_1$ is such that $v_1 = G(\boldsymbol{\gamma}_1)$ is an estimate of the maximum value of G. Therefore $l(H(\boldsymbol{\gamma}_1)|\boldsymbol{\theta})$ is an estimate of the maximum value of $l(\boldsymbol{\eta}|\boldsymbol{\theta})$.

In Step 2, $JG(\boldsymbol{\gamma}_0)$ stands for the array of partial derivatives $\{\partial G(\boldsymbol{\gamma}_0)/\partial\gamma_\iota, \iota \in I\}$. It follows from (12) that, for $\iota = (i_1, \ldots, i_n) \in \bar{I}$ and $c = (r_1, \ldots, r_n)$,

$$\frac{\partial G}{\partial \gamma_\iota}(\boldsymbol{\gamma}) = K\eta_\iota[(\lambda_\iota - \lambda_c - \{\psi(K\eta_\iota) - \psi(K\eta_c)\})(1 - \eta_\iota) + \sum_{\iota' \in I}(\lambda_{\iota'} - \lambda_c - \{\psi(K\eta_{\iota'}) - \psi(K\eta_c)\})\eta_{\iota'}],$$

and,

$$\frac{\partial G}{\partial \gamma_c}(\boldsymbol{\gamma}) = K\eta_c[-\sum_{\iota' \in I}(\lambda_{\iota'} - \lambda_c - \{\psi(K\eta_{\iota'}) - \psi(K\eta_c)\})\eta_{\iota'}],$$

where ψ is the digamma function and $\lambda_\iota = \log\theta_\iota$, $\iota \in I$.

The formulas to compute the projection $P_M \boldsymbol{d}_0$ in Step 3 are derived in a similar fashion to the formulas to compute fitted values of the cell means in ANOVA. However, these formulas are not the same because the parametrization for $\boldsymbol{\gamma}$ does not involve the constant term of ANOVA parametrizations.

The existence of $\hat{\alpha}$ in Step 4 is guaranteed by the concavity of the Dirichlet likelihood. We used routine e04abf from the *NAg* library for the unidimensional maximizations. Although it would take more programming, perhaps an algorithm that uses the derivative of $G(\boldsymbol{\gamma}_0 + \alpha\boldsymbol{\delta}_0)$

with respect to α would be more efficient for the unidimensional maximizations.

It is possible to use other convergence tests in Step 5. Since our interest here is not on the maximizer $\hat{\gamma}$, but on the maximum value $G(\hat{\gamma})$, it is appropriate to use the test in Step 5 to ensure that on exit γ_1 provides a function value v_1 sufficiently close to $G(\hat{\gamma})$.

4 Illustrative Example

In this section we reanalyze the 2×2 table given in Table 1 which classifies college students with respect to their political affiliation and their family structure (from Braungart 1971, and analyzed in Bishop, Fienberg and Holland, 1975, pp 379-380), and by Albert and Gupta (1984). We use this data to estimate the cell probabilities using the prior belief that the two variables under study are plausibly independent. This is the situation described in the introduction. For illustrative purposes we use normal distributions on the u-terms in the parametrization

$$\log \eta_{ij} = u + u_{1(i)} + u_{2(j)}. \tag{13}$$

More precisely, we use

Stage I: $\theta | K, \eta \sim d(K, \eta)$, with η_{ij} reparametrized according to equations (13).

Stage II: The $u_{1(i)}$ are independent, $i = 1, 2$. The $u_{2(j)}$ are independent, $j = 1, 2$, and also independent of the $u_{1(i)}, i = 1, 2$. The distribution of $u_{1(i)}$ is $N(\mu_{1(i)}, \sigma^2_{1(i)})$ and the distribution of $u_{2(j)}$ is $N(\mu_{2(j)}, \sigma^2_{2(j)})$, $i, j = 1, 2$.

To use this prior density one first specifies the parameter vectors $\boldsymbol{\mu}_1 = (\mu_{1(1)}, \mu_{1(2)})$, and $\boldsymbol{\sigma}_1 = (\sigma_{1(1)}, \sigma_{1(2)})$, reflecting the user's prior knowledge about the proportion of students in the two political affiliations and, $\boldsymbol{\mu}_2 = (\mu_{2(1)}, \mu_{2(2)})$, and $\boldsymbol{\sigma}_2 = (\sigma_{2(1)}, \sigma_{2(2)})$, reflecting the user's prior knowledge about the proportion of students in the two family structures.

In this example we set $\boldsymbol{\mu}_1 = (.5; .5), \boldsymbol{\sigma}_1 = (2.0; 2.0)$ and $\boldsymbol{\mu}_2 = (.5; .5), \boldsymbol{\sigma}_2 = (2.0; 2.0)$, reflecting a rather imprecise belief about the u-terms. Second, one specifies a value for the parameter K.

Table 1: Parental decision making and political affiliation. Source: Braungart(1971).

		Political Affiliation	
		SDS	YAF
Parental Decision Making	Authoritarian	29	33
	Democratic	131	78

Albert and Gupta (1982) and Epstein and Fienberg (1991) computed the posterior means (5) for this table but they used different distributions to reflect uncertain prior beliefs about independence. In both articles the posterior expectation of the η's were estimated using a Monte Carlo method.

Table 2 reports the computed values for the posterior means of each of the cell probabilities for several values of K (the column headed by $K = \infty$ actually corresponds to a very large, but finite, value of K). The estimates corresponding to finite values of K reflect the uncertain prior belief in independence by compromising between estimates obtained under a saturated model and estimates obtained under an independence model.

Figure 1 reports reports estimates of the marginal posterior densities for each of the cell probabilities. These estimates were obtained using formula (11) for the posterior density of θ_{11} and with the obvious modifications for the other cell probabilies. We used $m = 20$ independent replicates and each of the replicates was generated with $t = 20$ cycles of the Gibbs sampler. In addition we computed these density estimates using different values of m and t. On a plot the resulting estimates appeared to be fairly similar for values of t and m as low as 10.

Table 2: Computed values of posterior means for different values of K

K	0	100	200	400	600	1000	2000	∞
$\hat{\theta}_{11}$	.107	.115	.119	.125	.130	.126	.135	.133
$\hat{\theta}_{12}$	.122	.115	.110	.105	.102	.098	.100	.093
$\hat{\theta}_{21}$	.483	.474	.471	.469	.468	.463	.453	.459
$\hat{\theta}_{22}$	.288	.296	.300	.302	.299	.313	.312	.316

5 Discussion

This article reports on an implementation of the Gibbs sampler to estimate the full posterior density of the array of cell probabilities of n-way contingency tables using the method proposed by Epstein (1990) and Epstein and Fienberg (1991). One easily obtains estimates of the posterior distributions of the individual cell probabilities as a finite mixture of beta densities.

Gelfand and Smith (1990) proposed the Gibbs sampler as an easy to implement algorithm to generate deviates from posterior distributions. An expeditious implementation requires that all necessary distributions be available for sampling. This was not the case in this article

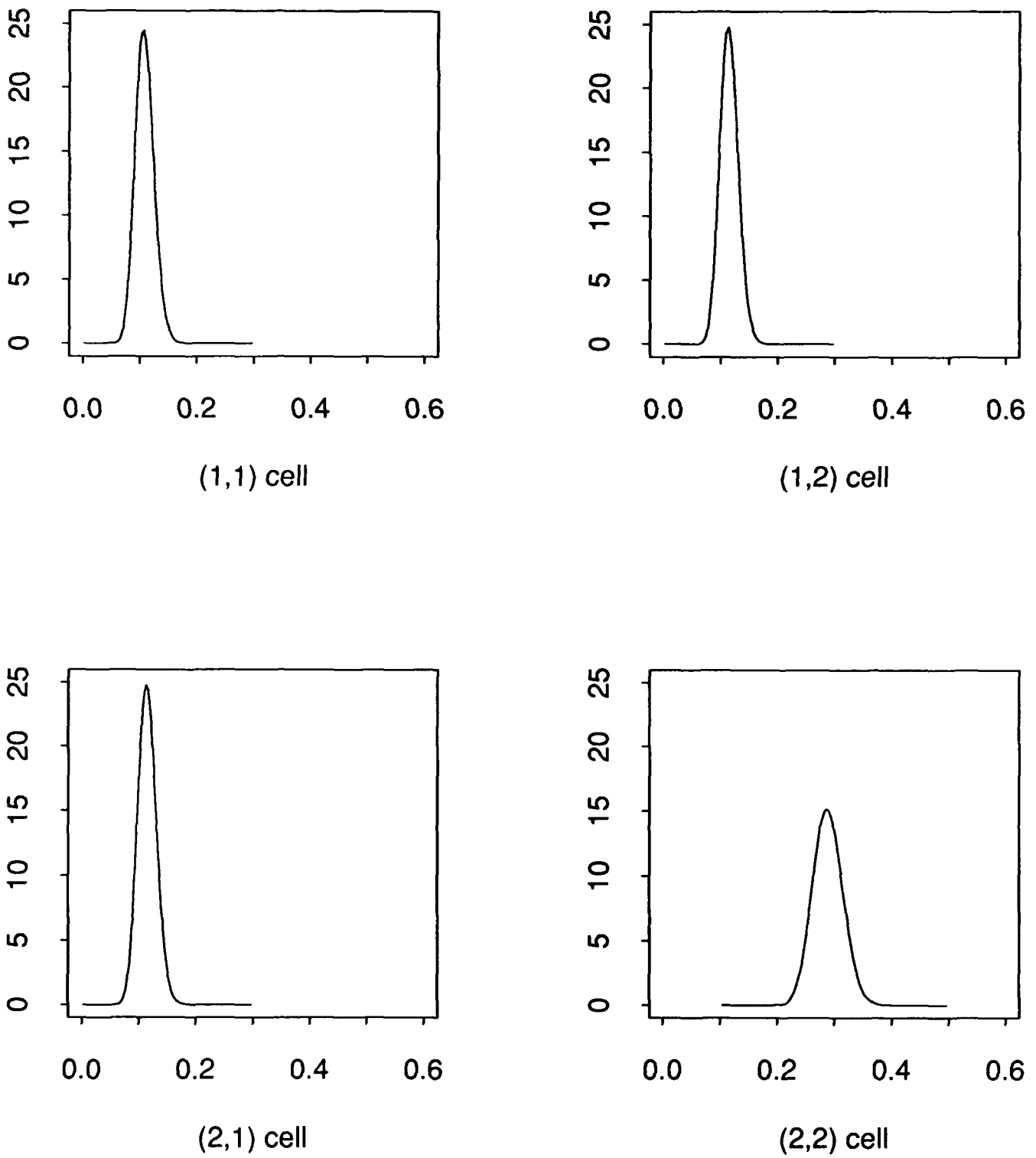


Figure 1:
Estimated posterior densities, K=150

and we expended some efforts to generate deviates from $[\eta|\theta]$.

To sample from $[\eta|\theta]$ we used a scheme that does not requires the marginal density $[\theta]$, which is often the main obstacle to compute $[\eta|\theta]$. The scheme uses the facts that $[\eta]$ is available for sampling, that $[\theta|\eta]$ as a function of η can be viewed as a concave likelihood function with a unique maximum. This maximum provides the height of a box for a rejection sampling method. The gradient projection method proved to be fast and very easy to program. We are currently investigating its use in maximum likelihood estimation for generalized linear models and will report on this work elsewhere.

Our scheme to sample from $[\eta|\theta]$ can be used to implement the Gibbs sampler for a variety of other problems involving two-stage priors where the first stage is the conjugate prior for the sampling distribution and the second stage distribution is available for sampling.

Furthermore, we feel that the simplicity of the Gibbs sampler warrants exploring new algorithms to generate deviates from distributions that thus far have not been available for sampling. For clarity we used a simple 2×2 example to illustrate our implementation.

In higher dimensional tables, it makes special sense to utilize the structure of η in terms of its marginals as part of the algorithm and to set up a cycle involving steps for the conditional densities for each of the marginals of η instead of a single step for $[\eta|\theta]$. We hope to report on the details of such an algorithm at a future date.

References

[1] ALBERT, A. H. & GUPTA, A. K. (1982), Mixtures of Dirichlet Distributions and Estimation in Contingency Tables. *Ann. Statist.*, **10** , *No. 4, 61-68.*

[2] ANDERSEN, A. H. (1974), Multidimensional Contingency Tables. *Scand. J. Statist.*, **1**, *115-127.*

[3] BISHOP, Y. M. M., FIENBERG, S. E. & HOLLAND, P. (1975). Discrete Multivariate Analysis: Theory and Practice. Cambridge, Mass: M.I.T. Press.

[4] BRAUNGART, R. G. (1971). Family status, socialization and student politics: a multivariate analysis. Cambridge, Mass: M.I.T. Press.

[5] EPSTEIN, L. D. (1990), Bayesian Estimation in Multidimensional Contingency Tables. Ph.D. Thesis. Department of Statistics, Carnegie Mellon University.

[6] EPSTEIN, L. D. & FIENBERG, S. E. (1991), Bayesian Estimation in Multidimensional Contingency Tables. *Bayesian Inference in Statistics and Econometrics: Proceedings of the Indo-US Workshop, 1988.* Lecture Notes in Statistics, Springer-Verlag New York. (To appear.)

[7] GELFAND, A. E., HILLS, S. E., RACINE-POON, A., & SMITH, A. F. M. (1990), Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling. *J. Am. Statist. Assoc.*, **85**,*No. 412, 972-985.*

[8] GELFAND, A. E. & SMITH, A. F. M. (1990), Sampling Based Approaches to Calculating Marginal Densities. *J. Am. Statist. Assoc.*, **85** , *398-409.*

[9] LEONARD, T. & NOVICK, M. R. (1986), Bayesian Full Rank Marginalization for Two-Way Contingency Tables. *J. Ed. Statist.*, **11** , *No. 1, 33-56.*

Seeing and Hearing Dynamic Loess Surfaces

W. M. Coughran, Jr.
AT&T Bell Laboratories
Murray Hill, NJ 07974, USA.
wmc@research.att.com

Eric Grosse
AT&T Bell Laboratories
Murray Hill, NJ 07974, USA.
ehg@research.att.com

Abstract

Scientific video, combining animated images with sound, is a powerful tool for understanding transient two-dimensional or static three-dimensional data. Animation of colored perspective plots often reveals subtle data characteristics not seen in a series of static images. Using the multidimensional data fitting technique *loess*, it is possible to construct dashed plots that visually represent the smoothed approximation and its local error. Sound can be used to add scalar parameters like time "tick marks" or the amplitude of an associated quantity; we describe means of processing such scalar data using variation-diminishing splines and specifics of sound generation including loudness equilibration. We also explain the limitations of our techniques and suggest some extensions.

1 Introduction

Our long-term research interests have been in simulation techniques and data fitting. Measured or simulated data often occur as values sampled at scattered locations in time and two or three spatial variables. Moreover, there are often one or more scalar parameters associated with such data sets, representing time, a global error value, an integral, and so forth. Such complex data fields are difficult to comprehend, but we have found that scientific video, combining animated images with sound, is of great help. We will describe some of the techniques we use to generate images and sound, trying to emphasize those that have not yet become common.

Although graphics hardware has become increasingly powerful, renderings of complex scenes with proper shading and lighting are still difficult to generate in real time for moderate cost. As a result, we employ interactive techniques where they are essential for data analysis, such as scatterplot brushing [2] and selecting a viewing perspective or position of a light source. Our model has been to generate high-quality rendered images that are then recorded a frame-at-a-time on NTSC videotape; by high quality, we mean anti-aliased perspective images including texture maps, lighting models, mild reflectivity, and transparency. Once an animated sequence has been generated on videotape, we then construct a synchronized "sound track".

We have presented some of our basic image tools elsewhere [5]. Our most important tool is the equi-spaced color-level plot with either orthographic or perspective projections. Here, we will describe a dashed surface plot that can simultaneously convey the shape of a two-dimensional function and its local error.

We also presented our basic sound tools in [5]. We have used sound in several forms. Sound has been most useful to underscore the passage of time in the form of beats. We have also found it useful to vary the pitch, volume, and tempo in order to represent other scalar quantities. Here, we will describe means based largely on variation-diminishing splines for stretching, smoothing, or compressing data to fit into a prescribed video segment. In addition, we explain how "loudness equilibration" can help make listener perception more uniform.

Our view is that both the monolithic system and the subroutine model of software communication are inappropriate for this application. It is often the case that data must be transmitted between machines or highly specialized programs. Hence, it is attractive to employ standardized, self-descriptive, ASCII file formats and use files or UNIX pipes to transmit data between disjoint processes. All of our tools use a uniform interface for exchanging data [4], based on the AWK paradigm [1]. We employ the RenderMan Interface Bytestream [12] (RIB) to decouple the modeling and rendering tasks while preserving reasonable generality in possible graphical techniques.

The next section (§ 2) presents our schemes for generating images. In § 3, our techniques for generating and manipulating sounds are discussed.

2 Colored and transparent surfaces for fields

The basic tool for understanding two-dimensional images is the color-level plot. This is the natural generalization of line-based contour plots adapted to modern raster graphics hardware that supports texture mapping. Orthographic projection results in a color-level plots that are close to traditional contour plots (using gray-scale instead of color brings one even closer). The eye does not respond uniformly in wavelength to light. Hence, it is important to choose colors that appear to be equi-spaced. This process can be reduced to a nonlinear least squares in an appropriate psychophysical metric [5].

A flat orthographic color-level plot is not the most intuitive representation for a two-dimensional data field, though in trained hands it is often the most informative. We have found that a perspective projection helps a great deal, particularly when the surface is given specular highlights. The choice of perspective is often best made with an interactive tool; we have described a "helicopter" model elsewhere [5]. Shading provides cues about inflections and other subtle phenomena.

Often a sequence of two-dimensional data fields is provided. The successive images may represent data at different times or as a function of another parameter. We employ frame-at-a-time animation to generate a video segment representing the data, where each frame is typically a color-level plot.

Our target medium is VHS videotape since, for now, that is the only universally presentable format. We have attended many meetings where the speaker complained that some critical feature was impossible to see in the displayed video but was quite clear in his lab. So we try to use all available techniques (such as anti-aliasing, color desaturation, and motion) to make legible at least the most important features. Forcing ourselves to stay with NTSC resolutions has also helped us avoid the tendency to add distracting dials and other dynamic icons to already complicated images.

Loess is a general mechanism for computing a smoothed approximation to scattered data, which produces local standard error estimates [3]. One new approach to viewing a *loess* surface and the local errors is the *dashed surface*. In two dimensions, the *dashed surface* is constructed by dividing the domain into squares and then trimming the color-level plot in each square in proportion to the error. That is, look at the a local square patch of the surface. The colored region of the square is retracted from the edges as the error increases. This trimming of the plot in the square must be done so that the patch area (not the perimeter) is inversely proportional to the error. This approach results in a plot that "breaks up" where the smoothed representation is not good; it is the natural generalization of a one-dimensional plot that becomes broken up into smaller and small dashes as the error increases. We reserve the use of error bars projected up from the surface for displaying actual residuals. See Figure 1.

3 Sound for scalar parameters

The use of sound is still rather speculative. In fact, many visually oriented people may question why it should be preferred over a more complex image. We would like to argue the case for sound to represent scalar parameters associated with an animated sequence of data fields. Examples of such scalar parameters are time, global error values, and functions computed from the data fields.

The simplest and most effective application of sound is to denote the passage of time in a simulation. Drumbeats are the natural sound to associate with a discrete moment in time and are analogous to the tick marks found on a scatterplot. (The same software that we use in our graphics software to pick "round" numbers for tick marks was immediately applicable to generating such time beats.) We found earlier that beats attached to a particular simulation hesitated at points where time steps had to be repeated. We had overlooked the repetitions of time in a laserprinter plot of the time progression; the beats stretched out the time points making it possible to hear things we had not seen earlier.

We have also considered the use of pitch, volume, and tempo to represent more complicated scalar functions or a combination of functions [5]. In one example, we had a scalar function of another scalar parameter. We varied the pitch to represent the "independent" scalar while increasing the volume and repetition rate in proportion to the "dependent" variable.

Our experience with auditory representations of more complex scalars is that they cannot replace line drawings. We have seen a number of animations where time (or another scalar) is represented by an analog clock or a number displayed on the periphery of the basic data field display, such as our color-level plots. (Variable length bars are a more workable alternative if there are only one or two bars.) Introducing extraneous visual information is often distracting — the eye must glance from the main display to the indicator and important information may be missed. If you examine a line drawing of the scalar parameters carefully, then the sound representation gives a qualitative impression of the scalar value without the annoying distraction. The other problem with adding scalar values as clocks or bars in the image is the limitations of

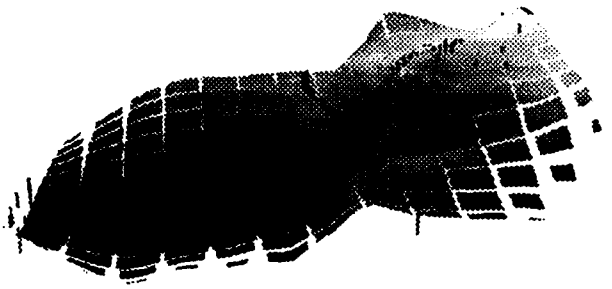


Figure 1: Qualitative display of standard errors by using a dashed surface.

NTSC resolution; we have found that the basic images are usually complex enough without trying to add other small auxiliary features. (Obviously, this latter comment doesn't apply to high resolution displays but the distraction comment does.)

Sound suffers from some of the same limitations as color: it has limited resolution, is not as familiar as conventional line drawings, and has some hard pyschophysical problems. In addition, sound is only applicable to animated sequences since it's inherently transient. Finally, human beings can assimilate much more information visually than via hearing so sound must be employed in limited ways.

Let us consider some of the parameters governing auditory perceptions and how we use them:

- pitch: If we constrain ourselves to the Western scale, we can take half-steps over three to five octaves. Although the use of major scales is more pleasing to the ear, it severely reduces the number of available notes and, hence, the resolution. Many (synthesized) instruments are incapable of generating notes over a wide frequency domain. We have considered trills to represent error bars but so far have found the technique to be of limited value.

- tempo: By this, we mean the duration and frequency of striking notes. Humans are surprisingly sensitive to variations in tempo.

- volume: It is possible to vary the volume over a number of relatively fine steps but perceptions are coarse. This problem is made worse by the fact that a particular fixed amplitude will be heard to have a different "loudness" as the pitch is varied.

- voice: We use this term to mean instrument. It is possible to distinguish and follow the notes generated by several instruments. We sometimes supplement this using stereo and reverberation to emphasize the separation of voices.

- melody: We have considered transitional notes and chording patterns to denote changes in scalar values, but have not been completely satisfied with the results at present. See [10] for a discussion of melody versus chord.

This variety of knobs would in principle allow the simultaneous presentation of many scalar variables. We have had better luck by instead presenting only one or two variables and using them to control several musical parameters; the redundancy overcomes some of the psychophysical difficulties.

We have built a number of tools to generate sounds based on our tensor/scatter file format [4]. The tools can generate percussion to denote time, percussion to act as a counter of discrete events (heard in the associated videotape to count number of Newton iterations), and a variety of more complex sounds with variable pitch, volume, and tempo. All of the tools generate standard MIDI [8] files to drive our synthesizer equipment [9]; MIDI allows the specification of events in time that start (or stop) a note with a particular velocity (roughly volume).

We have found that our generic approximation tools also play a role in sound generation. Often we will have a fixed sequence of scalar values that are uniformly spaced. Such a sequence may have to be translated into more than one similar sound sequence lasting different lengths of real time. Variation-diminishing splines can be used to expand, compress, or resample the sequence in a man-

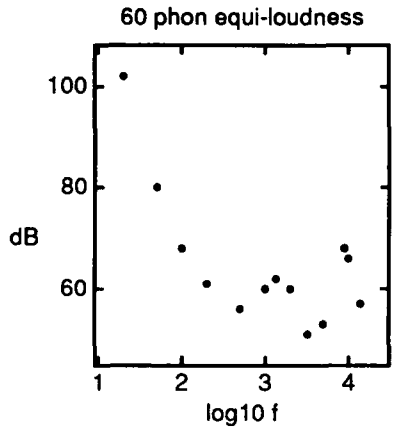


Figure 2: Plot of perceived dB level (60 phons) as frequency is varied.

ner suited to the sound segments of different lengths; for data with dramatic excursions (outliers), such variation-diminishing techniques also smooth the data so the sound is more palatable and, more importantly, easier to grasp (given the limitations of the ear). For non-uniform data, least-squares spline fits or *loess* can be employed.

We have experienced difficulties with loudness perception. As mentioned earlier, the human ear perceives notes of varying pitch but equal amplitude as having different loudness values. Figure 2 is based on measured data [11, p.45] and clearly shows that low and high frequency response is not flat. MIDI defines middle-C as note 60 so we can map note n onto frequency by

$$f = 2^{(n-60)/12} 523.25.$$

We can then build a loudness compensation function by using the above formula and logarithmic interpolation of the measured data from fig. 2. We have done this and find that it does improve volume perception — this permits us to use a wider range of frequencies and, hence, increase the resolution available with sound. The technique should be extended to include a second variable varying the amplitude (other phon values). Other possibilities would be to incorporate alternative loudness measures like $L = kI^{0.3}$ where L is loudness and I is intensity and to include temporal integration [11].

Another difficulty is that MIDI only lets us directly control the "velocity" and that is only loosely related to loudness for some instruments. Among the voices we use, piano and the xylophone seem about the most linear.

Sound is limited to providing a small number of cues for scalar parameters. We have found it possible to mark time with drumbeats and then use different instruments with variable pitch and so forth to track two or three scalars. Using more voices for more parameters seems ineffective.

There is a growing literature on the subjects discussed here. For example, other related papers in the proceedings that contain [5] are [6] and [7].

Acknowledgements

We thank Mark Kahrs and Tom Killian for supplying music generation and synchronization tools. RenderMan is a registered trademark of Pixar. Unix is a registered trademark of Unix Systems Laboratory, Inc.

References

[1] Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger. *The AWK Programming Language*. Addison-Wesley, New York, 1988.

[2] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29:127–142, 1987.

[3] William S. Cleveland and Eric Grosse. Computational methods for local regression. *Statistics and Computing*, 1:1, 1991.

[4] W. M. Coughran, Jr. and E. H. Grosse. A philosophy for scientific computing tools. *SIGNUM Newsletter*, 24:2–7, 1989.

[5] W. M. Coughran, Jr. and E. H. Grosse. Techniques for scientific animation. In E. Farrell, editor, *Proceedings of the 1990 SPIE/SPSE Conference #1259: Extracting Meaning from Complex Data*, pages 72–79. The Society for Imaging Science and Technology, 1990. Associated videotape in 1259-V collection.

[6] S. P. Frysinger. Applied research in auditory data representation. In E. Farrell, editor, *Proceedings of the 1990 SPIE/SPSE Conference #1259: Extracting Meaning from Complex Data*, pages 130–139. The Society for Imaging Science and Technology, 1990.

[7] G. G. Grinstein and S. Smith. Perceptualization of scientific data. In E. Farrell, editor, *Proceedings of the 1990 SPIE/SPSE Conference #1259: Extracting Meaning from Complex Data*, pages 190–199. The Society for Imaging Science and Technology, 1990.

[8] International MIDI User's Group. MIDI specification. Technical report, P.O.Box 593, Los Altos CA 94022, 1983.

[9] T. J. Killian. Computer music under the 10th Edition UNIX System. In *UNIX Research System Papers, Tenth Edition Volume II*, pages 477–482. Saunders College Publishing, 1990.

[10] David Lunney and Robert C. Morrison. Auditory presentation of experimental data. In E. Farrell, editor, *Proceedings of the 1990 SPIE/SPSE Conference #1259: Extracting Meaning from Complex Data*, pages 140–146. The Society for Imaging Science and Technology, 1990.

[11] B. C. J. Moore. *An Introduction to the Psychology of Hearing.* Academic Press, London, 2nd edition, 1987.

[12] Pixar. *The RenderMan Interface, Version 3.1.* 1989. 3240 Kerner Blvd, San Rafael CA 94901.

StatLog* An evaluation of machine learning and statistical algorithms

CHARLES C. TAYLOR
Department of Statistics
University of Leeds
Leeds LS2 9JT
United Kingdom

Abstract

StatLog is a European Community (ESPRIT) funded project which began in October 1990. There are about 10 academic and industrial partners involved in the project. Its aim is to complete an evaluation of the performance of Machine Learning and Statistical Algorithms on large-scale, complex commercial and industrial problems. The objectives of the project are threefold:

1. to provide critical performance measurements, and criteria for measurement on available Learning Algorithms which improve confidence in full exploitation;
2. to indicate the nature and scope of next-stage development which particular algorithms require to meet commercial performance expectations;
3. to indicate the most promising avenues of development for the commercially immature approaches.

This paper describes the project and the progress completed to date.

1 Introduction

In common with other ESPRIT projects, a consortium of academic and industrial partners work together, with different rôles, towards a common goal. The main goal of this project is in comparative testing of statistical and logical learning algorithms on large-scale applications in classification, forecasting, control and unsupervised learning. Other members of the consortium include the Universities of Strathclyde (U.K.), Granada (Spain), Porto (Portugal), and Lübeck (Germany), and industrial partners Daimler-Benz (Germany), Turing Institute (U.K.), Brainware (Germany), ISoft (France), Messerschmitt-Bölkow-Blohm (Germany) and the Institute of Automation (Germany).

Most of the algorithms in this project are applicable to problems in classification, and the main task is to run a large experiment involving a balanced design to measure the performance for algorithm × data set. In classification and forecasting problems it is fairly clear how to measure performance whereas in control and unsupervised learning these issues are still to be finalised. In this article we give some examples of the classification algorithms to be used in this project.

In addition there are a number of algorithms which deal with "unsupervised learning", i.e. methods which look for structure in the data. For example ITRULE [19], L-Induction [3] and some standard statistical methods such as principal components and projection pursuit. We mention this area of work in the section on different types of data. Finally, we discuss the procedure to obtain objective performance measures, and give the work schedule of the project.

2 Classification Algorithms

Due to time and resource constraints, the project will use, wherever possible, "off the shelf" packages. The methods to be considered can be grouped under a number of headings:

2.1 Neural Networks

Back-propagation is designed to overcome the limitations of the perceptron [18]. The architecture is composed of an input layer, an output layer and a set of internal "hidden units". We also consider a faster and more efficient variant known as **Quadratic back-propagation**.

*ESPRIT project number 5170.

Counter-propagation reduces the training time for back-propagation by making use of **Kohonen's** [11] self-organising algorithm and Grossberg's Outstar [6] algorithm.

2.2 "Classical Statistics"

These are all standard methods:
Discriminant analysis
Logistic regression
Multivariate analysis of variance
for which we will use routines from SAS, Splus and SPSS.

2.3 "Modern Statistics"

ALLOC80 [7] is a package which implements kernel density estimation methods using real, integer or nominal data.
Polytree algorithm. [16] Belief networks are directed acyclic graphs in which the nodes represent propositions or variables, the arcs signify direct dependencies between the linked propositions and the strengths of these dependencies are quantified by conditional probabilities. The polytree algorithm is used to recover the graph representing a probability distribution from a set of examples.
SMART [5] is a collection of fortran subroutines which perform Projection Pursuit classification and Projection Pursuit regression.

2.4 Bayesian Statistics

A naïve Bayes classifier which takes an encoded data set and builds a Bayesian Classifier. Real valued attributes are either given a normal model or a cut-point model.
Helen uses Bayes theorem without assuming independence of the attributes. A development of a method used for Galactic images [12].
IND[2]. A suite of C software which includes a CART [1] style decision tree system. Options allow CART style cost-complexity pruning by test set or by cross-validation, and a wide variety of splitting rules such as Bayesian, information gain and GINI (index of diversity) methods and a Wallace-style MML approach to cut points.

2.5 Genetic Algorithms

A Classifier system invented by Riolo [17] and implemented by Holland [9]. This set of algorithms allows learning to take place in parallel, rule-based, message-processing systems. Such a system contains: a classifier list containing condition-action rules; a message list, which acts as a "blackboard" or short-term memory; input and output interfaces with an environment. Learning can take place by competition between classifiers, discovery of new classifiers and a "Bucket-Brigade Algorithm" [8]

2.6 Machine Learning: Traditional and Relational

AlphaGolem [13] is a first-order induction algorithm based on relative least general generalisation. This generates rules from given examples, which are then used to classify new examples.
C4.5 [14] induces classification rules in the form of decision trees from a given set of examples which may contain unknown or noisy entries.
Cn2 [4] is an interactive induction algorithm which generates either rule sets (unordered rules), or rule lists (ordered rules) from examples, where each example is a set of attribute-value pairs. It can also determine the accuracy of a set of rules by applying it to a set of pre-classified examples.
First Order Inductive Logic (Foil) [15] is a relational machine learning algorithm which uses entropy as an heuristic.
Cal5 [21] constructs decision trees in real-valued domains. This uses an automatic analog-to-digital transformation. The definition of interval (corresponding to discretisation of the attributes) depends on the classification problem at hand and on the context, i.e. on the place of the test attribute within the tree, and must also be learned. Instead of using an entropy measure, interval formation is governed by statistical criteria.
AC2 includes an object oriented knowledge representation language. It is an extension of the decision tree algorithm ID3 to cope with relational data. It has a graphical user interface and outputs decision trees and rules. [10]
CRS learns relational structures based on graph theoretic measures [20].

3 Data Sets

In this setting, many statistical experiments would use a variety of simulated data with known properties whereas we are using real data, some of which has already been tried in machine learning problems. One of the criteria is that the data must be of commercial or industrial strength, so "toy" or "game" data sets have been deliberately excluded. Many of the data sets contain missing data and have other "warts" associated with real problems. We can group the data sets under a number of

headings according to the application, and we give one or two examples of each.

3.1 Classification problems

These constitute the main part of the effort, partly because it is clearer how they should be evaluated and partly because we can be more confident of achieving our stated aims. Examples include:

Protein folding. The database consists of examples of protein primary and secondary structure. The aim is to predict secondary structure from primary structure. There are about 10,000 examples, each consisting of 221 attributes followed by the decision class which represents alpha-helics, beta-strands and coil/turns.

Heart diseases. Several databases concerning heart disease diagnosis collected from various locations. There are up to 76 attributes including the angiographic disease status. This data has been used in previous studies so will provide external comparisons.

Hand-written digits. A 16 × 16 array of pixels with one of 256 grey-levels at each pixel. There are 10 classes (the digits $0, 1, \ldots, 9$) and 2000 examples for each class.

3.2 Forecasting or Prediction problems

These are typically short multivariate time series for which some Box-Jenkins methods have been tried, but there is interest in examining the performance of machine learning methods. The way in which performance measures are obtained will be similar to that given below, but since the outcome is real-valued, the proportion misclassified will be replaced by some other measure of discrepancy, such as mean squared error.

Car registration. Predicting the number of registrations for the whole car market and the heavy truck market. There are 56 examples constituting 11 predictive attributes, for example the industrial production index, selling prices in the retail trade, and the two values to be predicted. So far, standard Box-Jenkins methods and regression analysis have been used, and there is interest now in trying machine learning and neural net methods.

Currency exchange. The goal is to predict the US$-Sterling exchange rate three months ahead using current (and previous) financial indicators; for example retail sales volume, output per head, unemployment. In all there are 114 attributes and 141 examples. The decision "class" here is real-valued.

3.3 Control problems

A dynamic model has been used to describe the control of a TV satellite. There are high requirements for fuel consumption, pointing accuracy and positioning - one of the difficulties in the control task is the high disturbance during orbit correction manœuvres. The model uses differential equations to generate a time dependent output; typically the thruster exhibits non-linear characteristics with time delays so overshoots need to be kept to a minimum. A further difficulty is caused by fuel sloshing. The control system needs to be stable, fuel efficient and have good response times. It will be some combination of these factors that will be used in measuring the performance of the system. The question arises as to whether (and how) machine learning algorithms can be used in this process.

3.4 Structure problems

These are generally "unsupervised learning" in that the true class is not given in the training data. In inductive protein structure analysis the problem is to describe the protein super-secondary structure by clustering the examples. Each record consists of 30 floating point numbers which are normalised values of attributes describing a pair of secondary structures and the relationship between them. Each record is a potential example of a super-secondary structure. The performance measures for these problems will again be different to the supervised learning case.

4 Performance measures

The allocation of algorithm/data pairs will be done by the University of Strathclyde, who are directing technical aspects of the project. Each algorithm will be tested by an "expert user" and a "naïve user". Objective measures of performance will include processing time (for the training data and the test data), storage costs for the processing of the data and the consequent rule, and an error rate - probably measured by cross-validation and/or the bootstrap. Subjective measures will include ease of use, particularly as seen by the "naïve user", and robustness to required parameter input.

The procedure is that the data format will be revealed to the holder of the algorithm so that it can be modified to read the given format. The algorithm will then be deposited, together with clear instructions for usage, and the real data will be released. After the algorithm has been run the results will be validated and checked using the deposited algorithm. In the event that the results are radically different a third party will be asked to adjudicate.

5 Timetable

There will be some iterations within the testing process as algorithms are weeded out and refined in the early stages. At present the data sets and algorithms to be used are being finalised. From August 1991 those algorithms which are performing badly will receive an early warning and may be modified or excluded from the main trials. The full comparative trials are expected to commence in April, 1992 with all results summarised and analysed by January, 1993. The final three months will consider ways in which the best algorithms can be exploited and the results will be made known.

In tandem with the experimental findings there will be an effort to explain the results in a theoretical context. Amonst other things, this will determine whether the methods, or merely the implementation of the algorithm, has led to the results. A survey of previous comparisons (theoretical or experimental) will also be undertaken.

References

[1] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984) *Classification and Regression Trees*. Wadsworth, Belmont.

[2] Buntine, W. (1990) IND. Turing Institute, Glasgow.

[3] Chorbadijev, I. & Stender, J. (1990) Subsymbolic inductive learning framework for large-scale data processing, in *Symbols vs Neurons?* eds. Stender, J. & Addis, T.. IOS Press, Amsterdam, Washington, Tokyo.

[4] Clark, P. & Niblet, T. (1989) The Cn2 induction algorithm. *Machine Learning* 3(4), 261-283.

[5] Friedman, J.H. (1984) *SMART: Smooth Multiple Additive Regression Technique*. Department of Statistics and Stanford Linear Accelerator Center, Stanford University, California.

[6] Grossberg, S. (1969) Some networks that can learn, remember and reproduce ant number of complicated space-time patterns. *Journal of mathematics and mechanics* 19, 53-91.

[7] Hermans, J., Habbema, J.D.F., Kasanmoentalib, T.K.D., Raatgever, J.W. (1980) *ALLOC80 Discriminant Analysis Program*. Department of Medical Statistics, University of Leiden, Netherlands.

[8] Holland, J.H. (1985) Properties of the Bucket Brigade. *Proceedings of an International Conference on Genetic Algorithms and their Applications*, 1-7. Grefenstette, J.J. (Ed.) Carenegie-Mellon University, Pittsburg.

[9] Holland, J.H. (1986) Escaping brittleness: the possibilities of general-purpose learning algorithms applied to general rule-based systems, in *Machine Learning: an artificial intelligence approach, volume 2* Michalski, R.S., Carbonell, J.G. Mitchell, T.M. (eds.) Morgan Kaufmann Publishers Inc, Los Altos, CA.

[10] KATE: The Knowledge Acquisition Toolbox for Expert Systems. ISoft, Orsay, France.

[11] Kohonen, T. (1989) *Self-organisation and associative memory*. Springer-Verlag, 3rd edition.

[12] Molina, R. & Ripley, B.D. (1989) Using spatial models as priors in astronomical image analysis. *J. Applied Statistics*, **16**, 193-206.

[13] Muggleton, S., Feng, C. (1990) Efficient induction of logic programs, in *Proceedings ALT'90: First international conference on algorithmic learning theory.*

[14] Quinlan, J.R. (1986) Induction of decision trees *Machine Learning* 1, 81-106.

[15] Quinlan, J.R. (1990) Learning logical definitions from relations. *Machine Learning*, 5(3), 239-266.

[16] Pearl, J. (1988) *Probabilistic reasoning in intelligent systems: Networks of Plausible inference*. Morgan and Kaufmann Publishers Inc, Los Altos, CA.

[17] Riolo, R.L. (1986) *CFS-C: A package of domain independent subroutines for implementing classifier systems in arbitrary, user-defined environments*, Logic of computers Group, Univerisity of Michigan.

[18] Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986) Learning internal representations by error propagation, in *Parallel distributed processing: explorations in the microstructure of cognition* Edited by Rumelhart, D.E., McLelland, J. and the PDP Group, San Diego, MIT Press.

[19] Smyth, P. & Goodman, R.M. (1990) Rule induction using information theory, in *Knowledge discovery in databases* Piatetsky-Shapiro, G. Frawley, W. (eds.), MIT Press.

[20] Wysotski, F., Selbig, J. (1981) Learning structural descriptions from examples - an algebraic approach. *IJCAI, 7, Vancouver* pp. 153-158.

[21] Wysotski, F., Sobik, F., Müller, W. (1990) CAL5. Institute of Automation , Berlin, Germany.

COMPARATIVE STUDY OF SIX CLASSIFICATION METHODS FOR MIXTURES OF VARIABLES

By O . Cherkaoui(1) and R. Cléroux (2)

(1) Département d'informatique et de mathématiques, Université du Québec à Montréal, C.P. 8888 Succ. A, Montréal, Qué., H3C 3P8, Email: Cherkaou@info.uqam.ca
(2) Département d'informatique et de recherche opérationnelle , Université de Montréal, C.P. 6128 Succ. A, Montréal, Qué., H3C 3J7, Email: Cleroux@iro.udem.ca.

ABSTRACT

The performance of six discriminant methods is compared on simulated data consisting of mixtures of continuous, binary, ordinal and nominal variables. These methods are: Fisher's linear discrimination, logistic discrimination, quadratic discrimination, a kernel model, an independence model and the K-nearest neighbor method. In this paper, the simulation design was carrefully conceived. The independence model with an association parameter performs well and is very robust.

1 INTRODUCTION

In practice, the application of discriminant analysis is difficult because the underlying parameters which define the population are unknown. The choice between different discriminant analysis methods is hard specifically when the variables are mixtures of continuous and discrete variables. Most of the previous studies showed that the performance of the models are closely related to the underlying distribution. In recent years, several studies on the performance of discriminant methods have been published (Titterington et al. [1981], Knoke [1982] and Schmitz et al. [1985]). The performance of each discrimination methods is also dependent on the design simulation. The usefulness of a simulation depends highly on the quality of its design. When the objective is to choose between methods, modeling any multivariate interaction structure between a mixture of continuous and discrete variables is difficult. If the interaction design is too " sophisticated", it is hard to validate the model. In practical situations (for example, in supervised pattern recognition), we can only observe the interaction structure between two variables (in this paper we propose a simulation design which takes this into account). The main contribution here is the explicit parametrization of the simulation design and the consequent study of the effect of these parameters on each of the different discriminant methods.
The simulation design is given in section 2. Section 3 deals with the six methods of discrimination considered in this paper together with appropriate measures of performance. The results of the simulation study are described in section 4. Finally, we conclude with a brief discussion.

2 SIMULATION DESIGN

Most of the studies in which discrimination methods have been compared for mixed data deal with the simulation of multinormal distribution for all variables and with discretization of some of the components. This method unfortunatly has major drawbacks (Habbema et al. [1980]). In particular, the discretization of the continuous variables is such that it is hard to make a link between the multinormal distribution of the continuous data and the discrete distribution of the data obtained after discretization. The simulation of a mixture of continuous and discrete variables is still a difficult problem due mainly to the lack of a mixed distribution such as the multivariate normal distribution for continuous random variables or the multinomial model in the discret case. Here we will use the location model to simulate a mixture of continuous and discrete variables (Knoke [1982], Schmitz et al. [1985] and Krzanowski [1986]).

This study is concerned with the discrimination between two populations and four types of variables: X_1 binary, X_2 nominal, X_3 ordinal and X_4 continuous are considered. Two sets of sample sizes are used: (n_1=50,n_2=50) and (n_1=25,n_2=25). In each simulation, two sets of data are generated. The first set is the training set and is used to construct the discrimination rules. The second is the test set (or validation set) and is used to evaluate the performance of the different discrimination rules.

This simulation design deals with six parameters. Parameters A, B and D describe the distance between the two groups, while parameters C, E and F describe the association structure between the variables.

2.1 Interaction structure between continuous variables and discrete variables

Knoke [1982] suggested the use of the location model (Krzanowski [1975]) as a model of interaction between the continuous and the binary variables. The distance between the group means of continuous variables

depends on the binary and the nominal variables. The continuous variables X_3 and X_4 are simulated as functions of the discrete variables X_1, X_2 :

$$\mu_1-\mu_2 = \gamma \quad (2.1),$$

$$\mu_1-\mu_2 = (0.45)\frac{(-1)^{X_1(X_2+1)}}{4} + \gamma \quad (2.2),$$

$$\mu_1-\mu_2 = (0.45)(-1)^{X_1+X_2} + \gamma \quad (2.3);$$

where μ_1 and μ_2 are the mean vectors of variables X_3 and X_4 for group 1 and group 2 respectively. The parameter F of our simulation design describes this interaction structure: F=1 corresponds to equation (2.1), F=2,3 corresponds to (2.2),(2,3) respectively. The factor γ of this model corresponds to the parameter D of our simulation design. Three values are given for γ. The continuous variable X_3 is then discretized using quartile values to obtain an ordinal variable. Parameter E describes the covariance structure of the two populations for variables X_3 and X_4.

2.2 Interaction structure between nominal variables and binary variables

Kemps and Loukas [1978] considered the problem of random vector generation using only d-tuples of non-negative integers and they applying the inversion method. For our purpose, two discrete variables X_1 and X_2 are simulated as functions of the groups. The level of dependence is measured using level of significance of the Chi-square test because the sample sized dependancy of the Chi-square statistic. Parameter A corresponds to the dependence between X_1 and the groups, parameter B to the dependence between X_2 and the groups . Finally, C corresponds to the dependence between variables X_1 and X_2.

3THE DISCRIMINANT ANALYSIS METHODS

Before describing the six discriminant methods considered in this paper, it is convenient to introduce the notation and terminology of discriminant analysis. Individuals in the study are assumed to belong to one of two populations π_1 and π_2. The prior probabilities for populations 1 and 2 are respectively $p(\pi_1)$ and $p(\pi_2)$. Information is available on each individual in the form of a feature vector X of length p. Two sets of data are simulated. On the first set, a discriminant rule is a set up for assigning an individual to one of the two outcome categories given the feature **X** appropriate to that individual. In general, a discriminant rule will be a procedure for obtaining the posterior probabilities of the form

$$P(\pi_i/\mathbf{X}) = \frac{P(\mathbf{X}/\pi_i)\,P(\pi_i)}{P(\mathbf{X}/\pi_1)\,P(\pi_1)+P(\mathbf{X}/\pi_2)\,P(\pi_2)}$$

where $P(\pi_j)$ is the prior probability of group j and $P(\mathbf{X}/\pi_i)$ is the probability of observing the feature vector **X** for group i. Different choices for $P(\mathbf{X}/\pi_i)$ lead to different discriminant analysis methods.

Under the assumptions of multinormality of the density functions with equal or unequal covariance matrices, the linear or quadratic discriminant analysis are noted by $P_{LDA}(\mathbf{X}|\pi_i)$, respectively $P_{QDA}(\mathbf{X}|\pi_i)$.

If the density function $P(\mathbf{X}/\pi_i)$ is estimated by the non-parametric kernel method (KER), we willuse the density function as given by Habbema et al. [1978,a]. The smoothing parameters are estimated by the maximization of the modified likelihood function (according to the leaving-one-out method).

The logistic model (LOG), proposed by Day and Kerridge [1967] takes the parametric form P_{LOG} and the parameters are estimated by the maximum likelihood method.

The independence model (IND) assumes independence between the variables and deals only with discrete variables.The density is estimated by

$$P_{IND}(\mathbf{X}/\pi_i)\alpha\{\prod_{k=1}^{p} p_i(X_k)\}^{\beta} = \{\prod_{k=1}^{p}\frac{n_i(X_k)+1/C_k}{n_i+C_k}\}^{\beta}$$

where C_k is the number of categories of variable X_k, $n_i(X_k)$ is the number of elements with score X_k on variable k, n_i is the sample size of group i and β denotes an overall association parameter representing the "proportion of redundant information" between the variables (see Hilden et al. [1978]). To use this model here, we discretize the continuous variable using the quartiles of its distribution.

Fix and Hodges [1951] introduced the K-nearest neighbor method (KNN). The basic idea is to classify an individual into the population whose sample contains the majority of 'nearest neighbors'. The density is estimated by

$$P_{KNN}(\mathbf{X}/\pi_i) = \frac{K}{n_i V}$$

where K is the number of samples in the hypershere $\Gamma(\mathbf{X})$ centered at X, and V is the volume of the hypershere $\Gamma(\mathbf{X})$. Enas and Choi [1986] investigate the sensitivity of this method to the choice of K. They suggest choosing K as a function of the sample size and of the covariance matrix structure and propose K = 5.

3.1 Performance measures

Three measures of the ability to discriminate will be used to evaluate the performance of the six discriminant analysis methods. The performance of classification rules is often measured by estimating the error rates (i.e. the percentage of misclassified cases). Here we used the three measures: the percentage of missclassified, the quadratic score and the logarithmic score (see Habbema et al. [1978,b]. We will also compare the posterior probabilities obtained from the validation set for each discriminant analysis method.

4 RESULTS

The performances with respect to the simulation design parameters are compared for the two sample sizes.

4.1 Classification between the models

The rank-order score introduced by Schmitz et al. [1983] is used for rank-order analysis of the scores on the three performance measures : the error rates, the quadratic score and the logarithmic score. For each situation and each performance measure, the method with the best score gets rank 1, and the worst gets rank 6. Taking the average over all situations, the results are given in Table 4.1.

The KER method seems to be the best method. When the sample sizes are reduced, the LDA and LOG models are better than the IND model. Obviously, the parametric models (LGA,LOG,QDA) seem less affected by the sample size.

4.2 Performances of the methods with respect to the simulation parameters

Further analysis of each analysis of variance table revealed that the quadratic score is the most representative measure in terms of effect and interaction factors. Therefore, the quadratic score has been used to illustrate the results.

Table 4.2 shows the quadratic score for the remaining five methods and for parameters A,B,C and D. The performance improves in general with increasing dependence between the discrete variables and the discrimination groups (parameters A and B), except for the QDA and IND models. This improvement in performance is not linear with the level of dependence. We observe the converse for the QDA model.

The association parameter between the binary and nominal variables C has an unexpected result on the performance of the IND model. The performance of this model increases with the level of dependency of these two variables and the performances of the LDA and LOG models decrease with this parameter. The QDA model performs worse than the LDA and LOG models when C=2 and the converse is observed when the dependency is high (C=3).

The performance of all models increases with the distance parameter D. When the distance is important (D=3), the LOG and the LDA models are superior to the QDA model. This result follows from the fact that the parameter of dispersion E has less impact when the distance parameter D is important .

The QDA model is the most perturbed by the parameter of dispersion (E) for the ordinal and continuous variables. We note among other things that the classification rate changes with the parameters E and F. When the covariance matrices are equal , the LDA , LOG and KER models have the best performance (E=1). But for (E=2), the QDA model has a better performance. The IND model improves significantly when (E=3) over the QDA model.

The global association model parameter F yields almost the same results as parameter E. The IND model is less affected by this parameter than the other models.

4.3 Reliability of the reported error rates

In this section, we study the reliability of the reported error rates forthe six discriminant analysis methods. We compute the bias (also called apparent bias) for each method. This bias corresponds to the difference between the error rate obtained from the training set and the error rate obtained form the testing set. The results on this bias are given in Table 4.3. The smallest bias corresponds to the most reliable and the largest bias corresponds to the least reliable. The most reliable error rate is associated with the LDA model; the least reliable with the KER method.

4.5 Summary of the results

These results may be summarized as follows:

1) Previous studies showed comparable performances of the LDA and LOG models (Titterington et al. [1981], Schmitz et al. [1983], Schmitz et al. 1985]). We obtain similar results for these two models.

2) The difference in performance between the training samples of sizes 25 and 50 is very small.

3) The KER model showed the best ability to discriminate, but, in terms of reliability of reported error rates, it is the poorest. It is also the most computer intensive (ten times the computing time of the other methods).

4) In agreement with earlier results (Titterington et al. [1981]), the LDA and LOG models have remarkably reliable reported error rate.

5) The IND model with a good association parameter yields better results than the LDA and QDA models.

6) The IND model seems less affected by the discretization of the two continuous variables.

5 DISCUSSION

Surprisingly, the performance of the IND model seems less affected by the association parameters (C,E,F) than the other methods. The reliability of the reported error rates for this model is superior to that of the QDA and KER models although those models are supposed to be less sensitive to the association parameters.

Based on this study and on the recent literature on this topic, we have the following suggestions. When the discrimination is made for exploratory purposes (Schmitz et al. [1985]), we propose the use of a 'master' computer programme containing all methods. The KER and KNN models are too expensive in computing time, and can be excluded from this subset when the number of variables on the sample size is large. In this case, the LDA, LOG and QDA models can be used simultaneously and some derived methods (such as LDA augmented) can also be applied to improve the first results. On the other hand, when the object of the discrimination is to built an automatic classification system (as in clinical trials), we propose the IND model because it is the most flexible model for the choice of the best subset of variables. The simplicity and the good performance of this model will make it acceptable

when the set of predictor variables is not fixed and the sample size is large.

6 REFERENCES

Day,N.E. and Kerridge,D.F. [1967], A General Maximum Likelihood Discriminant, Biometrics,23,313-323.

Enas,G.G. and Choi,S.C. [1986], Choice of Smoothing Parameter and Efficiency of K-Nearest Neighbor Classification,Computers and Mathematics, 2A,235-244.

Fix,E. and Hodges,J.L. [1951], Nonparametric Discrimination: Consistency Proprieties. Project No 21-49-004, Report No.11, U.S. Air Force School of Aviation Medicine, Randolph Field, TX.

Habbema,J.D.F., Hermans,J. and Remme,J. [1978,a], Variable Kernel Density Estimation in Discriminant Analysis, In : Compstat 1978, Proceedings in Computational Statistics. Physica Verlag, Wien,178-185.

Habbema,J.D.F., Hilden,H. and Bjerregaard,B. [1978,b], The Measurement of Performance in Probabilistic Diagnosis I: The Problem, Descriptive Tools and Measures Based on Classification Matrices , Methods of Information in Medicine,17,217-226.

Habbema,J.D.F., Remme,J. and Hermans,J. [1980], A Simulative Comparaison of Linear Quadratic and Kernel Discrimination , Journal of Statistical Computation and Simulation,11,241-250.

Hilden,J., Habbema,J.D.F. and Bjerregaard,B. [1978], The Measure of Performance in Probabilistic Diagnosis III, Measures based on the Continuous functions of the Diagniostics Probabilities, Methods of Information in Medicine,17,227-237.

Kemps,C.D. and Loukas,S. [1978], The Computer Generation of Bivariate Discrete Random Variables, Journal of the Royal Statistical Society,Series A, 141,513-519.

Knoke,J.D. [1982], Discriminant Analysis with discrete and continuous variables, Biometrics,38,191-200.

Krzanowski,W.J. [1975], Discrimination and Classification Using Both Binary and Continuous Variables, Journal of American Statistical Association,70,782-790.

Schmitz,P.M.I., Habbema,J.D.F. and Hermans,J. [1985], A Simulation Study of the Performance of Five Discriminant Analysis Methods for Mixtures of Continuous and Binary Variables, Journal of Statistical Computation and Simulation,23,69-95.

Schmitz,P.M.I., Habbema,J.D.F., Hermans,J. and Raatgever,J.W. [1983], Comparative Performance of Four Discriminant Analysis Methods for Mixtures of Continuous and Discrete Variables , Communication in Statistics: Simulation and Computation,12,727-757.

Titterington,D.M., Murray,G.D., Murray,L.S., Spiegelhalter,D.J., Skene,A.M., Habbema, J.D.F. and Gelpke,G.J. [1981], Comparaison of Discrimination Techniques Applied to a Complex Data Se of Head Injured Patients , Journal of the Royal :tatistical Society, Series A, 144,145-175.

Table 2.1 Parameter E : Covariance matrix for each population

Parameter	Covariance matrix of population π_1	Covariance matrix of population π_2
E=1	$\Sigma_1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$	$\Sigma_2 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$
E=2	$\Sigma_1 = \begin{bmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{bmatrix}$	$\Sigma_2 = \begin{bmatrix} 1.0 & -0.5 \\ -0.5 & 1.0 \end{bmatrix}$
E=3	$\Sigma_1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$	$\Sigma_2 = \begin{bmatrix} 3.0 & 0.0 \\ 0.0 & 3.0 \end{bmatrix}$

Table 4.1 Classification score of the methods with respect to sample size

	KER	IND	QDA	LDA	LOG	KNN
(25,25)	2.34	3.44	3.80	3.09	3.38	4.91
(50,50)	2.01	3.09	3.60	3.45	3.72	5.10

Table 4.2 Quadratic score (n_1=50,n_2=50)

	KER	IND	QDA	LDA	LOG	KNN
A=1	.226	.227	.217	.226	.249	197
A=2	.216	.210	.209	..216	.271	.189
A=3	.207	.206	.202	.207	.270	.178
B=1	.223	.221	.212	.223	.249	.190
B=2	.214	.209	.208	.214	.271	.191
B=3	.212	.213	.209	.212	.270	.186
C=1	.213	.212	.212	.213	.249	.190
C=2	.217	.220	.211	.217	.271	.191
C=3	.219	.210	.206	.219	.270	.186
D=1	.228	.220	.219	.228	.249	.197
D=2	.219	.215	.211	.219	.271	.186
D=3	.202	.208	.199	.202	.270	.181
E=1	.217	.237	.219	.217	.249	.202
E=2	.215	.187	.217	.215	.271	.175
E=3	.217	.218	.193	.217	.270	.190
F=1	.218	.231	.212	.218	.249	.210
F=2	.208	.197	.206	.208	.271	.174
F=3	.223	.215	.210	.222	.270	.183

Table 4.3 Reliability of the reported error rates

	KER	IND	QDA	LDA	LOG	KNN
Biais	.138	.047	.107	.026	.027	.072

Localized Exploratory Projection Pursuit

Nathan Intrator*
Center for Neural Science
Brown University
Providence, RI 02912

Abstract

Based on CART, we introduce a recursive partitioning method for high dimensional space which partitions the data using low dimensional features. The low dimensional features are extracted via an exploratory projection pursuit (EPP) method, localized to each node in the tree. In addition, we present an exploratory splitting rule that is potentially less biased to the training data. This leads to a nonparametric classifier for high dimensional space that has local feature extractors optimized to different regions in the input space.

1 Introduction

Due to the *curse of dimensionality* (Bellman, 1961) it is desirable to extract features from a high dimensional data space before attempting a classification. This may be done in those cases where the important structure is assumed to lie in a low dimensional subspace of the original data. The most well know method for extracting features is principal components, however it has been argued that these features may not retain the structure needed for classification (Duda and Hart, 1973; Huber, 1985). A more general and powerful method for feature extraction is Projection Pursuit, and its unsupervised version - Exploratory Projection Pursuit (Friedman and Tukey, 1974; Friedman, 1987). This method has been extended in various directions, and is reviewed in (Huber, 1985).

One of the advantages of EPP is the use of locally smooth objective functions in the search for interesting features. Such functions are not related to the class labels, and have the potential of avoiding the curse of dimensionality (Huber, 1985). The method has an underlying assumption of homogeneity of the input space. Intuitively this means that a useful feature can only be found based on all of the input patterns. This posses a disadvantage which is due to the fact that the labels are not used through the search for good projections, and therefore, it is possible to ignore features that may only be important for classifying a small portion of the input data but are less interesting when considering the data as a whole. This observation is one of the motivations of recursive partitioning methods, including tree structured algorithms.

The proposed method is based on the classification and regression tree algorithm of CART (Breiman et al., 1984). Section 2 discusses CART briefly, and indicates how the hybrid tree is constructed. A new splitting criterion based on a variation of a back-propagation network is presented in section 3. Finally a short discussion containing the basic highlights of the method is given.

2 The Hybrid CART

CART addresses high dimensional space problems by partitioning the space and replacing complex classifiers (or regressors) designed for the whole input space, by a set of simpler modules working on smaller subregions of the space. There have been some recent attempts for recursive partitioning classification [see for example (Jacobs et al., 1991; Sankar and Mammone, 1991)].

CART's main contribution to earlier decision trees is the treatment of the additional bias introduced by the over-partitioning of the space. This is done by using a splitting rule that does not try to reduce missclassification error and by introducing a bottom up approach to pruning the full grown tree based on cross validatory error estimation. The pruning mechanism is a very powerful tool, and may be useful in remote applications of CART such as image compression using vector quantization (Riskin et al., 1990).

CART is not directly applicable to classification prob-

*This work was supported in part by the National Science Foundation, the Office of Naval Research, and the Army Research Office.

lems in very high dimensional spaces, such as gray level pixel images, since splitting based on a single dimension (single pixel in this case) is unlikely to increase the homogeneity of sub regions in the space. In this work, the recursive partitioning is based on features extracted using an EPP method. At each node of the tree, additional features are sought before the split is constructed, using only that portion of the input space that arrives to this node, and these new features are added to the features extracted so far, to construct an optimal split at that node. This leads to a combination of feature extraction and recursive partitioning that has the potential to be much more powerful than each of the methods by itself. Moreover, this method is still consistent with the monotonicity requirement of the cost at each split (Breiman et al., 1984), and therefore allows the use of the powerful pruning mechanism of CART.

The construction of the hybrid tree is the same as in the CART method (Breiman et al., 1984) with the exception that every node can perform additional feature extraction based on the high dimensional input patterns that arrive at that node, and based on the features extracted so far. The construction of a nested sequence of trees, the pruning based on cost, complexity cross-validation and the final tree selection can all be done exactly in the same way as in CART.

The feature extraction part of a node is implemented by an EPP method that seeks multimodality in the projected distributions (Intrator, 1990). This method is based on a biologically motivated synaptic modification equations (Bienenstock et al., 1982), and is computationally practical for high dimensional spaces, making it suitable to be used as the feature extractor in the proposed hybrid EPP/CART method.

3 Pseudo-Supervised Network

Although the proposed hybrid EPP/CART is able to use any of the CART splitting rules, we would like to consider a new exploratory splitting rule that allows linear combination splits. Linear combination splitting using linear discriminant functions was introduces in (Friedman, 1977) and was later replaced by the algorithm implemented in CART. The argument against linear combination splitting rules was that they were found to be more biased. This bias comes from the fact that the split is constructed in order to minimize some measure of nonhomogeneity based on the class labels, but with no concern to the structure of the space induced by the input patterns. A simple example of a possible bias is shown in figure 1. In this figure, a two dimensional structure (possibly part of a much higher dimensional space) can be split in various ways all of which are similar in the sense that they yield two pure subnodes. However, if the data contains patterns that lie outside the two ovals it is likely that only split 2 is optimal. In this section we

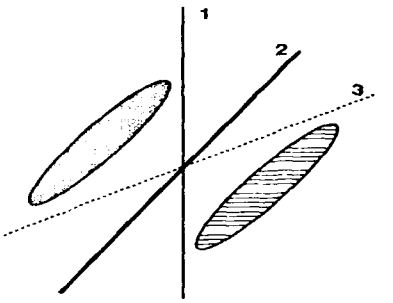


Figure 1: Optimal split (2) and nonoptimal ones (1,3). A method that tries to maximize homogeneity based only on the class labels will not distinguish between these splits, however, it is likely that split (2) will have better generalization properties (will be less biased to the training data).

present a splitting rule that is based solely on the input patterns. This rule can be incorporated into the original CART method, and potentially to other recursive partitioning methods.

Consider a split that assigns the value 1 to all the members of the training set at node t that belong to t_R, and the value 0 to the members of t_L, so that both sets are nonempty. Let $F = \{f_\alpha\}$ be a set of continuous functions that depend on a parameter α, f_α maps the input space to $[0, 1]$. Let χ_t be a characteristic function assigning the value 1 to $x \in t$, and 0 else. For a given split s, assume that f_{α_s} is the best approximator (not necessarily unique) to the characteristic function χ_{t_R} in the MSE sense. Now seek the optimal split s^* so that $E[(f_{\alpha_{s^*}} - \chi_{t_R^*})^2]$ is minimized.

Finding an optimal split in this way ensures that within a given set of continuous functions, this split results in a function which is able to assign the data in t_L a value closest to zero (in the MSE sense), and the data in t_R values that are closest to one. Thus ensuring that the patterns that belong to t_R are in some measure close to each other, and far apart from the patterns in t_L, i.e., increased homogeneity of the input space.

An example where this splitting rule along with feature extraction may be useful is given in figure 2. It shows a subregion in space in which two classes are strongly mixed. A supervised splitting algorithm will

split according to hyperplane 1 whereas the above unsupervised splitting rule will prefer to split according to hyperplane 2. This is because split 1 increases the purity of each node more than split 2 although split 1 does not focus on the confusion region between class A and B. It is conceivable that if the confused region is transferred in full to a node, and then an attempt to extract more informative features only from this region is made, the new representation will have a better chance to reduce the confusion between the classes in this subregion.

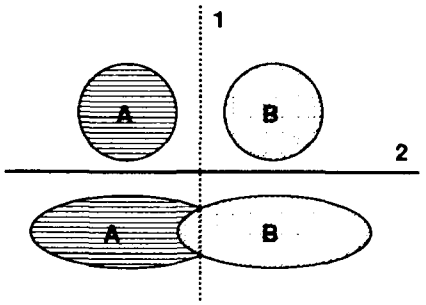


Figure 2: The ability of an unsupervised splitting rule to reduce confusion.

3.1 Splitting Rule Implementation

In order to use a gradient descent method for finding the optimal split, we need to overcome the discontinuity introduced by the function χ_{t_R}. Therefore, a continuous approximation to χ_{t_R} is used. We shall follow the notations presented in (Rumelhart et al., 1986), and present a splitting rule that is based on a variation of error backpropagation network.

Let o_{pj} be the output of the j'th splitting rule function for input pattern p. f_j is a sigmoidal activation function defined by $f_j(t) = [1 + \exp(-t)]^{-1}$, so that $o_{pj} = f_j(net_{pj})$, where $net_{pj} = \sum_i w_{ji} o_{pi}$. Let the target for output j be also defined in terms of the network activity, $t_{pj} = \tilde{f}_j(net_{pj})$, where $\tilde{f}_j$ is a sigmoidal function with a gain constant $\lambda > 1$, $\tilde{f}_j(t) = [1 + \exp(-\lambda t)]^{-1}$. The network is trained to minimize the empirical MSE $\sum_p (t_p - o_p)^2$. In order to avoid trivial splits it is possible to add penalty of the form

$$\kappa\left[1 - \left(\frac{1}{n}\sum_p o_p\right)\left(\frac{1}{n}\sum_p (1 - o_p)\right)\right],$$

for some small constant κ, however, simulations show that the trivial split does not usually happen especially when there are several neurons in the hidden and output layer.

The difference between t_{pj} and o_{pj} is shown in figure 3. This target function approximates a characteristic function, an approximation which will improve when $\lambda \to \infty$. In practice, there is no need to have λ be greater than 5. The calculation of the gradient with respect to the weight w_{ij} follows in the same way as in (Rumelhart et al., 1986), when taking into account the fact that the target depends on the network output as well. For an output layer unit j we have

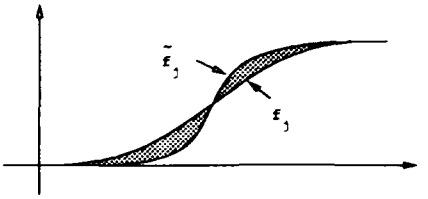


Figure 3: The minimization of the pseudo-supervised MSE, is equivalent to minimizing the shaded area in the picture.

$$-\frac{\partial E_p}{\partial w_{ji}} = -\left[\frac{\partial E_p}{\partial o_{pj}}\frac{\partial o_{pj}}{\partial net_{pj}} + \frac{\partial E_p}{\partial t_{pj}}\frac{\partial t_{pj}}{\partial net_{pj}}\right]\frac{\partial net_{pj}}{\partial w_{ji}},$$

and it follows that for

$$\delta_{pj} = (t_{pj} - o_{pj})\left[o_{pj}(1 - o_{pj}) - \lambda t_{pj}(1 - t_{pj})\right],$$

we get

$$-\frac{\partial E_p}{\partial w_{ji}} = \delta_{pj} o_{pi}.$$

The calculation of the gradient with respect to a hidden unit weight is exactly as in (Rumelhart et al., 1986), and will not be repeated here.

An intuitive explanation to this target definition is similar to the reasoning behind hard and soft competition approaches (Hinton and Nowlan, 1990). If a hard target (0 or 1) is imposed, then whenever the output is close to .5 which means that the input is close to the boundary, the error signal would be large. However if the input is close to the boundary, it is likely to be on the wrong side of the boundary, which will then lead to a large wrong correction signal. Using the soft target which takes into account the confidence in the output solves this problem, since the target is also close to 0.5. Another explanation is obtained by observing that the target is also dependent on the synaptic weights, and therefore the gradient of the synaptic weights with respect to the output should be taken into account as well. This requires the use of a soft target.

The construction of a binary splitting rule based on the above criterion is done by letting the PS network converge (or stop training based on another criterion) and then assign the patterns for which the output of the network is greater than .5 to t_R. In the case of a multi-split, assign to set j the patterns for which the output of unit j in the network is greater than .5.

4 Discussion

A method of recursive partitioning for high dimensional input spaces was introduced. This was done by combining the benefits from exploratory projection pursuit with those from the CART method. A new exploratory splitting rule was presented, and argued to have the potential to be less biased to the training data. This splitting rule, can have a boundary that contains an arbitrary predefind number of hyperplanes by defining the number of hidden units in the feedforward network, and is easily extended into multiple splits. The implementation of the splitting rule using a new unsupervised training algorithm to back-propagation is potentially useful to other purposes where a soft competition rule is better than a hard one, e.g. in adaptive equalization (Lucky, 1966; Hinton and Nowlan, 1990).

Combining all the above ingredients together, results in a computationally practical method for non-parametric classification in very high dimensional spaces, that is less sensitive to the curse of dimensionality due to the feature extraction, and is less biased to the training data, due to the sophisticated tree construction of the CART method.

References

Bellman, R. E. (1961). *Adaptive Control Processes.* Princton University Press, Princton, NJ.

Bienenstock, E. L., Cooper, L. N., and Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.*, 2:32–48.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees.* The Wadsworth Statistics/Probability Series, Belmont, CA.

Duda, R. O. and Hart, P. E. (1973). *Pattern classification and scene analysis.* John Wiley, New York.

Friedman, J. H. (1977). A recursive partitioning decision rule for nonparametric classification. *IEEE Trans. Comput.*, 26:404–408.

Friedman, J. H. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association*, 82:249–266.

Friedman, J. H. and Tukey, J. W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.*, C(23):881–889.

Hinton, G. E. and Nowlan, S. J. (1990). The bootstrap widrow-hoff rule as a cluster-formation algorithm. *Neural Computation*, 2(3):355–362.

Huber, P. J. (1985). Projection pursuit. (with discussion). *The Annal. of Stat.*, 13:435–475.

Intrator, N. (1990). Feature extraction using an unsupervised neural network. In Touretzky, D. S., Elman, J. L., Sejnowski, T. J., and Hinton, G. E., editors, *Proceedings of the 1990 Connectionist Models Summer School*, pages 310–318. Morgan Kaufmann, San Mateo, CA.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Lucky, R. W. (1966). Techniques for adaptive equalization of digital communications systems. *Bell Systems Technical Journal*, (45):255–286.

Riskin, E. A., Lookabaugh, T., Chou, P. A., and Gray, R. M. (1990). Variable rate vector quantization for medical image compression. *IEEE Transactions on Medical Imaging*, 9(3):290–298.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing: Explorations in the microstructure of Cognition. Vol. 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA.

Sankar, A. S. and Mammone, R. J. (1991). Neural tree networks. In Mammone, R. J. and Zeevi, Y., editors, *Neural Networks: Theory and Aplications.* Academic Press, New York.

Adaptive Probability Density Estimation in Lower Dimensions using Random Tessellations

Leonard B. Hearne and Edward J. Wegman

Center for Computational Statistics
George Mason University
Fairfax, Virginia

Abstract

This paper presents a class of non-parametric density estimators on a low dimensional space. The support of these estimators is defined by the convex hull of the set of observations. A random sample from the set of observations is used to tessellate the interior of the convex hull. The attribution of empirical probability mass to the tiles resulting from the tessellation produces a density estimate. With a set of appropriate linear constraints on the attribution of mass, the estimator is shown to be a conditional maximum likelihood estimator. Repeating this procedure, and averaging these density estimates within tiles, produces a bootstrap estimate of the density function. The results of this resampling and density estimation process are presented in graphic form.

1. Introduction

The objective of this paper is to construct a class of non-parametric probability density estimators, $\hat{f}(\boldsymbol{x})$ of $f(\boldsymbol{x})$, that make few, and comparatively weak assumptions about the support and characteristics of $f(\boldsymbol{x})$ beyond that provided by a set of observations, $\boldsymbol{Y}$. Let $\boldsymbol{Y} = \{Y_1, \cdots, Y_n\}$ be a set of observations, with $Y_i \in S^d$, $i = 1, \cdots, n$, and S^d a d-dimensional real product space, $\left(S^d, \mathfrak{B}(S^d), \mu\right)$, with μ the usual d-dimensional Lebesgue measure. A non-empty class of estimators, $\hat{f}(\boldsymbol{x})$, exists that has the maximum likelihood property, and is strongly consistent, given a set of observations $\boldsymbol{Y} \subset S^d$, $1 \leq d < \infty$, and a minimum number, λ, of observations per tessellating tile.

2. Support

The support for $f(\boldsymbol{x})$ is defined $\boldsymbol{A} \equiv \left\{\boldsymbol{x} \in S^d : 0 < f(\boldsymbol{x}) < \infty\right\}$. We will define the support $\boldsymbol{A}^n$ for $\hat{f}(\boldsymbol{x})$, given a set of observations $\boldsymbol{Y}$ of size n, as the smallest closed, convex region in S^d that contains $\boldsymbol{Y}$. Note that if another observation is added to $\boldsymbol{Y}$ then $\boldsymbol{A}^n \subseteq \boldsymbol{A}^{n+1}$. Another way to describe $\boldsymbol{A}^n$ is to say that $\boldsymbol{A}^n$ is the set defined by the convex combination of the elements of $\boldsymbol{Y}$. Let $\boldsymbol{H}$ be the set of $Y_i \in \boldsymbol{Y}$ that are on the convex hull of $\boldsymbol{A}^n$. Then the definition of $\boldsymbol{A}^n$ can be formulated as $\boldsymbol{A}^n \equiv \{\boldsymbol{x} \in S^d : \boldsymbol{x} = \boldsymbol{\alpha H} + (1 - \boldsymbol{\alpha})\boldsymbol{H}\}$, for all $\boldsymbol{\alpha}$, $\boldsymbol{0 \leq \alpha \leq 1}$.

$\boldsymbol{A}^n \in S^2$ can be seen in Figure 1 as the region defined by line segments connecting points in the point cloud of observations such that $\boldsymbol{A}^n$ is convex, and $\boldsymbol{Y}$ is contained in $\boldsymbol{A}^n$. Also, $\boldsymbol{H}$ is the set of $Y_i \in \boldsymbol{Y}$ that are vertices for the line segments that define $\partial \boldsymbol{A}^n$, the convex hull of $\boldsymbol{A}^n$.

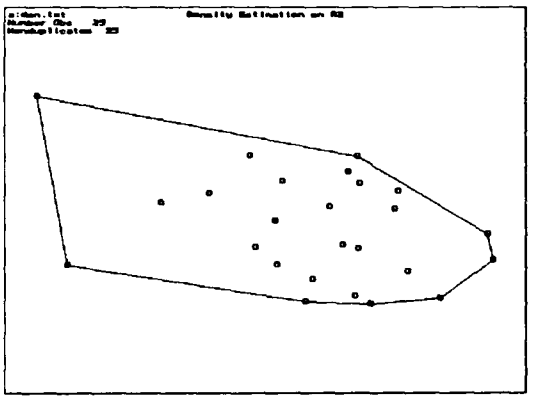


Figure 1

To derive an estimate of the density of observations on A^n it is necessary to examine sub-regions of A^n and compare the weight of observations on that region to the total weight of observations. Two general methods are currently well known, the kernel density estimation method [Rosenblatt, 1956], [Parzen, 1962] and the binning method [Scott, 1985], [Carr, 1987].

3. Density Estimation Methods

With the kernel method, a smoother with finite support, such as an Epanechnikov kernel, or smoother with infinite support, such as a Gaussian kernel, is convolved with the empirical distribution and a weighted sum of the contributions to the density at the center of the kernel is computed from the observations. This approach assumes support is continuous beyond the region defined by the set of observations. But more important, the theoretical computational complexity increases with the dimension.

The binning method tessellates the support into fixed size tiles and computes the density on a tile as the ratio of the weight of observations on each tile to the total weight of observations times the area of the tile. This method is computationally quite tractable. The problem with this method is that to get reasonably smooth, non-trivial, estimates where the data are sparse, the tiles must be relatively large. But, by making the tiles large, the fine structured features of the density are obscured where the data are closely packed.

A third method is proposed. This method is to tessellate the support A^n into tiles of varying sizes, based on the location of the observations. This might be called data directed tessellation. In this way the tiles will be large where data are sparse, and small where the data are closely packed. Furthermore, no assumptions are being made about support beyond the convex hull of the point cloud defined by the observations.

4. Tessellation of Support

Of the many possible ways that the data might direct the tessellation of $A^n \in S^d$, there is one that is unique for any inter-point distance measure l_p, $1 < p < \infty$, up to pathological cases [Preparata, 1988]. This is the high dimensional analog of the Delaunay tessellation, where $d+1$ points define a tessellating polytope, and any point in the interior of a defined polytope is closer to these $d+1$ points than to any other $d+1$ points in the tessellating point set. The Delaunay tessellation of the support A^n yields a set of convex polytopes $\{A_i^n\}$ of cardinality m, where $\emptyset = A_j \bigcap_{i \neq j} A_i$ and $A^n = \bigcup_i A_i^n$ have measure $\mu(A_i^n) > 0$, $1 \leq i, j \leq m$, and have a geometric nearest neighbor property for

points in the interior of each polytope. It is the geometric properties of the Delaunay tessellation that make it a more computationally tractable procedure for tessellating a d-dimensional product space.

Figure 2 is the Delaunay tessellation of 25 observations in S^2. Note that the $d+1$ points that define each triangle, or tessellating polytope, also define a circumscribed circle, and this circle contains no other points in the tessellating set of observations.

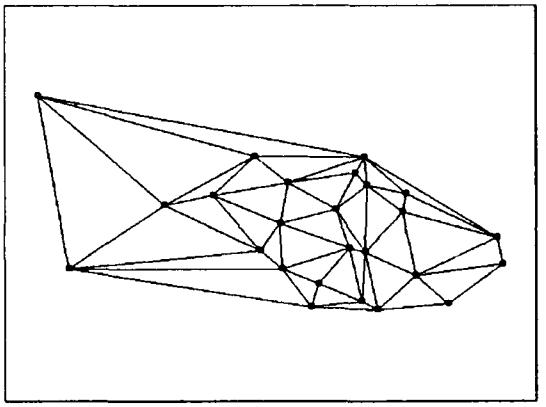


Figure 2

5. Probability Mass

The empirical probability mass on elements of the tessellating set $\{A_i^n\}$ needs to be examined. Let ∂A_i^n be the convex hull of A_i^n. Then $P[x \in A^n] = 0$ for a random $x \in S^d$. But since A_i^n is defined by elements of Y, there are $d+1$ elements of Y in ∂A_i^n. Those observations that are in the interior of A_i^n attribute all of their weight to A_i^n, $0 \le i \le m$. A question arises when the attribution of weight for points in ∂A_i^n is considered. Let w_i be the weight attributable to A_i^n from observations in the interior of A_i^n, and let w_i^* be the weight attributable from observations in ∂A_i^n. Then the total weight on a tile A_i^n is $W(A_i^n) = w_i + w_i^*$. The assignment of weight to w_i^* by the $d+1$ observations in ∂A_i^n can be computed by solving a linear system of equations to maximize the likelihood product.

6. Density Estimator

A class of density estimators can be defined on A^n by $\hat{f}(x) = \dfrac{W(A_i^n)}{W(A^n) \cdot \mu(A_i^n)}$ for $x \in A_i^n$, where $W(A^n)$ is the total weight of observations on the support A^n, and $\mu(A_i^n)$ is the integral measure of A_i^n. This class of estimators can be shown to have the maximum likelihood property [Robertson, 1967]. If the additional constraint is added to the tessellation procedure that at least λ observations are contained in each polytope, where λ is a function of the sample size, then this class of estimators can be shown to be strongly consistent, given a set of observations Y [Wegman, 1975].

7. Examples

The first two examples, Figure 3 and Figure 4, show the density estimate for a data set with 201 observations, and $\lambda = 6$.

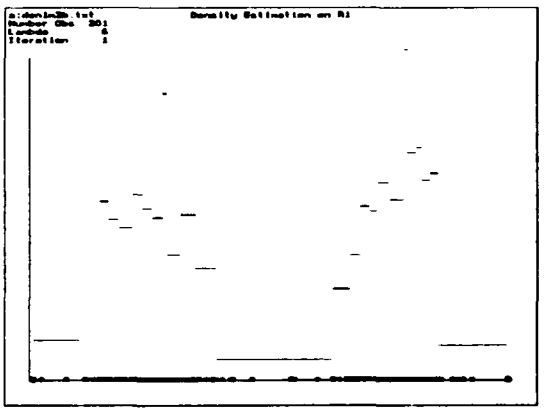


Figure 3

Figure 3 is the results of the first iteration. Each tile contains at least 6 observations. The estimate is rough but it shows that the data is at least bimodal on S^1.

Figure 4 is the average density on all subtiles of the same data set after 15 iterations of the resampling procedure. The solid bars on the ends are the result of always having to have at least 6 observations per tile. The plot is still rough, but it accurately reflects the density of the observations, without making continuity assumptions.

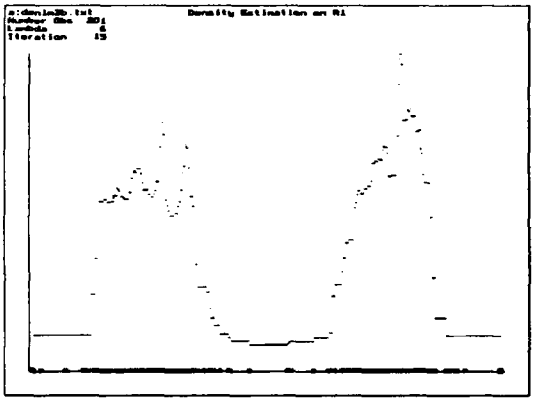


Figure 4

Figure 5 is the averaged results from 15 iterations on a data set with 400 observations, and $\lambda = 6$.

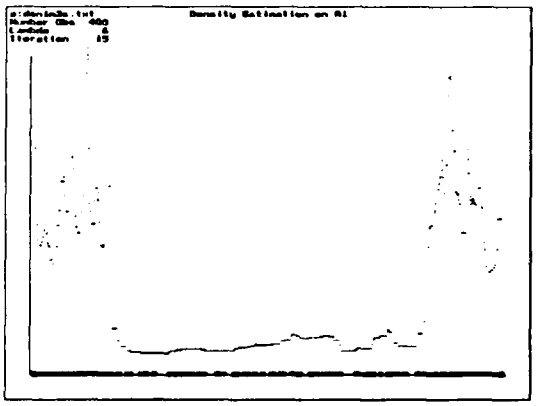


Figure 5

Again the density plot looks ragged, particularly over relatively small regions at both ends, and with a relatively smooth region between. This is caused by relatively sparse data between regions of relatively dense data. Also the density of the data would appear to be bimodal.

Figure 6 is the distribution computed by integrating the density from Figure 5 over the support. By applying integration as a natural smoother, it is reasonably clear that the data is a random sample from a step wise continuous uniform density function.

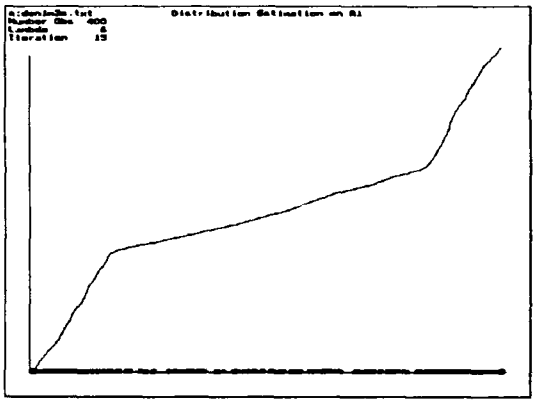


Figure 6

The next figure shows a Delaunay tessellation of S^2 by 25 observations.

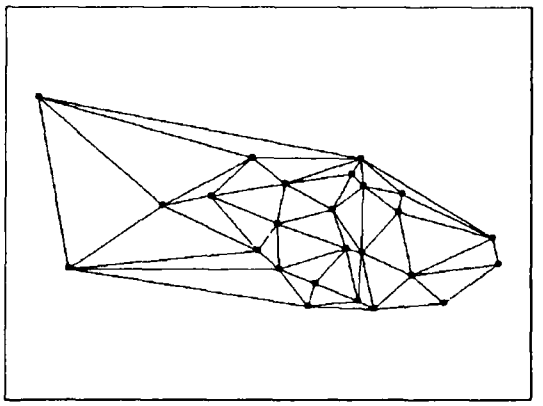


Figure 7

Figure 8 is the result of resampling for these same 25 observations, with $\lambda = 1$. After only three iterations the support has been partitioned into small subtiles where the data are dense, and relatively large subtiles where the data are sparse. The resulting average density estimate on subtiles is thus more refined where the data are dense, and less refined where the data are sparse. Note also, that the marginal densities will be piecewise continuous on the support A^n.

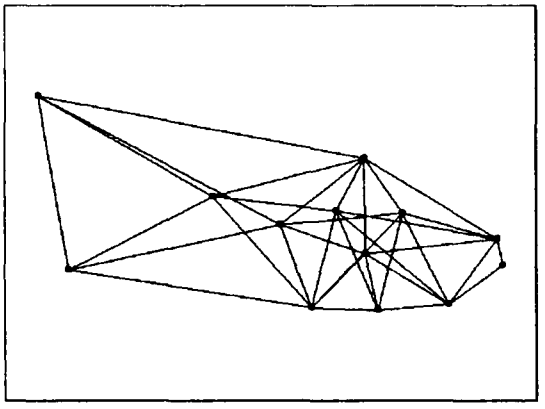


Figure 8

8. Conclusions

The adaptive density estimation procedure has much in common with the bootstrap. As such, it inherits many of the theoretical statistical properties of the bootstrap. It has additional properties derived from geometry and the tessellation procedure employed. In combination, these properties offer a rich area for theoretical statistical study, for exploratory data analysis, and for extending our understanding of computational geometry. From a computing perspective, this procedure is somewhere between binning methods and kernel density estimation methods for both compute time used and storage. The computational advantages of adaptive density estimation methods over kernel density estimation methods becomes quite dramatic as the dimension of the support increases.

Bibliography

Carr, D.B, Littlefield, W.L, Nicholson, W.L, Littlefield, J.S. (1987) "Scatterplot Matrix Techniques for Large N", J. Am. Statist. Assoc., 82, pp.424-436.

Kendall, M.G. (1961) The Geometry of *n* Dimensions, Charles Griffin & Co., London:UK.

Parzen, E. (1962) "On Estimation of a Probability Density Function and Mode",Ann. Math. Statist., 33, pp. 1065-1076.

Preparata, F.P., Shamos, M.I. (1988) Computational Geometry, Springer-Verlag, New York: NY.

Robertson, T. (1967) "On Estimating a Density Which is Measurable With Respect to a σ-Lattice", Ann. Math. Statist., 38, pp.482-493.

Rosenblatt, M. (1956) "Remarks on some Non-Parametric Estimates of a Density Function", Ann. Math. Statist., 27, pp. 832-837.

Scott, D.W. (1985) "Average Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions", Ann. Statist., 13, pp.1024-1040.

Wegman, E.J. (1975) "Maximum Likelihood Estimation of a Probability Density Function", Sankhyā: The Indian Journal of Statistics, 37, Ser. A, Pt. 2, pp. 211-224.

Non-parametric density estimation

Günter Weiss

University of Winnipeg, Winnipeg, Manitoba

Abstract

Procedures for estimating the probability density curve of the distribution from which a single sample of size n has been taken will often produce curves which are quite erratic and require much smoothing. We consider a simple method of density estimation which will produce smooth curve estimates without applying any smoother. We apply the method to both symmetric as well as skewed distributions.

1. Introduction

In this paper we consider the estimation of the probability density curve, $f(x)$, of a continuous distribution from which a single sample of size n has been taken. There are numerous parametric procedures of density estimation in which the sample is assumed to arise from some family of distributions from which it is then necessary to select the member of this family which best describes the given data set. For example, the method of maximum likelihood selects that member of the distributional family for which the probability of having obtained the given sample is maximized. In Bayes estimation a prior distributional model is combined with the information provided by the sample to produce a posterior model.

Non-parametric methods of density estimation typically make use of the spacing or clustering of the points in the data set. For large data sets, the frequency histogram and its corresponding frequency curve provide a rough estimate of the shape of the distribution. However these methods tend to be somewhat arbitrary as to the choice of class interval and method of smoothing. They also are of little use for small samples. Chambers, Cleveland, Kleiner & Tukey (1983) suggest a generalization in which the class interval (window) is allowed to move along the entire range of the data. The fraction of the entire data set in the window is a measure of the density at the center of the window. They call this the *density trace*. Since this method can be quite erratic as points enter or leave the window as the window moves along the real line, a smoother result is obtained by averaging the number of data points using a weight function (the kernel) which is a maximum near the center of the window and decreases to zero at the edges of the window. This leads to a method now called kernel density estimation. Kernel estimation does not seem to depend too much on the type of kernel used, but does vary greatly with the length of the window, or "band-width". The method also cannot properly estimate the density curve beyond the data set as the density drops suddenly to zero. See also Tapia & Thompson (1980) for more details on these different methods.

In our approach we will begin with a suitably smooth density curve which is then stretched or compressed to fit the spacing of the the data in the sample.

2. Basic method

Consider the ordered sample $x_0 = -\infty < x_1 < x_2 < \cdots < x_n < x_{n+1} = \infty$, taken from a distribution $F(x)$ with density function $f(x)$, which is what we wish to estimate. Let us define the centering points,

$$t_i = \frac{x_i + x_{i+1}}{2}, \qquad i = 1, 2, 3, \ldots, n-1. \tag{1}$$

These t_i partition $\Re$ into n sub-intervals, each containing exactly one observation. Suppose we take a continuous distribution, $G(x)$, which we shall call the trial distribution. For now, we will assume that G is location-scale invariant. If we use G to divide $\Re$ into n intervals of equal probability $\frac{1}{n}$,

$$G\left(\frac{t-\mu}{\sigma}\right) = \frac{i}{n}. \tag{2}$$

Then, let us estimate $f(x)$, by requiring that the cumulative distribution satisfy (2) and that the curve between these points be continuous and smooth. We can then estimate $f(x)$, up to a multiplicative constant, by stretching or compressing $g(x)$, the density curve corresponding to $G(x)$,

$$f(x) = \frac{c}{\sigma_i} g\left(\frac{x-\mu_i}{\sigma_i}\right), \qquad t_i < x < t_{i+1}, \tag{3}$$

where μ_i and σ_i are chosen to satisfy (2),

$$\begin{aligned} \frac{t_i - \mu_i}{\sigma_i} &= G^{-1}\left(\frac{i}{n}\right) \equiv \lambda_i, \\ \frac{t_{i+1} - \mu_i}{\sigma_i} &= G^{-1}\left(\frac{i+1}{n}\right) \equiv \lambda_{i+1}, \end{aligned} \tag{4}$$

which gives

$$\mu_i = \frac{\lambda_{i+1} t_i - \lambda_i t_{i+1}}{\lambda_{i+1} - \lambda_i},$$
$$\sigma_i = \frac{t_{i+1} - t_i}{\lambda_{i+1} - \lambda_i}. \tag{5}$$

The constant c is required since we are changing the scale as we stretch or compress the distribution to fit the quantiles, and hence we must determine c by (numerically) integrating the resulting density curve given by (3) and (5). In order to apply this procedure we will need to compute the quantiles of the trial distribution,

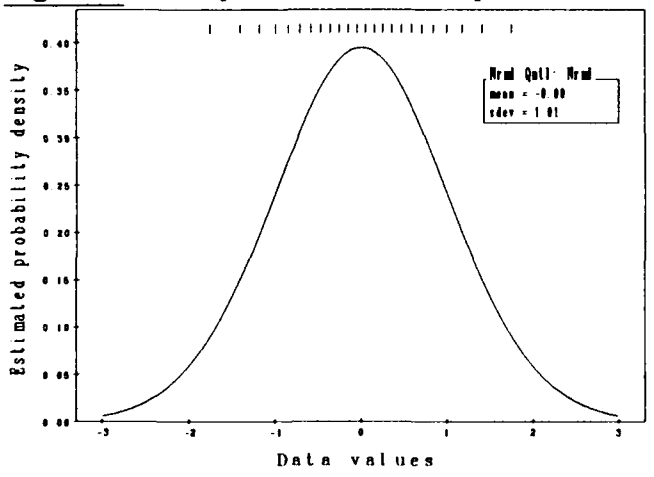
$$\lambda_i = G^{-1}\left(\frac{i}{n}\right), \qquad i = 1, 2, 3, \ldots, n-1. \tag{6}$$

Actually, since the t_i's are taken halfway between the sample values, it would be better to adjust the quantiles for this shift and use:

$$\lambda_i = G^{-1}\left(\frac{i + \frac{1}{2}}{n+1}\right), \qquad i = 1, 2, 3, \ldots, n-1. \tag{7}$$

Indeed when we used a sample of 24 "observations" consisting of the quantiles of a standard normal and applied the procedure using (3), (5) and (6) and a normal trial distribution, we obtained an density estimate which was very "normal" in shape, but with mean near zero and standard deviation of only 0.85 (both quantities numerically integrated from the density curve estimate). On the other hand, when we use (7) in place of (6), we obtained the estimate shown in figure 1, which has a computed mean of almost exactly

Figure 1: Density estimate of normal quantiles "data"



0 and standard deviation 1.01. [Using only a sample of 4 quantiles produced even worse results using (6), but again produced the standard normal when using (7).]

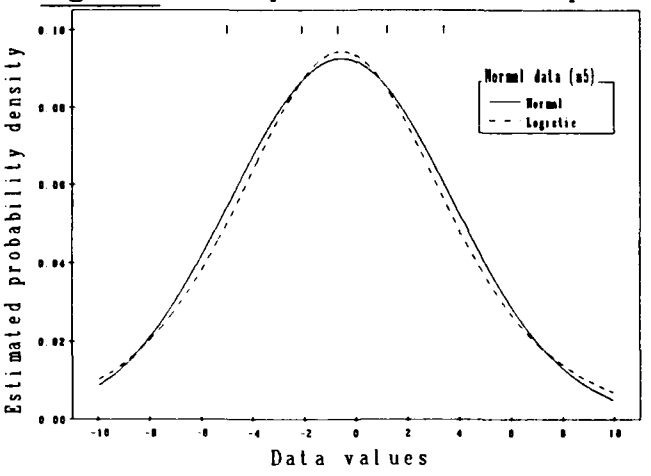
When we used a logistic trial distribution with the sample of 24 standard normal quantiles, and equations (3), (5) and (7) , we again obtained a very normal-shaped curve but with a slightly smaller standard deviation of only 0.95. This is not surprising as the logistic distribution tends to have a similar shape to the normal but is slightly narrower and has longer tails.

For an open-ended distribution with positive probability over the entire real line, $\lambda_0 = -\infty$ and $\lambda_n = \infty$, and so there is no solution possible for the first or last interval. The simple solution is to take μ_i and σ_i for the neighboring intervals:

$$\mu_0 = \mu_1,\ \sigma_0 = \sigma_1;\ \mu_n = \mu_{n+1},\ \sigma_n = \sigma_{n+1}. \tag{8}$$

A sample of five observations is generated from a normal population and the density estimate obtained using a normal trial distribution (solid line) and also using a logistic trial distribution is shown in figure 2. Both plots are very similar and both plots depend very greatly on the trial distribution since the sample is so small. For a sample $n = 2$ or $n = 3$, the estimated density would be identically the trial distribution with the location and scale determined from the sample.

Figure 2: Density estimate for normal sample

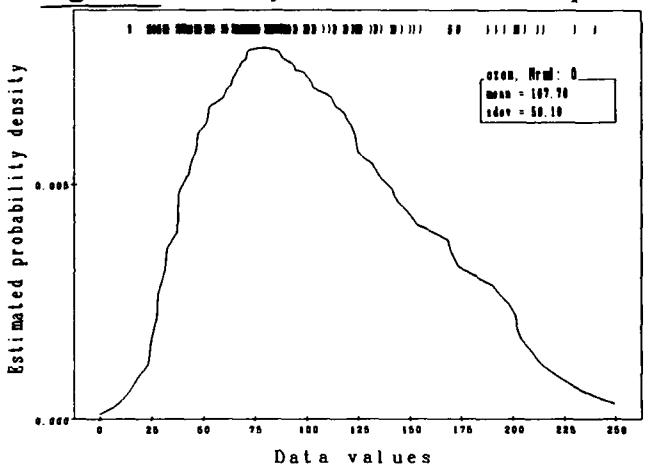


2. Smooth density estimates

We can apply the procedure to a sample showing definite skewness, as shown in figure 3. Here we get an estimate which has some skewness, but not to the extent demonstrated by the sample. The problem is that the resulting estimate is not very smooth (at the boundaries of the intervals). Also, our estimate will always be strictly decreasing and hence cannot

adequately model multi-modal situations.

Figure 3: Density estimate for skewed sample



The data used in figure 3 is the measurements of the ozone level at Stamford, Connecticut for 136 days which is used by Cleveland, Chambers, Kleiner & Tukey (1983) in their example of the density trace.

The density estimates introduced in the last two sections are only piece-wise smooth, as we are using pieces of different members of the family of trial distributions for each of the intervals. To obtain a smooth estimate, let us extend the estimate for each interval to the entire real line and average the estimates of the $n-2$ (in this case) intervals together:

$$\hat{f}(x) = \frac{1}{n-2}\sum_{i=2}^{n-1} g\left(\frac{x-\mu_i}{\sigma_i}\right), \tag{9}$$

where μ_i and σ_i are given by (5) as before. As long as $g(\cdot)$ is a proper density function, we will not need to normalize (9) as it will properly integrate to one.

To further illustrate this procedure, let us consider using the three-parameter Weibull distribution as the trial
distribution,

$$1 - e^{-(x-\nu)^p/\sigma} = G\left(\frac{(x-\nu)^p}{\sigma}\right),\ x > \nu, \sigma > 0, p > 0, \tag{10}$$

where $G(z) = 1 - e^{-x}$. The density is then estimated by

$$\hat{f}(x) = \frac{1}{n-1}\sum_{i=1}^{n-1} \frac{p_i(x-\nu)^{p_i-1}}{\sigma_i} g\left(\frac{(x-\nu)^{p_i}-1}{\sigma_i}\right), \tag{11}$$

In this case the value of ν must be determined from the data or fixed arbitrarily. The remaining two parameters, p and σ, can then be determined for each interval by

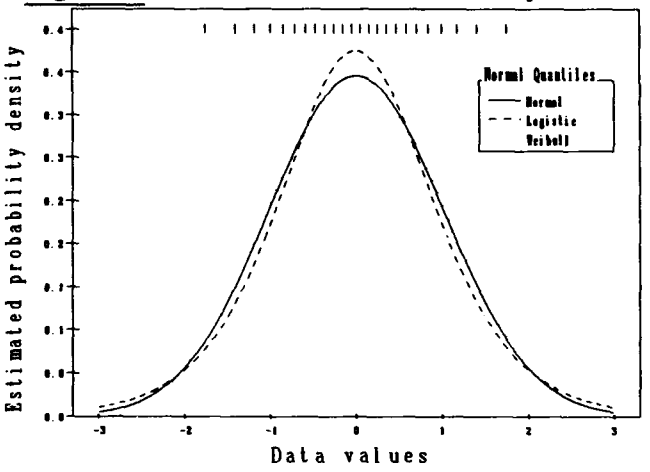
$$\frac{(t_i-\nu)^{p_i}}{\sigma_i} = G^{-1}\left(\frac{i}{n}\right) \equiv \lambda_i,$$
$$\frac{(t_{i+1}-\nu)^{p_i}}{\sigma_i} = G^{-1}\left(\frac{i+1}{n}\right) \equiv \lambda_{i+1}, \tag{12}$$

which gives

$$p_i = \frac{log\ \lambda_{i+1} - log\ \lambda_i}{log(t_{i+1}-\nu) - log(t_i-\nu)},$$
$$\sigma_i = exp\left(\frac{log\ \lambda_i log(t_{i+1}-\nu) - log\ \lambda_{i+1} log(t_i-\nu)}{log(t_{i+1}-\nu) - log(t_i-\nu)}\right). \tag{13}$$

The normal quantile data with Weibull trial distribution taking $\nu = -3$ together with the plot for normal and logistic trial distribution are shown in figure 4. All three curves have a very similar "normal" shape differing only slightly in the amount of peakedness.

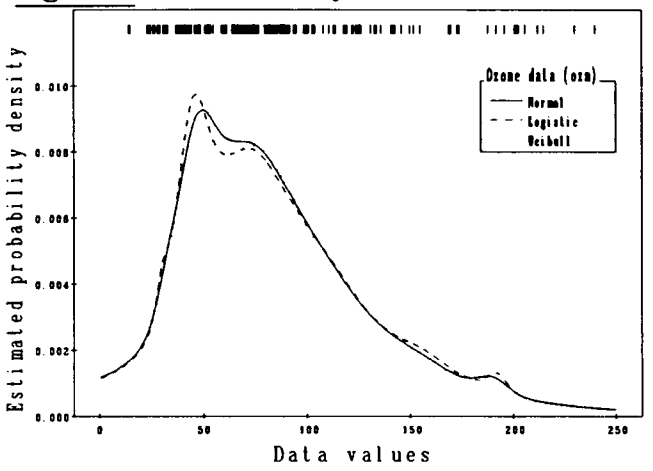
Figure 4: Smooth estimates for normal quantiles



The value of ν, which must be less than the smallest observation, can be arbitrarily assigned based on the physical situation (*e.g.*, $\nu = 0$ when data must be non-negative). This gives a graph which behaves somewhat strangely near zero, with the density estimate suddenly shooting up to a large value. This might be explained in that, although we are assuming a strictly continuous and positive distribution for the ozone level, the actual distribution may have a positive probability of a zero level. If we instead assume the density begins at some negative value, then we can estimate the probability at zero to be the area under the curve to the left of $x = 0$. (The same argument can be applied to the normal and logistic estimates.)

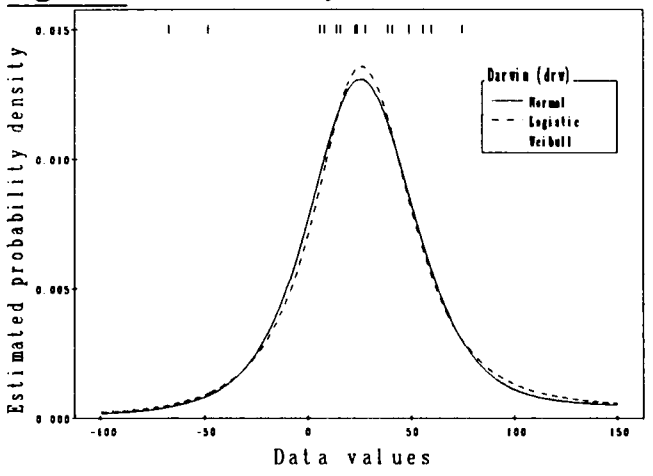
Thus, let us estimate the value of ν from the data by selecting the value of ν which will result in the maximum value for the estimated likelihood function. By inspection we find that this "maximum likelihood estimate" is obtained when $\hat{\nu} = -143$, which gives the curve shown in figure 5. Figure 5 also shows the

Figure 5: Smooth density estimates for ozone data



density estimates for normal and logistic trial distributions. Note again that all three estimates are very similar, the major differences being the peakedness of the three local maxima.

Figure 6: Smooth density estimates for Darwin data



We now consider two more "real data" examples. The Darwin data of the heights of 15 plants, as given in Box & Tiao (1973), which is often modeled as a Cauchy density:

$-67, -48, 6, 8, 14, 16, 23, 24, 28, 39, 41, 49, 56, 60, 75.$

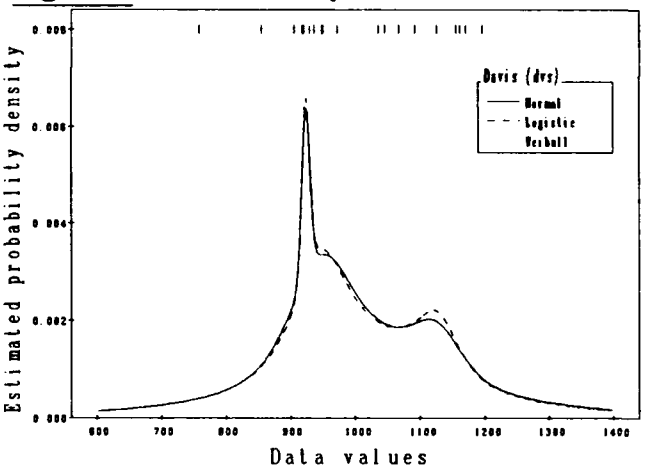
Again, using normal, logistic and maximum likelihood Weibull trial distributions we get density estimates as shown in figure 6. Again, all three estimates are very normal-shaped, except with very noticeable long tails as is characteristic of the Cauchy distribution. (Note however, this data does not necessarily arise from a Cauchy; it is only often modeled by a Cauchy due to the apparent "outliers" in the tails.)

Another data set, due to Davis (1952), consists of reliability measurements of an electronic component:

758, 855, 905, 918, 919, 920, 929, 936, 948, 950, 972, 1035, 1045, 1067, 1092, 1126, 1156, 1162, 1170, 1196.

Again using normal, logistic and maximum likelihood Weibull trial distributions we obtain very similar estimates, as shown in figure 7, which again differ mainly in the amount of peakedness near each of the local maxima.

Figure 7: Smooth density estimates for Davis data



References

Box, G. E. P., & Tiao, G. C. (1973) ***Bayesian Inference in Statistical Analysis***, John Wiley & Sons, New York, NY.

Cleveland, Chambers, J., Kleiner & Tukey, J. (1983) ***Graphical Data Analysis***. Duxbury Press, North Situate, MA.

Davis, D. J. (1952) "An analysis of some failure data," ***JASA***, 47: 113—150.

Tapia, R., & Thompson, J. R. (1980) ***Non-parametric Probability Density Estimation***, Johns Hopkins University Press, Baltimore, MD.

A COMPARISON OF TWO LARGE SAMPLE CONFIDENCE INTERVALS FOR A PROPORTION: A MONTE CARLO SIMULATION

Ken Hung
College of Business and Economics
Western Washington University
Bellingham, WA 98225

Abstract

Two pairs of confidence intervals for a proportion as in page 394 of Larson's (1982) are compared. It can be shown through computer simulation experiments that, for certain values of p, the confidence interval obtained by the approximation is superior.

1. Introduction

Computers are to the study of statistics much as test tubes to the study of chemistry. Many theoretical derivations in statistics can be investigated and confirmed by brute-force computing experiments. The purpose of this paper is to study empirically two large sample confidence intervals for a proportion in Larson's (1982) and the issue raised in Alt and Walker (1981). Alt and Walker (1981) derived analytically that, for certain ranges of p, the approximated $(1-\alpha)100\%$ confidence interval for a proportion is shorter than the unapproximated one. The basis for comparison in their paper is the expected value of squared confidence interval length, while in this paper the expected value of the confidence interval length. This should be more direct to the truth of the nature.

Section 2 discusses the theoretical and analytical aspects of the comparison of two confidence intervals. The approach and method used in this study is presented in Section 3. The results are reported in Section 4. Section 5 concludes the paper.

2. Theory

Let $x_1, x_2, \ldots, x_n$ be a sequence of n independent Bernoulli random variables with parameter p as the probability of success on each trial. Then, $X = \sum x_i = n\bar{x}$, for $i = 1, \ldots, n$, is a Binomial random variable where $\bar{x} = X/n = \hat{p}$. Given $E(X) = np$ and $Var(X) = np(1-p)$, it follows from the Central Limit Theorem that

$$\frac{n\bar{x} - np}{\sqrt{np(1-p)}} = \frac{\bar{x} - p}{\sqrt{p(1-p)/n}} \tag{2.1}$$

is distributed asymptotically as a standard normal random variable. Thus, we have

$$P(-Z_{\alpha/2} \le \frac{\bar{x} - p}{\sqrt{p(1-p)/n}} \le Z_{\alpha/2}) \doteq 1-\alpha \tag{2.2}$$

which is equivalent to

$$P\left(\left| \frac{\bar{x} - p}{\sqrt{p(1-p)/n}} \right| \le Z \right) \doteq 1 - \alpha \tag{2.3}$$

where $Z = Z_{\alpha/2}$. The equation

$$(\bar{x} - p)^2 \le Z^2 p(1-p)/n \tag{2.4}$$

as a quadratic inequality in p has two real and unequal roots. These two roots are the desired confidence limits for p. Let Q_1 and Q_2 denote the lower and upper confidence limits respectively. Use of the quadratic formula yields

$$Q_1 = \frac{\bar{x} + Z^2/n - (Z/\sqrt{n})\sqrt{\bar{x}(1-\bar{x}) + Z^4/n^2}}{1 + Z^2/n} \tag{2.5}$$

$$Q_2 = \frac{\bar{x} + Z^2/n + (Z/\sqrt{n})\sqrt{\bar{x}(1-\bar{x}) + Z^4/n^2}}{1 + Z^2/n}$$

When n is large and for reasonable $(1-\alpha)$, Z^2/n should approach zero. Therefore, the approximated large sample confidence limits are

$$L_1 = \bar{x} - (Z/\sqrt{n})\sqrt{\bar{x}(1-\bar{x})} \tag{2.6}$$

$$L_2 = \bar{x} + (Z/\sqrt{n})\sqrt{\bar{x}(1-\bar{x})}$$

The above approximated confidence limits can also be derived from the asymptotical standard normal

random variable

$$\frac{\bar{x} - p}{\sqrt{\bar{x}(1-\bar{x})/n}} = \frac{\hat{p} - p}{\sqrt{\hat{p}(1-\hat{p})/n}} \qquad (2.7)$$

since $E\left(\frac{\bar{x}(1-\bar{x})}{n}\right) = (1 - \frac{1}{n})\frac{p(1-p)}{n}$ is

asymptotically unbiased. See Alt and Walker (1981). For the confidence interval in (2.5), we can have

$$(Q_2-Q_1)^2 = \bar{x}(1 - \bar{x})\frac{4nZ^2}{(n + Z^2)^2} + \frac{Z^4}{(n + Z^2)^2} \qquad (2.8)$$

For the confidence interval in (2.6), we can have

$$(L_2 - L_1)^2 = \bar{x}(1 - \bar{x})\frac{4Z^2}{n} \qquad (2.9)$$

Let $Q=E(Q_2 - Q_1)^2$ and $L=E(L_2 - L_1)^2$. Solving for the inequality $Q > L$ gives us the ranges of p values that satisfy the inequality. Since $E(\bar{x}) = p$ and $E(\bar{x}^2) = Var(\bar{x}) + [E(\bar{x})]^2 = p(1-p)/n + p^2$, we can easily show that

$$E[\bar{x}(1 - \bar{x})] = \frac{p(1-p)(n-1)}{n} \qquad (2.10)$$

$$E(Q_2-Q_1)^2=\frac{p(1-p)(n-1)}{n}\frac{4nZ^2}{(n+Z^2)^2}+\frac{z^4}{(n+Z^2)^2}=Q \qquad (2.11)$$

and

$$E(L_2 - L_1)^2 = \frac{p(1-p)(n-1)}{n}\frac{4Z^2}{n} = L \qquad (2.12)$$

Simplify

$$\frac{p(1-p)(n-1)}{n}\frac{4nZ^2}{(n+Z^2)^2}+\frac{z^4}{(n+Z^2)^2}>\frac{p(1-p)(n-1)}{n}\frac{4Z^2}{n} \qquad (2.13)$$

to $$\frac{n}{4(1-1/n)(2n+Z^2)} > p(1-p). \qquad (2.14)$$

When n is large and Z^2 ignored, the left hand term of (2.14) can be approximated by $\frac{1}{8}$. Thus, a quadratic inequality in p

$$p^2 - p + \frac{1}{8} > 0 \qquad (2.15)$$

can be solved with two roots $p = .146$ and $p = .854$. This means that for $p < .146$ and $p > .854$, the expected squared length of (2.5) is greater than that of (2.6).

3. Method

A Fortran program is written so as to call IMSL subroutine GGBN for generating binomial random numbers and to compute the average lengths for the two pairs of confidence interval in (2.5) and (2.6). The Z value is set to be 1.96 so as to give the same 95% confidence intervals for both pairs. The number of trials n in each experiment is increased from 50 to 250 by 50 at different proportion values in the inclusive (0,1) range. One hundred experiments are performed generating 100 binomial random numbers (X_j for j= 1, ..., 100) for each fixed number of trials (n = 50, 100, 150, 200, 250) at different proportion values (p = .05, .10, ..., .95). Specifically, for instance, X_1=2, X_2= 3, X_3=3,....., X_{100}=4 for, say, n = 50 at p=.05 . The computational scheme is outlined below:

$$\bar{x}_j = X_j/n = \hat{p}_j \text{ for } j= 1,2,.....,100 \qquad (3.1)$$

$$Q_j = Q_{2j} - Q_{1j},\ E(Q) = \sum Q_j /100 \qquad (3.2)$$

$$L_j = L_{2j} - L_{1j},\ E(L) = \sum L_j /100 \qquad (3.3)$$

Hence, two lengths of confidence interval are computed for comparison. Meantime, the lower limits are checked for negative values as it is meaningless to have negative proportion vlaues.

4. Results

The results are reported below in tabular forms. All numbers are significantly different from zero at $\alpha <$ .0000001 .

Table 1 Confidence Lengths for p = .05 to .95

	n = 50		n = 100	
p=	$E(Q)$=	$E(L)$=	$E(Q)$=	$E(L)$=
.05	.129240	.111604	.086883	.081212
.10	.160865	.153494	.118447	.116699
.15	.196048	.196297	.138551	.138601
.20	.216052	.219484	.155206	.156499
.25	.228749	.233905	.167382	.169501
.30	.245147	.252505	.175688	.178338
.35	.251126	.259244	.182566	.185642
.40	.259527	.268688	.187935	.191335

.45	.262571	.272100	.190823	.194393
.50	.264110	.273823	.191229	.194823
.55	.263677	.273339	.190303	.193842
.60	.258847	.267923	.187152	.190505
.65	.252268	.260526	.182632	.185712
.70	.242492	.249517	.176238	.178923
.75	.228482	.233639	.164684	.166623
.80	.213716	.216753	.153372	.154526
.85	.191278	.190726	.138976	.139064
.90	.162923	.156850	.119114	.117444
.95	.127037	.107727	.090024	.094820

	n=150		n=200	
p=	$E(Q)$=	$E(L)$=	$E(Q)$=	$E(L)$=
.05	.073089	.070345	.059756	.057695
.10	.096291	.095323	.083232	.082603
.15	.113671	.113716	.099044	.099094
.20	.127443	.128165	.110006	.110455
.25	.135893	.136994	.118682	.119423
.30	.144120	.145572	.125345	.126298
.35	.150863	.152591	.130473	.131584
.40	.154345	.156211	.133971	.135186
.45	.156761	.158723	.136266	.137549
.50	.157607	.159602	.137054	.138360
.55	.156862	.158828	.136222	.137503
.60	.154620	.156498	.134362	.135589
.65	.150248	.151951	.130681	.131798
.70	.144797	.146278	.125889	.126860
.75	.136504	.137632	.118849	.119596
.80	.125992	.126643	.110431	.110897
.85	.114032	.114097	.098272	.098292
.90	.094655	.093598	.083051	.082419
.95	.072323	.069487	.062522	.060699

	n=250	
P=	$E(Q)$=	$E(L)$=
.05	.053906	.052499
.10	.074071	.073608
.15	.088089	.088108
.20	.097574	.097872
.25	.107009	.107560
.30	.112697	.113392
.35	.117070	.117871
.40	.120240	.121117
.45	.122119	.123040
.50	.122779	.123715
.55	.122190	.123112
.60	.120358	.121237
.65	.117629	.118444

Table 2. Confidence Lengths for p=.141 *to* .150

	n=50		n=100	
p=	$E(Q)$=	$E(L)$=	$E(Q)$=	$E(L)$=
.141	.188695	.187642	.134747	.134477
.142	.185552	.183980	.133452	.133074
.143	.192676	.192358	.135125	.134881
.144	.190294	.189785	.137330	.137280
.145	.190473	.189798	.137713	.137706
.146	.196234	.196575	.138436	.138469
.147	.184484	.182840	.136524	.136401
.148	.193216	.193015	.137517	.137468
.149	.191201	.190609	.138854	.138921
.150	.197258	.197718	.139562	.139674

	n=150		n=200	
p=	$E(Q)$=	$E(L)$=	$E(Q)$=	$E(L)$
.141	.111916	.111867	.097674	.097671
.142	.110867	.110761	.096849	.096812
.143	.109992	.109845	.097173	.097148
.144	.111087	.110995	.096666	.096622
.145	.110784	.110675	.098194	.098211
.146	.111497	.111430	.097714	.097709
.147	.111510	.111437	.096706	.096664
.148	.112772	.112768	.098053	.098065
.149	.113501	.113536	.097940	.097947
.150	.113884	.113940	.098267	.098286

	n=250	
p=	$E(Q)$=	$E(L)$=
.141	.086157	.086115
.142	.086490	.086459
.143	.086628	.086600
.144	.086604	.086575
.145	.086900	.086881
.146	.087078	.087064
.147	.086871	.086852
.148	.087173	.087163
.149	.088275	.088299
.150	.088051	.088069

Table 3. Confidence Lengths for p=.848 *to* .857

	n=50		n=100	
p=	$E(Q)$=	$E(L)$=	$E(Q)$	$E(L)$
.848	.194058	.194033	.138874	.138952
.849	.195307	.195373	.137464	.137415
.850	.191125	.190103	.139139	.139223
.851	.195690	.195666	.135951	.134787
.852	.195164	.195387	.136982	.136902

.853	.193889	.193788	.136169	.136023
.854	.191018	.190548	.137492	.137440
.855	.188169	.187087	.134055	.133727
.856	.189168	.188288	.136403	.136265
.857	.188540	.187527	.136354	.136236

	n=150		n=200	
p=	$E(Q)$=	$E(L)$=	$E(Q)$=	$E(L)$=
.848	.114195	.114264	.099455	.099521
.849	.113754	.113802	.099357	.099421
.850	.112745	.112745	.098589	.098622
.851	.113870	.113926	.098091	.098104
.852	.113628	.113668	.098187	.098204
.853	.111333	.111254	.098315	.098336
.854	.111889	.111840	.097823	.097828
.855	.111643	.111579	.097202	.097178
.856	.110922	.110815	.097405	.097390
.857	.111827	.111776	.096491	.096441

	n=250	
p=	$E(Q)$=	$E(L)$=
.848	.089044	.089091
.849	.088881	.088923
.850	.088370	.088397
.851	.088361	.088387
.852	.087940	.087954
.853	.088464	.088494
.854	.086444	.096412
.855	.087600	.087604
.856	.085962	.085912
.857	.085957	.085909

Table 4 Percentage of $L_1 < 0$

p\ n=	50	100	150	200	250
.05	68	25	3	2	0
.10	30	0	0	0	0
.11	26	1	0	0	0
.12	16	0	0	0	0
.13	13	1	0	0	0
.14	3	0	0	0	0
.15	6	0	0	0	0
.16	6	0	0	0	0
.17	0	0	0	0	0
.18	1	0	0	0	0
.19	0	0	0	0	0
.	.	.	.	.	.
.	.	.	.	.	.
.99	0	0	0	0	0

5. Conclusion

It can be inferred from data presented in results that $E(L) > E(Q)$ in general. However, in the ranges of the proportion values in Table 5 below, $E(Q)$ is greater than $E(L)$.

Table 5 Ranges of $E(Q) > E(L)$

n=	$p <$	$p >$
50	.146	.852
100	.146	.850
150	.149	.852
200	.145	.854
250	.149	.855

This is quite consistent with the conclusion in Alt and Walker (1981). The difference between the two papers is that this paper uses the direct expected length while that paper uses the expected squared length.

The problem of negative lower confidence limit is only with L_1 when the proportion value is low and the number of trials n is as small as 50. If the number of trials n is above 250, the problem will disappear entirely. See Table 4.

References

Alt, Frank B. and James W. Walker (1981). "A Comparison of two large sample confidence intervals for a proportion," Proceedings, Tenth Annual Meeting, Northeast Conference, American Institute for Decision Sciences, 118-120.

Larson, Harold J. (1982). Introduction to Probability Theory and Statistical Inference, 3rd Ed., New Yoark, John Wiley and Sons.

Reconstruction of Evolutionary Trees from Pairwise Distributions on Current Species

Joseph T. Chang and John A. Hartigan
Department of Statistics
Yale University

Abstract

Suppose that the evolution of a character possessed by a number of current species is modelled as a Markov random field on an evolutionary tree. Suppose that for each pair of current species we know the joint probability distribution of the pair of characters possessed by that pair of species. We give conditions under which the evolutionary tree can be reconstructed from knowledge of these pairwise joint distributions, that is, conditions under which there is only one evolutionary tree topology consistent with the given pairwise distributions. In this way we establish consistency of a method for reconstructing evolutionary trees using pairwise distributions estimated from observed homologous DNA sequences.

1 Introduction

Evolutionary relationships among species are commonly conceptualized in terms of an "evolutionary tree." A tree consists of nodes and arcs. The *degree* of a node is the number of arcs incident to the node. Nodes of degree one are *terminal nodes*, and nodes of higher degree are *internal nodes*. In an evolutionary tree, the terminal nodes are labelled by current species observable today, and the internal nodes correspond to ancestral species. We assume speciation events occur at internal nodes, so that we do not allow nodes of degree two. The scientific problem of interest to us is to infer the evolutionary tree relating a given set of current species. This inference is to be based on data, which might typically be a set of observed DNA sequences, one from each of the given current species. For most of the paper, we will restrict our attention to the topology of the tree together with the labels of the terminal nodes. In particular, we are not interested in length of time, direction of time, or the root of the tree.

Let T denote a finite set of current species, let C denote a finite set of characters, and for each $t \in T$ let X_t denote the character possessed by species t. For example, C might be the set of four nucleotides, and X_t might identify the nucleotide occupying a particular site in the DNA of a representative of species t. We consider $\{X_t : t \in T\}$ to be random variables generated by a Markov random field model on an evolutionary tree. To describe the model, we begin with the tree $\mathcal{T} = (S, A)$, characterized by its set of nodes (or species) S and its set of arcs A. S may be decomposed into the union $S = T \cup N$ of the set T of terminal nodes and the set N of nonterminal nodes; since current species correspond to terminal nodes, there is no conflict with the notation T introduced above. Each arc $a \in A$ is undirected and may be represented as a subset $\{r, s\}$ containing two distinct nodes $r, s \in S$. For each $s \in S$ let X_s be a random variable taking values in C. We assume that $\{X_s : s \in S\}$ is a Markov random field on $\mathcal{T}$, which means that for each $s \in S$ the conditional distribution of X_s given all of the other values $\{X_r : r \neq s\}$ is the same as the conditional distribution of X_s given just the values $\{X_r : \{r, s\} \in A\}$ at the "neighbors" of s. This completes the description of the probabilistic model for the evolution of a single character. In general we observe n characters for each species. In this case, we make the standard but undoubtedly unrealistic assumption that distinct characters are independent and identically distributed (*iid*), that is, we imagine that $X^1, \ldots, X^n$ are *iid*, where each $X^i = \{X^i_s : s \in S\}$ is a Markov random field on $\mathcal{T}$.

The following brief remarks about methods of evolutionary tree reconstruction are intended to provide some context for this work; the excellent survey of Felsenstein (1988) should be consulted for more background and references. The most popular method is probably the *parsimony* method of Camin and Sokal (1965), which chooses a tree in which characters can be assigned to nodes so that the number of changes of character across arcs of the tree is minimal over all trees. This method has the very considerable virtue of ease of implementation. However, the unfortunate truth observed by Felsenstein (1978) is that parsimony is *inconsistent*; in fact, Felsenstein exhibited an example in which the probability that parsimony would choose an incorrect tree approached one as the number n of observed characters per species approached infinity. A maximum likelihood method for the present model was considered by Barry and Hartigan (1987b). This method overcomes parsimony's defect of inconsistency, at the cost of a great increase in computational difficulty. The distance method of Barry and Hartigan (1987a), which will be described more fully below, is intermediate between parsimony and maximum likelihood in terms of computational difficulty. The question that originated the present investigations was whether the distance method is consistent. This question will be addressed in section 3.

2 Identifiability of the Tree

A principal ingredient in the consistency proof is an "identifiability" result that says that under the assumptions of our Markov model and certain other conditions, distributions of pairs of the form (X_t, X_u), where t and u are terminal nodes, determine the evolutionary tree. This result in turn follows from Lemma 1 below, which says that knowing the values of an "additive function" on pairs of terminal nodes of a tree is enough to determine the tree.

The statement of Lemma 1 requires some definitions. Let $\mathcal{T}_1 = (S_1, A_1)$ and $\mathcal{T}_2 = (S_2, A_2)$ be two trees with $S_1 = T \cup N_1$ and $S_2 = T \cup N_2$, so that the terminal nodes of $\mathcal{T}_1$ and $\mathcal{T}_2$ are the same. We say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are *equivalent* if there is a bijective "relabelling" function $\rho : S_1 \to S_2$ such that $\rho(t) = t$ for all $t \in T$ and $A_2 = \{\{\rho(r), \rho(s)\} : \{r, s\} \in A_1\}$. In this case we write $\mathcal{T}_1 \sim \mathcal{T}_2$. Thus, $\mathcal{T}_1 \sim \mathcal{T}_2$ means that $\mathcal{T}_1$ and $\mathcal{T}_2$ are the same up to a possible relabelling of nonterminal nodes.

Next, for a given tree $\mathcal{T} = (S, A)$, let $\vec{A} = \{(r, s) : \{r, s\} \in A\}$ be the set of *directed arcs* of $\mathcal{T}$. Then for all distinct $r, s \in S$ either $\pi(r, s) := \{(r, s)\} \subset \vec{A}$ or there is a unique $n \geq 1$ and a unique sequence $s_1, \ldots, s_n$ of distinct nodes such that $\pi(r, s) := \{(r, s_1), (s_1, s_2), \ldots, (s_n, s)\} \subset \vec{A}$. This defines $\pi(r, s)$, the *path from r to s*. We say that a function $f : S \times S \to \mathbb{R}$ is *additive* on the tree $\mathcal{T}$ if for all $r, s \in S$ we have

$$f(r, s) = \sum_{a \in \pi(r,s)} f(a),$$

with the sum being defined to be 0 if $r = s$.

Lemma 1 *Let $\mathcal{T}_1 = (S_1, A_1)$ and $\mathcal{T}_2 = (S_2, A_2)$ be two evolutionary trees with the same set of terminal nodes T. Suppose there exist functions $f_1 : S_1 \times S_1 \to \mathbb{R}$ and $f_2 : S_2 \times S_2 \to \mathbb{R}$ such that*

1. *$f_i(r, s) + f_i(s, r) \neq 0$ for all $\{r, s\} \in A$ and $i = 1, 2$*
2. *f_i is additive on $\mathcal{T}_i$ for $i = 1, 2$*
3. *$f_1(t, u) = f_2(t, u)$ for all $t, u \in T$.*

Then $\mathcal{T}_1 \sim \mathcal{T}_2$.

Results appearing in the papers of Dobson (1974) and Sattath and Tversky (1977) are clearly closely allied but apparently not the same. They focus on existence of trees satisfying certain conditions, while we are interested in uniqueness. We also find the fact that our additive function need not be nonnegative to be interesting.

The key to the identifiability result mentioned above is the notion of distance introduced by Barry and Hartigan (1987a), which takes the form

$$d(r, s) = -(1/4) \log[\det(P^{rs})]$$

for four-valued characters, where P^{rs} is the Markov transition matrix whose $(i, j)th$ entry is $P\{X_s = j | X_r = i\}$. For the Markov model we have assumed, d is an additive function. The identifiability result is then just the statement obtained by taking the additive function in Lemma 1 to be Barry and Hartigan's distance. The conditions required are

$$\det(P^{rs}) > 0 \quad \text{for} \quad \{r, s\} \in A \qquad (2.1)$$

and

$$\det(P^{rs})\det(P^{sr}) < 1 \quad \text{for} \quad \{r,s\} \in A. \qquad (2.2)$$

Under conditions (2.1) and (2.2), the pairwise distributions of characters at terminal nodes determine the evolutionary tree. Condition (2.2) corresponds to condition 1 of the lemma. To get an idea of its significance, note that an example of a situation it rules out is $P^{rs} = P^{sr} = I$ for two internal nodes r and s joined by an arc. This is reasonable, since in such a case we could eliminate the arc $\{r,s\}$ and combine nodes r and s into one node without changing any probability distributions at the terminal nodes of the tree. Condition (2.1) ensures that the logarithm in Barry and Hartigan's distance is defined. This would presumably hold in biologically realistic models, for example, models in which characters evolve as a Markov chain in continuous time. In any case, both conditions (2.1) and (2.2) may be relaxed by the device of using the distance $d^*(r,s) = -(1/4)\log|\det(P^{rs})|$ in place of d. The resulting conditions would be $\det(P^{rs}) \neq 0$ and $|\det(P^{rs})\det(P^{sr})| < 1$ for $\{r,s\} \in A$.

3 Consistency

The method Barry and Hartigan (1987a) propose for choosing an evolutionary tree from given data applies the least squares idea of Cavalli-Sforza and Edwards (1967) in the following manner. For each ordered pair (t,u) of terminal nodes, form the estimated distances

$$\hat{d}(t,u) = -(1/4)\log[\det(\hat{P}^{tu})], \qquad (3.1)$$

where $\hat{P}^{tu}$ is the usual empirical estimate of P^{tu}. For a candidate tree $\mathcal{T} = (S,A)$ under consideration, for each $(r,s) \in \vec{A}$ introduce a variable x_{rs}. Define the "departure from additivity" of the tree $\mathcal{T}$ to be the minimum of the quantity

$$\sum_{t,u\in T}\left(\hat{d}(t,u) - \sum_{(r,s)\in\pi(t,u)} x_{rs}\right)^2$$

over all possible values of the variables x_{rs}. Choose the tree having the smallest departure from additivity.

To state the consistency result, let $\mathcal{T}$ denote the true evolutionary tree, and as usual assume that $X^1, X^2, \ldots$ are *iid* Markov random fields on $\mathcal{T}$. Suppose that $\mathcal{T}^n$ denotes the estimate given by a tree reconstruction method when applied to the data $X^1, \ldots, X^n$. We say that the method is *strongly consistent* if with probability one there is a finite N such that $\mathcal{T}^n = \mathcal{T}$ for all $n \geq N$.

Theorem 2 *Suppose the true evolutionary tree $\mathcal{T} = (S,A)$ is bifurcating, that is, all internal nodes have degree 3. Then under conditions (2.1) and (2.2), the method of Barry and Hartigan (1987a) is strongly consistent.*

The assumption that the true tree is bifurcating rules out nodes of degree higher than 3. This restriction is necessary for the following reason. Lemma 1 states that different trees cannot have exactly the same distances between pairs of terminal nodes. However, if the true tree has nodes of degree higher than 3, then there are different trees that may have distances arbitrarily close to the true distances. If the true tree is bifurcating and the true model satisfies the conditions (2.1) and (2.2), then there are no such different trees.

4 No Information in Asymmetry

The distance d is asymmetric in general: $d(r,s) \neq d(s,r)$. Since the symmetrized distance function

$$\tilde{d}(r,s) = [d(r,s) + d(s,r)]/2$$

is also additive, one could work with $\tilde{d}$ rather than d, and effectively cut in half the number of equations and unknowns in each least squares calculation. However, replacing d by $\tilde{d}$ involves ignoring some of the information in the data, so that one might suspect that we would pay for the gain in computational simplicity by sacrificing efficiency. It turns out that as far as the method of Barry and Hartigan is concerned, no efficiency at all is lost by symmetrizing. The reason is contained in the following result.

Proposition 3 *Let a tree $\mathcal{T}$ having terminal nodes T be given. Suppose we are also given the estimated distances $\hat{d}(t,u)$ of (3.1) for all $t,u \in T$. For additive functions f on $\mathcal{T}$ define $S(f)$ and $\tilde{S}(f)$ to be*

$$\sum_{t,u\in T}[f(t,u) - \hat{d}(t,u)]^2$$

and

$$\sum_{t,u\in T}\left(\frac{f(t,u)+f(u,t)}{2}-\frac{\hat{d}(t,u)+\hat{d}(u,t)}{2}\right)^2,$$

respectively. Then we have

$$\inf_f S(f)=\inf_f \tilde{S}(f),$$

where the infima are taken over all functions f additive on T.

5 Identifiability of the Full Model

Although section 2 showed that pairwise distributions over terminal nodes determine a tree, it is interesting that such pairwise distributions do not determine the full model, that is, they do not determine the Markov transition matrices P^{rs} for $\{r,s\}\in A$. This can be seen in the smallest nontrivial case: a tree having 3 terminal nodes $T=\{a,b,c\}$ and one nonterminal node $N=\{m\}$, say. Begin with an "original" model having marginal probability vector π^m at node m and Markov transition matrices P^{ma}, P^{mb}, and P^{mc}. These specifications determine the complete joint distribution of the Markov random field $\{X_a,X_b,X_c,X_m\}$, and in particular the Markov transition matrices P^{am}, P^{bm}, and P^{cm}. Let $\mathbf{1}$ denote a vector of ones and let a prime "(')" denote transpose. Then it turns out that if R is an invertible matrix satisfying the conditions

1. $R\mathbf{1}=\mathbf{1}$
2. $(R^{-1})'\pi^m R^{-1}=\mathrm{diag}(\tilde{\pi}^m)$ for some probability vector $\tilde{\pi}^m$
3. RP^{mi} and $P^{im}R^{-1}$ have nonnegative entries for $i=a,b,c$,

then the model having marginal probability vector $\tilde{\pi}^m$ at node m and Markov transition matrices $\tilde{P}^{mi}=RP^{mi}$ for $i=a,b,c$ has the same pairwise distributions over the terminal nodes as the original model. It is not difficult to find such examples; in fact, two characters are enough.

On the other hand, under conditions, the joint distribution of the values $\{X_t:t\in T\}$ at all of the terminal nodes is enough to determine the full model, at least for two characters, as the following result shows.

Proposition 4 *Let $\mathcal{T}_1=(S_1,A_1)$ and $\mathcal{T}_2=(S_2,A_2)$ be two evolutionary trees with the same set of terminal nodes T. For i=1 and 2, let $X_i=\{X_i(s):s\in S_i\}$ be a Markov random field on $\mathcal{T}_i$ taking on two values. Suppose that conditions (2.1) and (2.2) hold with P^{rs}, P^{sr}, and A replaced by P_i^{rs}, P_i^{sr}, and A_i, respectively, for i=1,2. Suppose also that the joint distributions of $\{X^1(t):t\in T\}$ and $\{X^2(t):t\in T\}$ are the same. Then $\mathcal{T}_1\sim\mathcal{T}_2$, and we have equality of the full joint distributions of $\{X_1(s):s\in S_1\}$ and $\{X_2(\rho(s)):s\in S_1\}$, where ρ is the "relabelling" function in the definition of the equivalence $\mathcal{T}_1\sim\mathcal{T}_2$.*

We conjecture that a similar result holds for the case of more than two characters.

6 References

1. Barry, D. and Hartigan, J. A. (1987a). Asynchronous distance between homologous DNA sequences. *Biometrics* **43** 261–276.
2. Barry, D. and Hartigan, J. A. (1987b). Statistical analysis of hominoid molecular evolution. *Statistical Science* **2** 191–210.
3. Camin, J. H. and Sokal, R. R. (1965). A method for deducing branching sequences in phylogeny. *Evolution* **19** 311–326.
4. Cavalli-Sforza, L. L. and Edwards, A. W. (1967). Phylogenetic analysis: models and estimation procedures. *Evolution* **32** 550-570.
5. Dobson, A. J. (1974). Unrooted trees for numerical taxonomy. *Journal of Applied Probability* **11** 32–42.
6. Felsenstein, J. (1978). Cases in which parsimony or compatibility methods will be positively misleading. *Syst. Zool.* **27** 401–410.
7. Felsenstein, J. (1988). Phylogenies from molecular sequences: inference and reliability. *Annu. Rev. Genet.* **22** 521–565.
8. Sattath, S. and Tversky, A. (1977). Additive similarity trees. *Psychometrika* **42** 319–345.

Quantitative Trait Loci in *Brassica rapa*

Brian S. Yandell*
Departments of Horticulture and Statistics, University of Wisconsin–Madison

Abstract

This paper briefly examines current methodology for developing genetic linkage maps and using them to find loci for quantitative traits (QTL). Maximum likelihood interval mapping is viewed as an extension of classical least squares methods when the trait of interest is normally distributed and located near a genetic marker. Some problems in finding multiple loci for a quantitative trait are examined for days to budding as measured on F2 plants from a *Brassica rapa* cross. Relevant design aspects of molecular biology experiments are briefly noted.

Introduction

Last year colleagues in the plant sciences approached me with questions about recently developed programs for locating quantitative traits on genetic linkage maps (Lander and Botstein, 1989). They wanted to know how this related to "classical" approaches, and whether this new maximum likelihood approach was more appropriate.

The present study concerns the cross of two varieties of *Brassica rapa*, a Michihili Chinese cabbage (M) female with a Spring broccoli (S) male plant, producing a single F1 offspring which was self-pollinated. The resultant F2 seeds were germinated, yielding 95 plants which were measured for various phenotypic traits (observable characteristics such as days to first flower, days to budding, etc.). The F2s were assayed by 297 restriction fragment length polymorphisms, or RFLPs, which were drawn from previous studies of *Brassica*, DNA from both grandparents or the F1 parent, or selfed progeny of same.

A genetic linkage map was constructed (Song et al., 1990) which locates markers relative to one another based on the frequency of genetic recombination (crossover of chromosome pairs during meiosis) on the ten chromosome pairs in this genome. They measured the association between the RFLP patterns for each marker and the phenotypic traits and used the genetic linkage map to find probable sites, or loci, of genes controlling those traits (Song, Slocum and Osborn, 1991).

The purpose of this paper is to examine the statistical properties of such quantitative trait loci (QTL). In particular we show the connection between the classical regression model at markers and the maximum likelihood interval mapping method presented in Lander and Botstein (1989) and discuss some inferential questions concerning confidence intervals and finding multiple loci (major and minor genes) which control days to budding.

RFLPs and Linkage Maps

Chromosomes come in pairs, and offspring inherit one of a pair from each parent. Any locus on a chromosome pair has two "alleles," or forms of DNA, one from each parent. The F1 was "heterozygous" (had two different alleles) at each marker locus and presumably at all QTL. F2 plants inherit (via F1) both alleles at a locus from one grandparent (MM or SS) or one from each (MS). Genetic recombination can lead to different allele types along the same chromosome, which is exploited to generate RFLP linkage maps (Lander and Botstein, 1989).

RFLP involves digesting DNA with an enzyme and using discrepant fragment lengths as markers for genetic differences among individuals. The enzyme cuts DNA adjacent to a specific base pair pattern, say `ACGTAT`. A change (mutation or recombination) in this restriction site for one variety (say `ACTTAT`) would be missed by the enzyme, resulting in one long fragment rather than two shorter ones—a polymorphism. Other forms of DNA rearrangement between restriction sites (e.g. insertion/deletion/transposition) can also create polymorphisms. DNA fragments are separated by size on a Southern blot and "probed" by ^{32}P-labelled DNA pieces which bond to homologous

*Research supported by USDA-CSRS grant 511-100. Conference accomodations supported in part by Interface Foundation of North America. Special thanks to Keming Song and Mary Slocum for providing data.

DNA fragments. Ideal genetic markers are probes which highlight exactly one RFLP, scoring F2s at this marker as MM, SS (parent types) or MS (hybrid, having both length fragments). However, RFLP patterns may be difficult to align (Branscomb, 1991) or distinguish (Figure 1).

Nearly adjacent genetic markers should largely agree in allele type across the F2s, with differences probably due to recombination between the markers. Distance is roughly proportional to the frequency of recombination: $1\% \approx 1$ centi-Morgan (cM) $\approx 10^5 - 10^6$ DNA base pairs (may vary along a chromosome and between species). Genetic linkage maps are currently constructed by examining pairs of markers, then triplets, then piecing together whole chromosomes (Lander and Green, 1987; Song et al. 1991).

Lander's MAPMAKER program provides a user-friendly environment for this empirical maximization of the joint likelihood of all marker loci across the genome. Interesting questions remain about further optimizing the search algorithm and ascertaining that it converges to a unique global maximum.

QTL, LOD and MLE

Genetic linkage maps are used to find quantitative traits loci, or QTL. The mean for a single locus quantitative trait y depends on the allele type, i.e.,

$$E(y) = \mu + ax + d(1 - |x|),\ V(y) = \sigma^2,$$

with $x = 1, -1, 0$ if the locus is of grandparent type MM, SS, or hybrid (MS), respectively. Here μ is the reference mean, a the additive allelic effect and d the dominance effect of allele type M. Under the null hypothesis of no QTL ($a = d = 0$), the F-statistic

$$F = [\sum(\bar{y} - \hat{y})^2/\nu_1]/\hat{\sigma}^2 \text{ is distributed as } F_{\nu_1,\nu_2},$$

with $\bar{y}$ the sample mean, $\hat{y}$ the least squares estimate, $\hat{\sigma}^2$ the variance estimate and degrees of freedom $\nu_1 = 2$ and $\nu_2 = n - 3$ for n F2s scored at this locus. For normal y, this is equivalent to the likelihood ratio statistic, typically presented in human genetics as

$$\begin{aligned} LOD &= \log_{10}(\textit{likelihood ratio}) \\ &= [0.5\sum(\bar{y} - \hat{y})^2/\hat{\sigma}^2]/\log(10) \\ &= \nu_1 F/2\log(10)\ . \end{aligned}$$

For normally distributed traits, these two approaches are equivalent and exact. Transformations toward normality are used in practice. Qualitative traits (counts and +/−) should use the "deviance" (McCullagh and Nelder, 1983) instead of the sum of squares. In some cases, this reduces to a χ^2 test on two-way frequency tables at each marker locus.

The "classical approach" computes the F-statistic at all markers, concluding that a QTL is near the marker locus with the most significant value. Lander and Botstein (1989) expanded the normal model to examine intervals between marker loci. Consider markers m and m' with recombinant frequency r and indicators x and x'. A QTL with ρr recombination with m has conditional expectation

$$E(y|r, m, m') = \mu + a[(1 - \rho)x + \rho x'] + df(x, x'; \rho, r),$$

where f is complicated but tractable (Knapp, Bridges and Birkes, 1990). However, conditional on position (ρ and r) the model is linear in parameters a and d.

Lander's MAPMAKER/QTL program profiles the likelihood (*cf.* Kalbfleisch and Sprott, 1970; Bates and Watts, 1988) across intervals for adjacent markers on the linkage map, with the maximum likelihood estimator (MLE) corresponding to the highest peak. At the MLE, if $a = d = 0$ then

$$\max(LOD) \approx [\nu_1/2\log(10)]F_{\nu_1,\nu_2} \approx \chi^2_{\nu_1}/2\log(10),$$

with the latter approximation used in practice, ignoring the extra variation of the estimate $\hat{\sigma}^2$.

Confidence Regions for QTL

Confidence regions arise by inverting the probability statement $Pr\{\max(LOD) \leq c_\alpha\} = 1 - \alpha$. The 99% theoretical confidence region for the major QTL of the phenotypic trait "days to budding" lies on chromosome 3 (Figure 2a), primarily around 100 cM but with small intervals around 60 and 80 cM. These intervals have LOD scores at least $\max(LOD) - 2$, with $c_{.01} \approx \chi^2_{2;.01}/2log(10) \approx 2$. Some QTL had confidence regions spanning intervals on several chromosomes. Beware that such regions may be too narrow, having much smaller coverage probabilty than expected (Terry Speed, pers. comm.).

The LOD score should be roughly quadratic near the true locus. In practice, the profile is quite irregular (Figure 2a) and the profile traces (Bates and Watts, 1988; Ritter, Bisgaard and Bates, 1991) for a and d exhibit strong nonlinearity and some numerical problems (Figure 3). This suggests caution in interpreting the parameter estimates from current methods, and a need for some refinement.

Major and Minor QTL

Finding multiple loci which control a quantitative trait is a stepwise process in which one identifies the major QTL, removes its effect, then proceeds to the most important minor QTL, and so on. For the two-loci additive model (ignoring interval mapping),

$$E(y) = \mu + a_1 x_1 + d_1(1 - |x_1|) + a_2 x_2 + d_2(2 - |x_2|),$$

where the major (1) and minor (2) loci may be on different chromosomes. The LOD can be decomposed, $LOD(1,2) = LOD(1) + LOD(2|1)$, suggesting that one fit the major locus model (as $\hat{y}_1$) and then conditionally fit the minor locus,

$$E(y - \hat{y}_1|x_1) = a_2 x_2 + d_2(2 - |x_2|) \ .$$

If the two loci were on separate chromosomes, one would expect estimates of a_2 and d_2 to be independent of x_1 and $LOD(2|1) = LOD(2)$. However, the profile likelihoods for possible minor loci for days to budding on chromosomes 6 and 7 changed substantially after removing a major QTL on chromosome 3 (Figures 2 and 4). Further, the MLE for the first minor QTL is at one end of chromosome 7, not in the middle of chromosome 6 as Figure 2 implies. These discrepancies may be due to epistasis (interaction), cosegregation of chromosomes during meiosis, or to a problem with modest sample size imbalance.

Discussion

Maximum likelihood interval mapping of QTL builds naturally on classical approaches. Important computational and theoretical issues remain in linkage map construction and finding QTL.

Several sources of variation arise in building linkage maps. "Riflotyping" of polymorphisms involves a visual assay of thousands of columns on blots, although these may soon be scanned by computer. Riflotype errors of RFLP patterns along linkage maps may affect estimates of map distance and marker loci order (Steve Knapp, Tom Osborn, pers. comm.).

The interval mapping approach to QTL assumes independence between marker intervals and that epistasis (interaction) between loci is negligible (Steve Knapp, Terry Speed, pers. comm.). Variation in estimated marker location on the linkage map may affect QTL peaks and parameter estimates (Figure 3). Further, the empirical distribution of LOD scores needs investigation under varied conditions.

As markers become more closely spaced, one wonders how information from neighboring regions could be effectively included in the estimation of QTL, particularly when there may be multiple loci. Present technology allows closely spaced markers (1–2 cM), increasing the problems of riflotyping. This raises both estimation and design questions: should one gather more F2s or more markers? How can one account for riflotype and other errors in the estimation procedure? Finally, how can one efficiently use information in the local neighborhood of QTL to smooth the likelihood surface by appropriate penalization?

References

Bates, D. M., and Watts, D. G. (1988) Graphical summaries of nonlinear inference regions. *Nonlinear Regression Analysis & Its Applications*, ch. 6. Wiley: New York.

Branscomb, E. (1991) Building physical genome maps by random clone overlap; a progress assessment of work on human chromosome 19. Proc. Workshop on Computational Molecular Biology (this conference).

Kalbfleisch, J. D., and Sprott, D. A. (1970) Application of likelihood methods to models involving large numbers of parameters. *J. Roy. Statist. Soc. B32*: 175-194.

Knapp, S. J., Bridges, Jr., W. C., and Birkes, D. (1990) Mapping quantitative trait loci using molecular marker linkage maps. *Theor. Appl. Genet. 79*: 583-592.

Lander, E. S., and Botstein, D. (1989) Mapping Mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics 121*: 185-199.

Lander, E. S., and Green, P. (1987) Construction of multilocus genetic linkage maps in humans. *Proc. Natl. Acad. Sci. USA 84*: 2363-2367.

McCullagh, P., and Nelder, J. A. (1987) *Generalized Linear Models*. Chapman and Hall: New York.

Ritter, C., Bisgaard, S., and Bates, D. M. (1991) A comparison of approaches to inference for nonlinear models. (this conference).

Song, K., Slocum, M. K., and Osborn, T. C. (1991) Use of RFLP markers for locating genes controlling morphological traits in *Brassica rapa*. *Ms. in prep.*

Song, K., Suzuki, J. Y., Slocum, M. K., Williams, P. H., and Osborn, T. C. (1991) A linkage map of *Brassica rapa* (syn. *campestris*) based on restriction fragment length polymorphism loci. *Theor. Appl. Genet.* (in press).

Figure 1. RFLP Southern Blots for 3 Probes with parents (S,M), F1, and some F2s. Note blurring in A and B, double RFLP in C. From Song et al. (1991).

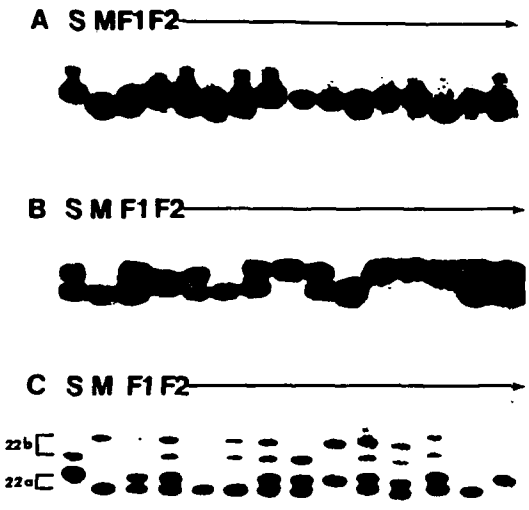


Figure 2. Profile Likelihood for Days to Budding

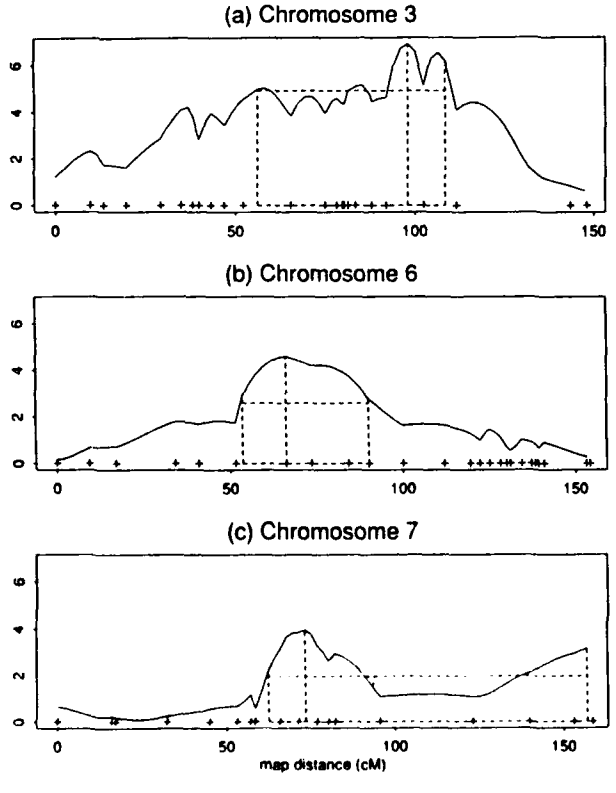


Figure 3. Profile Detail for Additive and Dominance Effects

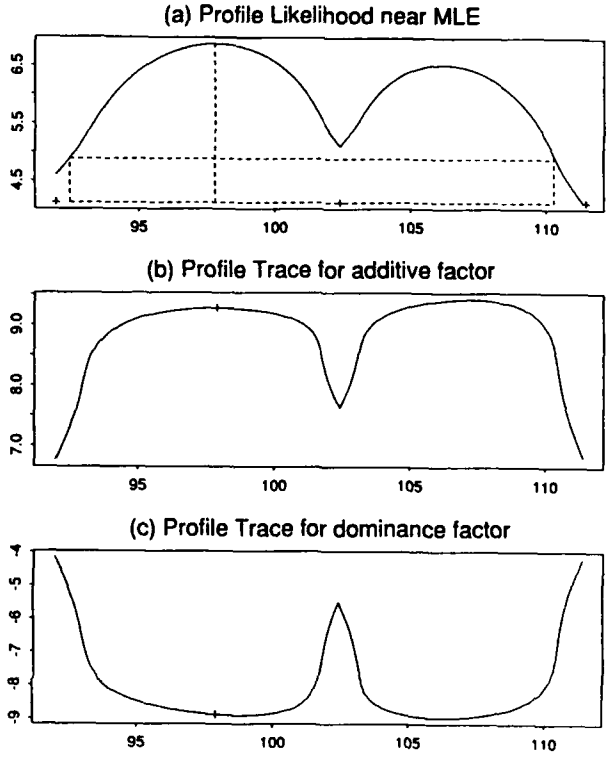
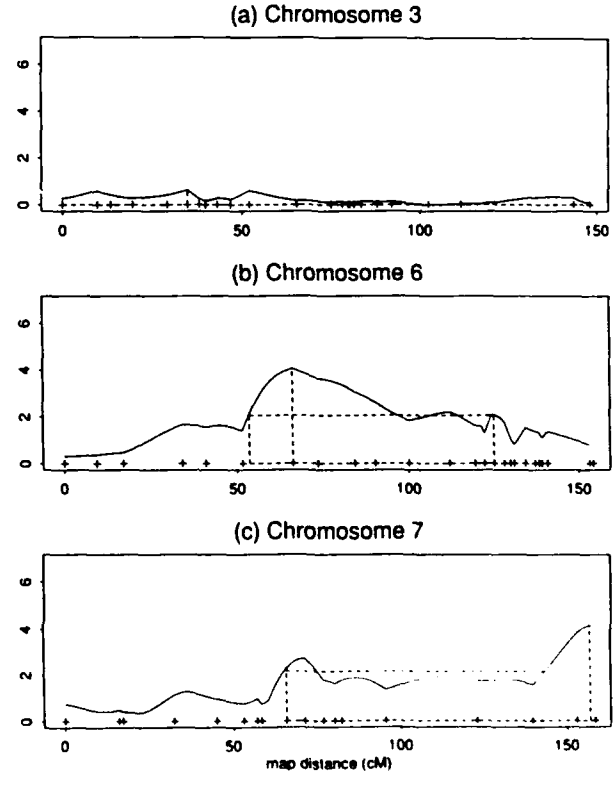


Figure 4. Conditional Profile Likelihood for Days to Budding



THE PARALLEL COMPUTATION OF PEDIGREE LIKELIHOODS

Nicholas J. Schork

Division of Hypertension
Department of Medicine
University of Michigan
Ann Arbor, Michigan, 48197

I. Introduction

The recent and almost monolithic surge in interest in molecular genetics and genetic analysis in general has been complimented to a great degree by recent advances in computer science. For instance, the analysis of protein structure and function — a vital component in assessing causal pathways between low-level genomic phenomena and phenotypic expression — has been greatly aided by contemporary visualization and high-speed data processing machinery. On another plane, the comparison and analysis of genome sequence data would be virtually impossible without supercomputers. Despite this apparent affinity between contemporary genetic analysis and computer science, there exist a number of areas in genetic research which have not yet tried to exploit specialized high speed computing machinery. One such area is the statistical analysis of linkage and segregation phenomena involving quantitative traits. This is odd given the fact that a great deal of theoretical or analytic work in these areas suggests and explicitly recommends the use of high speed or novel-design computers; see, for example, Elston and Stewart (1971), Lange and Elston (1975), Elston (1981), Boyle and Elston (1979), and Cannings, Thompson, and Skolnick (1978). In this paper, a parallel strategy for computing likelihoods on large complex pedigrees with quantitative phenotype data is discussed that makes use of a basic master/worker interconnect paradigm.

II. Evaluating Pedigree Likelihoods

Consider a locus with 2 alleles, A and a, that produces 3 genotypes, AA, Aa, and aa, occurring in the Hardy-Weinberg equilibrium dictated proportions $f(AA) = p^2$, $f(Aa) = 2p(1-p)$, and $f(aa) = (1-p)^2$, where p is the frequency of the A allele. Associated with each genotype is a mean effect μ_g, $g \in \{AA, Aa, aa\}$, and a common variance, σ^2. It should be understood that trait values are taken to be normally distributed around the relevant genotype mean.

Consider further a pedigree with N members of arbitrary complexity but without loops (i.e., inbreeding). For those pedigree members whose parents are not in the pedigree, the unconditional probability that they have genotype g is dictated by the frequency of the genotype $f(g)$. For those pedigree members, o, whose parents, m and f, are in the pedigree, the $f(g)$ parameters are replaced by *transmission probabilities*, $\tau(g_o|g_f, g_m)$, or the probabilities that an offspring, o, has genotype g_o given that his mother m and father f have genotypes g_m and g_f, respectively. Using this, the likelihood of the parameters $p, \mu_{AA}, \mu_{Aa}, \mu_{aa}, \sigma^2$ given data $X = (x_1, \ldots, x_n)$ collected on a pedigree with N members can be written as:

$$L(p, \mu_{AA}, \mu_{Aa}, \mu_{aa}, \sigma^2 | X) = \sum_{g_1} \sum_{g_2} \cdots \sum_{g_N} \delta(g_1)\phi(x_1|\mu_{g_1}, \sigma^2)\delta(g_2)\phi(x_2|\mu_{g_2}, \sigma^2) \cdots \delta(g_N)\phi(x_N|\mu_{g_N}, \sigma^2), \quad (1)$$

where the sums over the g_i, $i = 1, \ldots, N$, are sums over all possible genotypes member i might have, $\delta(\cdot)$ is either an $f(\cdot)$ function or an $\tau(o|m, f)$ function, depending on whether or not the pedigree member's parents are pedigree members, and $\phi(x|\mu, \sigma^2)$ is the normal density function. The compound sum in equation (1) over the g_i can be quite large and therefore prohibitive computationally (e.g., for 3 genotypes, the sum for a pedigree with N members would involve 3^N terms).

III. Parallel Likelihood Evaluation

The pioneering papers of Elston & Stewart (1971), Lange & Elston (1974), and Cannings, Thompson, & Skolnick (1978), all showed how the compound sum in equation (1) could be written as an iterated sum. The basic idea is to take small groups of closely related pedigree members (e.g., nuclear families or small pedigrees) and compute likelihoods involving these members conditionally on the genotypes of some "pivotal"

member of the group who is also the member of another group. These conditional likelihoods are saved and incorporated into the evaluations of likelihoods of other groups of pedigree members. As an example, consider the pedigree in figure 1a and the groupings for that pedigree depicted in figure 1b. Note that likelihoods involving group n_1 with members 1, 2, 4, 6, 8, 9, 11, 13 must be computed *after* likelihoods involving groups $n_1, \ldots, n_7$ since the genotypes of members 4, 6, 8, 9, 11, 13, are needed in the computations for groups $n_2, \ldots, n_7$. Consider calculations involving n_2. The likelihood involving members 4, 15, 16 can be computed conditionally on each possible genotype (AA, Aa, aa) for member 4. These three conditional likelihoods are saved. This same process is done for $n_3, \ldots, n_6$ by conditioning on members $6, 8, 9, 11, 13$, respectively. The resulting condtional likelihoods are then used to "weight" the possible genotype arrangements considered in the likelihood evaluation of n_1 members 4, 6, 8, 9, 11, 13 in conjunction with members 1 and 2.

For pedigrees which have a single line of descent emanating from each spouse-pair (e.g., only one spouse has his/her parents in the pedigree, as in figure 2), an implemenation of the Elston-Stewart algorithm on a parallel computer can be described as follows. Starting with the youngest (or latest generation, G), nuclear families compute the conditional likelihoods of the members of these nuclear families conditioning on those parents who are offspring in the $G-1$ generation of nuclear families. If there are n_ℓ nuclear families in the ℓth generation, then these conditional likelihoods can be computed on ℓ processors simultaneously. Once conditional likelihoods for a generation's nuclear families have been computed, they are saved and sent out for processing with the next oldest generation's nuclear families. This process continues until the final "root" nuclear family's likelihood is computed. The running time of this strategy would have the simple form:

$$s(n_1) \div \sum_{\ell=2}^{G} max[s(n_{\ell,i}); i = 1, \ldots, n_\ell], \qquad (2)$$

where $s(n_i)$ is the *size* of nuclear family n_i, G is the number of nuclear family generations, and n_ℓ is the number of nuclear families at generation ℓ. It can be seen that the largest nuclear family at generation ℓ dominates the computation time spent computing the conditional likelihoods associated with nuclear families at that generation, and that a single processor time is needed to compute the likelihood involving the final "root" nuclear family (n_1). Note that since there can be no genotype elimination based on phenotype information for quantitative trait analysis, running time scales with the size, not phenotype arrangement, of a nuclear family.

For more complex pedigrees a different strategy is needed. Consider the pedigree as a connected graph, where each nuclear family in the pedigree is a node. The edge relationships between the nodes are dictated by the relatedness of the members of the nuclear families comprising the pedigree. Each node is assigned a weight equal to the number of members in the nuclear family it represents. Figure 3b depicts the graph-theoretic representation of the pedigree displayed in figure 3a. Figure 4 displays a more complicated pedigree's graph respresentation. To optimally compute a likelihood involving a complex pedigree in parallel, compute, for each node, the sum of the weights of all those nodes in an edge relationship with nodes themselves in an edge relationship that ultimately leads to the node in question. Call the nodes entering into each of these sums a "path". Let H_t be the "depth" of (i.e., the number of nodes implicated in) each path, t. The node with the smallest maximum path (i.e., sum of weights) should be taken as representing the "root" nuclear family whose likelihood calculations are computed last. The nuclear families furthest away in node representation from the root for each path are distributed to different processors for conditional likelihood evaluation. Once the conditioning processes are completed for a nuclear family, the likelihoods are sent to a processor which will compute likelihoods of a nuclear family whose node representation is in an edge relationship with the node representation of the nuclear family in question. Note that some conditional likelihoods will be sent to a common processor (i.e., two nodes have an edge relationship with a common node). In this way, the conditioning processes will converge to the root node, and thus give the complete likelihood of the pedigree. See Schork (1991) for more details, an assessment of the running time, and some experimental results.

IV. Discussion

The algorithm to compute pedigree likelihoods outlined above is intuitive, but does possess some minor problems. First, it is imperative that an efficient way of determining the optimal order in which to compute the nuclear families in the pedigree be used or

Amdahl's law will render the computational savings in computing the conditional likelihoods in parallel useless. Second, not all pedigrees will work well with the algorithm; for instance, a pedigree which is simply a horizontal chain of nuclear families can have no more than two of its constituent nuclear families' conditional likelihoods computed simultaneously. Third, it may be the case that over the course of the computations through the various "paths" some processors will not be utilized, resulting in an inefficient use of the computer. On the other hand, the algorithm can be improved by letting a number of processors work on parts of the sums needed in the computation of a given nuclear family's likelihood calculations. In this way, processors would be utilized to a greater degree and a faster turn around time would result also.

References

Boyle CR, Elston RC (1979) Multifactorial genetic models for quantitative traits in humans *Biometrics* 35:55-68.

Cannings C, Thompson EA, Skolnick MH (1978) Probability functions on complex pedigrees, *Advances in Applied Probability* 10:26-61.

Lange K, Elston RC (1975) Extensions to pedigree analysis I. Likelihood calculations for simple and complex pedigrees *Hum Hered* 25:95-105.

Elston RC (1981) Segregation analysis, in Harris H, Hirshhorn K (eds) *Advances in Human Genetics* New York: Plenum, 63-120.

Elston RC, Stewart J (1971) A general model for the genetic analysis of pedigree data *Hum Hered* 21:323-342.

Schork NJ (1991) A parallel "clipping" algorithm for pedigree likelihood evaluation involving quantitative phenotype data, *submitted*.

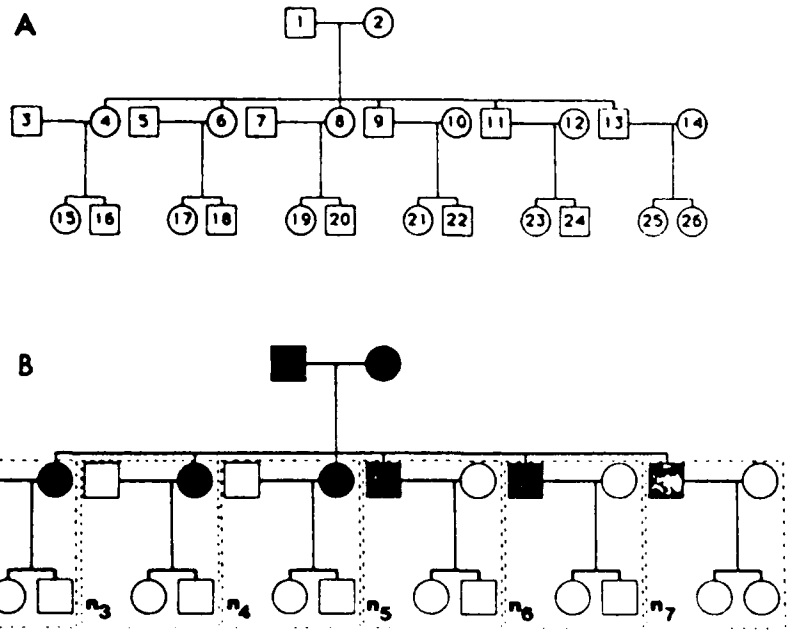


Figure 1

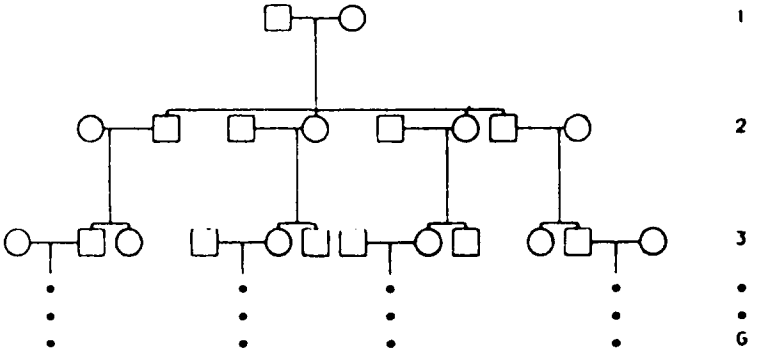


Figure 2

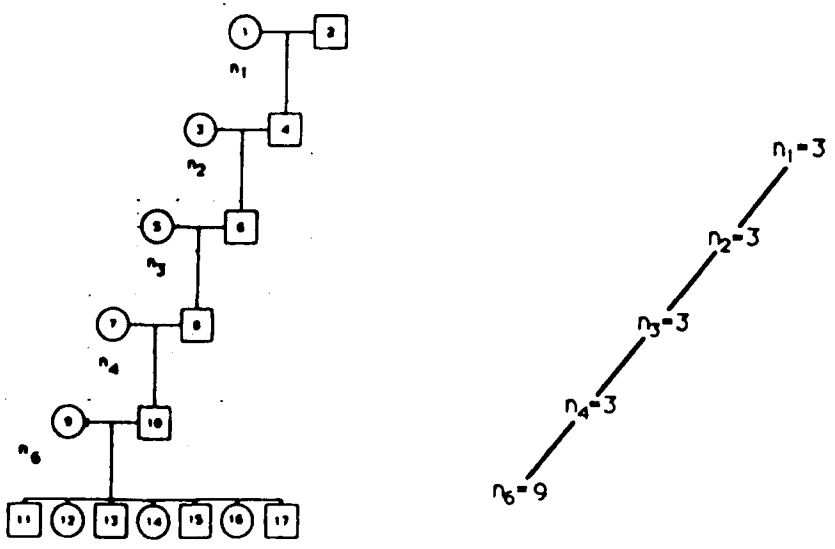


Figure 3

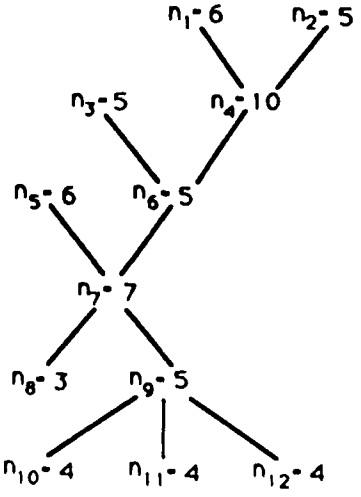


Figure 4

Tuning Complex Computer Code to Data

Dennis Cox[1,2] **Jeong Soo Park**[1,2] **Jerome Sacks**[1]
Department of Statistics

Clifford Singer[1]
Department of Nuclear Engineering

University of Illinois
Champaign, IL 61820

Abstract

The problem of estimating parameters in a complex computer simulator of a nuclear fusion reactor from an experimental database is treated. Practical limitations do not permit a standard statistical analysis using nonlinear regression methodology. The assumption that the function giving the true theoretical predictions is a realization of a Gaussian stochastic process provides a statistical method for combining information from relatively few computer runs with information from the experimental database and making inferences on the parameters.

1 Introduction and Problem Formulation

Mathematical models of natural phenomena are often implemented in complex computer programs. Sometimes, the mathematical model is completely specified, and it is only necessary to execute the code to make predictions about the natural process under study. However, there may be unknown parameters in the mathematical model. If there is an available database of experimental results, then statistical methods may be employed to make inferences about the unknown parameters and predictions of the natural process. In this article, we consider an example of such a problem in nuclear fusion research. In this application, the computer implementation of the mathematical model is so complex that it is not practically possible to execute the program as many times as is needed to perform a classical statistical analysis. We propose a Bayesian methodology which allows us to combine limited information on the mathematical model (obtained from relatively few computer runs) with the experimental database and still make the requisite inferences. The methodology we propose may prove useful in other applications in other disciplines where complex computer codes must be tuned to real data sets.

A *tokamak* (a Russian acronym for "toroidal magnetic chamber") is a device for producing plasmas capable of nuclear fusion (Wesson, 1987). There are currently 45 tokamaks worldwide and a large database of individual "shots" (i.e., experimental runs) potentially available. Currently, none of the tokamaks is capable of producing more energy from fusion reactions than the amount of energy neded for confinement and heating of the plasma. It is desired that the next generation of tokamaks should reach at least the break-even point. As such devices are extremely expensive (approximately $\$10^{10}$), it is critical that a good model is available for design purposes.

There are several variables which the experimenters can control from shot to shot, of which the main ones are: I = Plasma current; P = Heating power; n = particle density; B = toroidal magnetic field. There are also some geometrical variables, but for simplicity we will limit ourselves to two tokamaks, ASDEX (in Germany) and PDX (in Princeton), which have fixed simple geometries, so that the only independent variables are given above. We will use the first four components of the five dimensional vector $\mathbf{x}$ to denote the logarithms of the independent variables, and the fifth component will be an indicator of the machine, say $x_5 = 0$ for PDX and $x_5 = 1$ for ASDEX.

A primary factor in promoting fusion is energy confinement, which is measured by the *global energy confinement time* τ_E. It is a measure of the rate at which energy is escaping from the tokamak at steady state. We will let y denote the measured value of $\log \tau_E$. We have 32 observations from ASDEX and 42 from PDX, giving a total of 74 observations $(\mathbf{x}_i, y_i)$.

A comprehensive mathematical model for tokamaks has been developed and implemented in a computer code

[1] Research supported by U.S. Department of Energy contract DE-FG02-88ER53269.

[2] Research supported by NSF Grant DMS 90-01726, and NSA Grant MDA 904-89-H-2011. Support for computing was provided by Cray Research and National Center for Supercomputing Applications at the University of Illinois, Urbana-Champaign.

known as Baldur (Singer, et. al., 1988). There are several unknown parameters in the Baldur model, however, which we generically denote by the vector $\mathbf{c}$. Each Baldur run requires inputting a value of $\mathbf{x}$, the experimentally observable independent variables and machine indicator, and a value of $\mathbf{c}$. We let exact theoretical prediction (which is the ideal output of Baldur) be denoted $Y(\mathbf{x}, \mathbf{c})$.

In principle, inferences about $\mathbf{c}$ can be made with nonlinear regression techniques. Thus, $\mathbf{c}$ may be estimated by minimization of the residual sum of squares

$$\mathrm{RSS}(\mathbf{c}) = \sum_{i=1}^{74} [y_i - Y(\mathbf{x}_i, \mathbf{c})]^2$$

where y_i is an observed $\log \tau_E$ and $Y(\mathbf{x}_i, \mathbf{c})$ is the corresponding theoretical prediction. Since $\mathbf{c}$ is four dimensional for our case, maybe about 100 evaluations of RSS($\mathbf{c}$) would be needed for a nonliner optimizer to find $\mathbf{c}$. Even if the nonlinear least squares estimate $\hat{\mathbf{c}}$ and RSS($\hat{\mathbf{c}}$) are known, it requires a minimum of 10 evaluations to estimate the Hessian for contructing confidence regions.

There are two important features of the Baldur code which make this classical nonlinear regression approach infeasable. For one thing, each Baldur run takes about 4 minutes of CPU time on a Cray II supercomputer. Thus, even a single evaluation of RSS($\mathbf{c}$) requires about 5 hours of supercomputer time, and 10 to 100 evaluations of RSS($\mathbf{c}$) are simply not practical. Secondly, Baldur does not output the exact value of the prediction $Y(\mathbf{x}, \mathbf{c})$, but has a sampling error because of a Monte Carlo integration inside of one routine. Multiple runs of Baldur with the same inputs $(\mathbf{x}, \mathbf{c})$ but different seeds would be required to obtain an accurate value of $Y(\mathbf{x}, \mathbf{c})$.

We are thus constrained to making a limited number of Baldur runs in order to obtain noisy values of $Y(\mathbf{x}, \mathbf{c})$, and then to somehow combine this incomplete and inexact computer data with the experimental data in order to make inferences on the parameter vector $\mathbf{c}$.

2 Statistical Model

We propose a statistical model for the problem described above. The true function $Y(\mathbf{x}, \mathbf{c})$ is assumed to be a realization of a stochastic process. Such models have been successfully used for design and analysis of computer experiments (Sacks, Schiller, and Welch, 1989, abbreviated [SSW]; Sacks, Welch, Mitchell, and Wynn, 1989, abbreviated [SWMW]). Further details on our proposed methodology are given in Park (1991, abbreviated [P]).

We will treat the models for the two tokamaks ASDEX and PDX entirely independently. Thus, for most of this section, the variable $\mathbf{x}$ does not include an indicator for the machine and we will assume only one machine. Having obtained a likelihood for one machine, the combined likelihood for both machines is obtained my multiplication of their individual likelihoods (addition of log likelihoods). We are not convinced this is the best approach, but it is easy and probably leads to valid if not fully efficient results.

For convenience, we will let $\mathbf{s}$ and $\mathbf{t}$ denote a value of the vector $(\mathbf{x}, \mathbf{c})$, where $\mathbf{x}$ and $\mathbf{c}$ are both 4-dimensional, so $\mathbf{s}$ and $\mathbf{t}$ are 8-dimensional. We will use d to denote the general dimensions of $\mathbf{s}$ and $\mathbf{t}$.

It is assumed that $Y(\mathbf{s})$ is a Gaussian stochastic with constant mean

$$EY(\mathbf{s}) = \beta,$$

and covariance function

$$\mathrm{Cov}[Y(\mathbf{s}), Y(\mathbf{t})] = \sigma_Y^2 \exp\left[-\sum_{i=1}^{d} \theta_i (s_i - t_i)^2\right].$$

Here, $\beta, \sigma_Y^2 > 0$, and $\theta_i > 0$, $1 \leq i \leq d$, are parameters. More general models are possible (see [SSW], [SWMW], or [P]), but some data analysis and model fitting has suggested a model of this form is appropriate.

The Baldur code is executed at a set of inputs $\mathbf{s}_i$, $1 \leq i \leq n_C$, giving observations y_{iC}, $1 \leq i \leq n_C$, which are modelled as

$$y_{iC} = Y(\mathbf{s}_i) + \varepsilon_{iC}$$

where the random errors ε_{iC} are assumed to be i.i.d. $N(0, \sigma_C^2)$, and independent of Y. Note that the subscript "C" designates computer data.

The experimental data $(\mathbf{x}_i, y_{iE})$, $1 \leq i \leq n_E$, is modelled as

$$y_{iE} = Y(\mathbf{x}_i, \mathbf{c}_0) + \varepsilon_{iE}$$

where the ε_{iE} are i.i.d. $N(0, \sigma_E^2)$, independent of all previously mentioned random quantities. Also, $\mathbf{c}_0$ denoted the true unknown value of the fusion theory parameters to be estimated.

It is convenient to reparameterize the variances of the random errors in terms of variance ratios, viz.

$$\gamma_C^2 = \sigma_C^2/\sigma_Y^2, \qquad \gamma_E^2 = \sigma_E^2/\sigma_Y^2.$$

Thus, in addition to the 4-dimensional theory parameter $\mathbf{c}_0$, we also need to estimate β, θ (8-dimensional), γ_C^2, γ_E^2, and σ_Y^2. Further, we assume each of these parameters (other than $\mathbf{c}_0$) is different for the tokamaks PDX and ASDEX. The Gaussian process assumptions allow us to develop formulae for the multivariate normal likelihoods,

which are maximized by a numerical optimization program.

One perspective on the above is that we are fitting a nonparametric regression function $Y(\mathbf{x}, \mathbf{c})$ using an empirical Bayesian methodology. Part of the data (the experimental observations) are missing components of the independent variable, and the missing components have a common value $\mathbf{c}_0$. The parameters $\boldsymbol{\theta}$, γ_C^2, and γ_E^2 control the "smoothness" of the fit, analogously to the smoothing parameters m and λ in the Bayesian interpretation of smoothing splines (Eubank, 1988, pp. 233–248).

Among the several strategies we have tried for estimating parameters, the one which has worked best in simulations (reported in [P]) is the following: estimate the parameters $\boldsymbol{\theta}$ and γ_C^2 by maximum likelihood using the computer data alone. Then combine the computer data and experimental data to estimate the remaining parameters: β, σ_C^2, σ_E^2, and $\mathbf{c}_0$. One reason this method may work well is that it uncouples some of the smoothing parameter estimation from the estimation of $\mathbf{c}_0$. It has the added advantage that it reduces computational time.

There was strong prior knowledge about the theory parameter vector $\mathbf{c}$ which was codified into a log-normal prior distribution. The components of $\mathbf{c}$ were *a priori* independent with the following normal distributions

$$\log c_1 \sim N(0, (2\log 2)^2),$$
$$\log c_2 \sim N(\log 3, (\log 2)^2),$$
$$\log c_3 \sim N(0, (\log 2)^2),$$
$$\log c_4 \sim N(\tfrac{1}{2}\log 2, (\tfrac{1}{2}\log 2)^2).$$

The corresponding quadratic terms were added in to the log likelihood to penalize values of $\mathbf{c}$ for being far from the prior mean. The maximization of this penalized log likelihood amounts to finding the posterior mode.

One strategy which we found useful for parameter parsimony was to constrain components of $\boldsymbol{\theta}$ to be equal. Based on initial estimates wherein all components of $\boldsymbol{\theta}$ were varied independently, we chose three blocks of the components of $\boldsymbol{\theta}$ which were constrained to have a common value, and then computed maximum likelihood estimates under these constraints.

To assess the accuracy of our estimates of $\mathbf{c}_0$, estimated standard errors are computed using the diagonal entries of the inverse of the Hessian of the posterior log likelihood evaluated at the maximum. This Hessian was evaluated numerically. While we are aware of no directly relevant asymptotic (or finite sample) theory to justify this, the simulations reported in [P] suggest that it does not work badly, although we have far too few simulations to assess coverage probabilities. In any event, it does provide a reasonable indication of error.

3 Numerical Results

In this section we report the results of applying the methodology of the previous section to the Baldur/Tokamak problem. More details, including the data sets, may be found in [P].

Table 1 shows the smoothing parameter estimates obtained from only the computer data. To recapitulate, for each computer data set (e.g., $n_C = 34$ observations for PDX), we can maximize the likelihood based in the vector of observations $y_C = (y_{iC} : 1 \le i \le n_C)$ to obtain estimates of β, σ_Y^2, γ_C^2, and $\boldsymbol{\theta}$. However, only the estimates of γ_C^2 and $\boldsymbol{\theta}$ are used in the subsequent analysis. One will note that we constrained $\theta_1 = \theta_8$, $\theta_2 = \theta_3 = \theta_7$, and $\theta_4 = \theta_5 = \theta_6$. This decision was made to parsimoniously parametrize after initially estimating 8 independent θ_i's for each simulated machine.

Tables 2 and 3 present the results of the subsequent likelihood maximization when computer and experimental data from both machines were pooled. Table 2 shows parameter estimates which are individual to a given machine. The values in Table 3 are the ones of most interest — the estimates of the theory parameters. The estimate of c_2 has a relatively large estimated standard error, indicating that our knowledge of it at this time is rather uncertain.

Finally, in Figures 1 and 2 we show residual plots (residuals vs. predicted values). Based on our simulation experience with toy models reported in [P], these plots suggest a relatively good fit. In particular, the predicted values for the computer data (dots) have a wider (horizontal) range than the experimental predicted values. Also, the residuals for the computer data have a much smaller (vertical) range then those of the experimental data. These indicate we are fitting the computer data relatively well, and also getting good coverage of the range of $Y(\mathbf{x}, \mathbf{c})$.

Concluding Remarks

There are a number of issues which arose in this investigation which we have not mentioned for lack of space. One is the design of the computer experiment. One problem which arose during the collection of computer data is that the Baldur code was modified to improve convergence. There was some change in output values between the new and old codes when the same inputs were tried, but it was on the same order as our estimate at the time of the Monte Carlo sampling error. However, subsequent analysis suggested the Monte Carlo error was much smaller than we thought, and that the two versions of the code produced somewhat different answers. All of our results above are based on the

data from the new and improved program. This does raise the question of the size of the approximation and roundoff error in the code and the extent to which that affects the parameter estimates.

This is a complex problem and there is much opportunity for future research.

References

Eubank, R. L. (1988), *Spline Smoothing and Nonparametric Regression*, Marcel Dekker, New York.

[P] Park, J. S. (1991), Ph.D. Thesis in preparation, Department of Statistics, University of Illinois, Champaign, IL.

[SSW] Sacks, J., Schiller, S. B., and Welch, W. J. (1989), Designs for computer experiments, *Technometrics* **31**, 41–47.

[SWMW] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), Design and analysis of computer experiments (with discussion), *Statistical Science* **4**, 409–435.

Singer, C. E., Post, D. E., Mikkelsen, D. R., Redi, M. H., McKenney, A., Silverman, A., Seidl, F. G. P., Rutherford, P. H., Hawryluk, R. J., Langer, W. D., Foote, L., Heifetz, D. B., Houlberg, W. A., Hughes, M. H., Jensen, R. V., Lister, G., and Ogden, J. (1988), Baldur: a one-dimensional plasma transport code, *Computer Physics Communications* **49**, 275–398.

Wesson, J. (1987), *Tokamaks*, Clarendon Press, Oxford, United Kingdom.

Table 1. Smoothing Parameter Estimates from Computer Code

Symbol	Description	PDX Value	ASDEX Value
n_c	Computer data set	34	31
θ_1	Correlation Coefficient for c_1	1.6	.033
θ_2	Correlation Coefficient for c_2	.19	.13
θ_3	Correlation Coefficient for c_3	.19	.13
θ_4	Correlation Coefficient for c_4	1.1	.35
θ_5	Correlation Coefficient for $\log I$	1.1	.35
θ_6	Correlation Coefficient for $\log B$	1.1	.35
θ_7	Correlation Coefficient for $\log n_c$	.19	.13
θ_8	Correlation Coefficient for $\log P$	1.6	.033
γ_C^2	Variance ratio σ_C^2/σ_Y^2	4.9×10^{-4}	1.1×10^{-3}

Table 2. Parameter Estimates for Individual Tokamaks

Symbol	Description	PDX Value	ASDEX Value
n_E	Experimental data set sample size	42	32
β	Mean value for Y	-1.68	-1.59
σ_Y^2	Variance of Y	.0045	.35
γ_E^2	Variance ratio σ_E^2/σ_Y^2	.023	.0064

Table 3. Estimates for Theory Parameters

Symbol	Description	PDX Value	ASDEX Value
c_1	Drift Waves Coefficient	1.65	.15
c_2	Rippling Coefficient	2.08	.65
c_3	Resistive Ballooning Coefficient	1.14	.22
c_4	Critical Value for the Ion Temperature Gradient Mode	1.16	.095

Figure 1 : Residual Plot for ASDEX

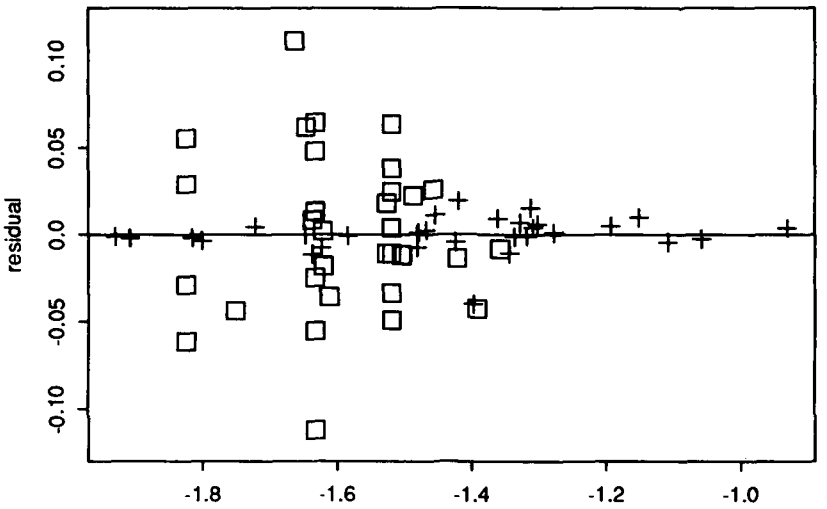
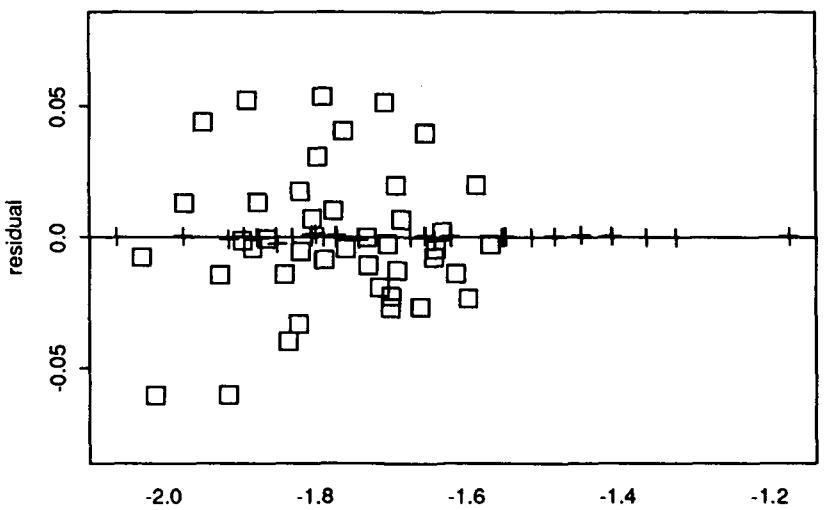


Figure 2 : Residual Plot for PDX



Using Computer Experiments to Construct a Cheap Substitute for an Expensive Simulation Model

Toby Mitchell and Max Morris*
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6367

Abstract

There is widespread use of computer models as tools in scientific research. As surrogates for physical or behavioral systems, such models can be subjected to experimentation, the goal being to predict how the corresponding real system would behave under certain conditions. For long-running (expensive) model codes, there may be a severe limitation on the number of experiments that can reasonably be done. This motivates the construction of a fast-running (cheap) approximation to the original code, for use in experiments where a large number of runs may be necessary. Here we discuss our approximation of a simulation model for the compression molding of sheet molding compound, applied to the manufacture of an automobile hood. The approximation was constructed using Bayesian interpolation methods for prediction of the movement of the flow front. The predictions were based on data generated by a sequence of computer experiments, using designs chosen according to a type of D-optimality criterion.

1 Introduction

The purpose of this paper is to demonstrate the application of Bayesian methods for design and analysis of computer experiments to the construction of a "cheap" substitute for an "expensive" computer model. As our example, we shall use a computer simulation model for a compression mold-filling process that is used in the manufacture of automobile hoods. Our primary use of this model was to generate prediction formulas that could serve as fast substitutes for the real model in certain well-defined tasks. This done, we did not follow through any further, so this account is best considered as a realistic example rather than a complete scientific application.

*Research sponsored by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy Contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

Except for some philosophical differences, our underlying approach is essentially the same as that discussed by Sacks, Welch, Mitchell, and Wynn (1989). As noted there, versions of this approach have been used for a long time in various settings, e.g., kriging and Bayesian interpolation. The details of the method (e.g., choice of correlation function, design criterion) are more in line with Currin, Mitchell, Morris, and Ylvisaker (1991).

2 The Computer Model

Sheet molding compound (SMC) is composed of polymer resin, chopped fibers, filler, and additives. Prior to the molding process, a "charge", or piece of SMC, is cut from a sheet and placed in a heated mold. The process is begun by closing the mold slowly; during the process the material flows and fills the mold cavity. After filling, a constant force is maintained on the mold, as the curing reaction proceeds; then the part is removed and the curing is completed.

Designers of the manufacturing process are concerned with the movement of the flow front; it is desirable that the charge fill the mold evenly and rapidly, without the presence of "knit lines" formed when two parts of the flow front meet. To help determine the effect of the design parameters (e.g., the initial shape and placement of the charge) on the flow front movement, a computer simulation model is used. This model is a version of the TIMS (Thin Mold filling Simulation) model, which was developed by Tim Osswald and Charles Tucker of the Department of Mechanical Engineering at the University of Illinois. The version we used came to us through the courtesy of Alonzo Church, Jr. and Daniel Fleming of GenCorp Research, who were of great help to us in learning to use it and in evaluating the results. The theory and numerical implementation are described in Osswald and Tucker (1990). The inputs to the code include the geometry of the part, the material properties (e.g., viscosity), the closing speed, the final thickness of the part, and

the shape and location of the charge. The output consists of all the information needed to predict the position of the flow front as a function of time. The code uses a finite element method to solve a system of differential equations based on the physics of the process. This is not a trivial computation -- each run of the model code takes 4-5 minutes on a Cray X-MP computer. For specific, well-defined experiments, it is worthwhile, therefore, to seek a fast approximation to the model; this is the purpose of the exercise we shall describe here. Of special interest to us is the highly multidimensional nature of the response (flow front movement). Previous applications of our prediction method, and of similar methods described by other authors, have been concerned with prediction of a single response computed from the output. Although we shall do nothing more than apply the same prediction method separately to 2345 related responses, we shall see that even this kind of naive approach can be useful.

3 Predictors and Responses

In this example, we are concerned only with the effect of the initial shape and location of the charge. The input that defines this is a list of "nodes" (in the finite element discretization of the mold surface) that are filled initially by the charge. There are 469 nodes altogether, and the initial charge typically fills 30 to 40 of them. (Although nodes are actually points, each is associated with a small subvolume of the mold. When we refer to a node as being "filled", we are really referring to this associated subvolume.) In order to represent the list of initially filled nodes by a few predictor variables, we require the initial shape of the charge to be rectangular. The predictor variables are then defined by the boundaries of the rectangle. This is done conveniently using the node map as constructed for the finite element method, where the nodes form an approximately uniform grid over the part of the mold where the charge might be placed. The north and south boundaries of the charge correspond to the predictor variables t_1 and t_2, while the east and west boundaries correspond to t_3 and t_4, respectively. (The scaling is such that $0 < t_2 < t_1 < 1$ and $0 < t_4 < t_3 < 1$.) For other geometries, of both the charge and the region of the mold into which the charge is to be placed, the representation of the initial shape and location of the charge by a few predictor variables might be considerably more difficult.

The next part of the setup of the prediction problem is to define, from the mass of output, a manageable set of response variables that will permit prediction of the flow front. The output gives values of the function p_m for all nodes $m = 1,..., 469$ at each time step in the simulation, where $p_m(\tau)$ denotes the proportion of node m that is filled at time τ.

At each node m, we defined the five responses

y_{m1}: the last recorded time at which node m is empty ($p_m(y_{m1}) = 0$),

y_{m2}: the time at which node m becomes 25% full ($p_m(y_{m2}) = 0.25$),

y_{m3}: the time at which node m becomes 50% full ($p_m(y_{m3}) = 0.50$),

y_{m4}: the time at which node m becomes 75% full ($p_m(y_{m4}) = 0.75$),

y_{m5}: the first recorded time at which node m is 100% full ($p_m(y_{m5}) = 1$).

Since these values are not given directly by the output, which gives values of p_m at various times, we approximated them by linear interpolation of the output data. The prediction problem was then taken to be: Approximate the 2345 functions $y_{mr} = y_{mr}(t_1, t_2, t_3, t_4)$, where $m = 1, ..., 469$ and $r = 1, ..., 5$, over the region defined by $0 < t_2 < t_1 < 1$, $0 < t_4 < t_3 < 1$. Two further practical constraints on the region of interest were added. The first restricted the placement of the charge to be symmetric about the north-south center line, i.e., $t_3 + t_4 = 1.0$. The second required that the number of the nodes initially filled by the charge be between 30 and 40; this was our way of implementing a requirement that the area of the mold surface initially covered by the charge be fairly constant.

4 Design

The central idea (which is not original with us) is to represent uncertainty about each function y_{mr} on the k-dimensional region of interest T by means of a stochastic process (random field) Y_{mr}. For simplicity and convenience, we use stationary Gaussian (normal) processes as priors. These are fully described by a constant $\mu_{mr} = E[Y_{mr}(t)]$, a constant $\sigma^2_{mr} = V[Y_{mr}(t)]$, and a correlation function R_{mr}, where $R_{mr}(d) = \text{Corr}[Y_{mr}(t+d), Y_{mr}(t)]$ and where $t = (t_1, \cdots, t_k)$ and $t+d = (t_1+d_1, \cdots, t_k+d_k)$ are any two "sites" (points in T) separated by a difference vector d. For simplicity, we also take the 2345 Y_{mr} processes independent of one other, and $R_{mr}(d) = R(d)$ for all (m,r). (The choice of independence is made at the cost of ignoring information about the relationships among the y_{mr}'s at any site. We have not found it feasible to

implement such information here.)

For a design criterion, we use the "maximum entropy" principle (Lindley 1956), which in this case leads to a kind of D-optimality, namely, the maximization of $|C_{DD}|$, where C_{DD} is, for any one of the processes Y_{mr}, the n×n matrix of *prior* correlations among the design sites (Shewry and Wynn 1987). We find this criterion appealing, for reasons given by Currin et al. (1991), but other criteria could be used. (See, e.g., Sacks, Schiller and Welch 1989 and Sacks, Welch, Mitchell, and Wynn 1989.)

Of course, one cannot maximize $|C_{DD}|$ without specifying how C_{DD} depends on D. For our priors, this means specifying the correlation function R. We favor using a weak correlation function, i.e., one for which R(d) decreases rapidly to zero as d increases. Such a strong conviction of prior ignorance is not useful for analysis, since one would need to observe y at very many sites, located densely in T, in order to yield predictions that are usefully precise. At the design stage, however, we feel that the choice of a weak correlation function is appropriately conservative.

For design purposes then, we use the exponential correlation:

$$R(d) = e^{-\theta \sum_j |d_j|} \qquad (4.1)$$

where θ is "large". Asymptotically (as $\theta \to \infty$), it can be shown that the D-optimality criterion, where (4.1) is used to construct C_{DD}, maximizes the minimum intersite distance $\sum |d_j|$ among design points, and favors those designs with the fewest pairs whose intersite distance matches this minimum. This is a special case of a result due to Johnson, Moore, and Ylvisaker (1990), who called such designs "maximin distance" designs. In this sense, the designs we construct will attempt to push the design points as far away from each other as possible.

For design construction, we use an algorithm similar to DETMAX (Mitchell 1974). Starting with a random set of n sites, the algorithm does a series of "excursions" in which candidate sites are added to and removed from the design. When adding a site, the chosen site is intended to be the one at which the posterior variance, based on the current design, is largest. It may not be possible to ensure this if there are many sites to consider; if this is the case, the algorithm does a limited search. When removing a site, the chosen site is the one corresponding to the largest diagonal element in the inverse of the current C_{DD} matrix. See Currin, et al. (1991) for further details.

Here the set of candidate runs was formed by first letting t_1 and t_2 take any of 11 levels and t_3 and t_4 take any of 13 levels, subject to the restrictions on the region of interest noted above.

The initial 10-run design, plus an additional 5 runs that were chosen later, are shown in Table 1.

Initial Design

Run	t_1	t_2	t_3	t_4
1	0.40	0.00	0.75	0.25
2	0.40	0.20	1.00	0.00
3	0.80	0.60	1.00	0.00
4	1.00	0.00	0.58	0.42
5	0.80	0.40	0.75	0.25
6	0.60	0.40	0.92	0.08
7	0.50	0.20	0.83	0.17
8	0.70	0.10	0.67	0.33
9	0.90	0.60	0.83	0.17
10	1.00	0.50	0.67	0.30

Additional Points

Run	t_1	t_2	t_3	t_4
11	0.50	0.00	0.67	0.33
12	0.70	0.40	0.83	0.17
13	1.00	0.60	0.75	0.25
14	0.60	0.20	0.75	0.25
15	0.90	0.20	0.67	0.33

Table 1. Design for experiment on compression molding model.

The need for the additional runs was clear after inspection of the cross-validation predictions based on the initial experiment. These runs were chosen using the same algorithm and the same correlation function which generated the first ten runs. The full 15-run design populates the region of interest (which is relatively small here) quite densely; the maximum distance $\sum_{j=1}^{4} |t_j - s_j|$ between any feasible site t not in the design and the closest design site s is 0.2.

5 Prediction

Predictions were made using standard formulas for conditional normal distributions. Let $y_{mr,D}$ be the vector of the n observed values of y_{mr}. The mean of $Y_{mr}(t)$ given $Y_{mr,D} = y_{mr,D}$ is:

$$\hat{y}_{mr}(t) = \mu_{mr} + C_{tD}C_{DD}^{-1}(y_{mr,D} - \mu_{mr}J_n) \qquad (5.1)$$

where C_{tD} is a row vector that holds the n prior correlations between $Y_{mr}(t)$ and $Y_{mr,D}$ and J_n is the column vector composed of n 1's. In order to use (5.1), one needs to specify the prior mean μ_{mr} and the correlation function (needed for C_{tD} and C_{DD}). In our approach, we arbitrarily chose a family of correlation functions, indexed by a set of parameters θ, and then used cross-validation to select μ_{mr} and θ.

For the present example, we chose the product piecewise cubic correlation (Currin, et al. 1991):

$$R(d_1, \cdots, d_k) = \prod_{j=1}^{k} R_j(d_j), \qquad (5.2)$$

where k is the number of predictor variables, and

$$R_j(d_j) = 1 - 6(\frac{d_j}{\theta_j})^2 + 6(\frac{|d_j|}{\theta_j})^3, |d_j| \in I_1 \qquad (5.3a)$$

$$R_j(d_j) = 2(1 - \frac{|d_j|}{\theta_j})^3, \quad |d_j| \in I_2 \qquad (5.3b)$$

$$R_j(d_j) = 0 \quad |d_j| \in I_3, \qquad (5.3c)$$

where $I_1=[0, \theta_j/2]$, $I_2=[\theta_j/2, \theta_j]$, and $I_3=[\theta_j, \infty]$.

There is no particularly compelling reason to use this instead of some other family of correlation functions. However, the piecewise cubic does have two appealing features: (i) $R(d_j)$ decreases to 0 as $|d_j|$ increases to θ_j, so that predictions can be made more local or less local by controlling θ_j, and (ii) $\hat{y}$ is a cubic spline in every t_j if the other t_j's are fixed. (This is because each element of C_{tD}, regarded as a function of t_j, is itself a cubic spline.) Cubic splines are quite highly regarded as interpolators and data smoothers; Bayesian prediction based on (5.2)-(5.3) produces an interpolating cubic spline with very little effort on the part of the user.

To select the parameters by "leave-one-out" cross-validation, each of the n experimental runs is deleted in turn, and the data at the remaining sites are used to predict y at the deleted site. Computationally, this is not as exhausting as it seems, since it can be shown that the error of prediction for response m,r at the deleted site i is

$$e_{mr,i} = q_i(g_{mr,i} - \mu_{mr}w_i)$$

where

$$g_{mr} = C_{DD}^{-1} y_{mr,D}$$

$$w = C_{DD}^{-1} J_n$$

and q is the inverse of the diagonal of C_{DD}^{-1}. Here C_{DD} is based on the *full* n-run design. The cross-validation root mean squared error is then:

$$\text{CVRMSE} = \left[\frac{1}{2345n} \sum_{i=1}^{n} \sum_{m=1}^{469} \sum_{r=1}^{5} e_{mr,i}^2 \right]^{1/2} \qquad (5.4)$$

Given the θ_j's, this is easy to minimize over the μ_{mr}'s, but minimization over the θ_j's requires iterative search -- this is by far the most (computer) time-consuming part of the prediction method.

To save time in the search for the optimal correlation parameters (θ_j's), we used only one response at each node, namely y_{m3}, the time to 50% filling. This seemed reasonable since we expected the other response functions to be similar in form. The values of μ_{m3}, m = 1,..., 469, and θ_j, j = 1,..., 4, were chosen to minimize (5.4) with r=3 and a divisor of 469n. Then, fixing the θ_j's at these values, we determined values of μ_{mr} for all m and r (again by cross-validation), this time using all 5 responses at each node.

In our first analysis, the cross-validation results at particular nodes indicated that the predictions of y_{mr} tended to be lower than the true values when the area of the charge was smaller than average and higher than the true values otherwise. That is, the predictions had the flow front moving too fast when the area of the charge was relatively small. We assumed that this was due to the increase in the height of the charge when the area is small (since the volume is held constant), which would presumably result in a slowing of the movement of the front as computed by TIMS. At any rate, we decided to introduce an additional predictor: $t_5 = (t_1 - t_2)(t_4 - t_3)$, which represents the approximate area of the charge, and we repeated the analysis. This reduced the cross-validation errors, so the area was used as a predictor in all subsequent predictions.

We then implemented the prediction equations for all responses in the form of a short computer code "FTIMS", which serves as a fast emulator of TIMS for investigating the effects of changing the shape and location of the charge. The input and output files for FTIMS are of exactly the same form as those for TIMS. The only difference is that the output for FTIMS is based on the prediction equations that followed from the computer experiment we described here, rather than the finite element solution to the differential equations of the model.

FTIMS converts the TIMS input into the site $(t_1, \ldots, t_5)$ at which predictions are desired. The 15×1 vector C_{tD} of correlations between this site and the design sites are computed using the values of θ_j, $j = 1,..., 5$, that we found to be optimal by the cross-validation criterion.

The predictions of the responses y_{mr}, $m = 1,..., 469$, $r = 1,..., 5$ are made using (5.1), where the 15×1 vector $w = C_{DD}^{-1}J_n$ (which is the same for all m, r) is provided by a fixed input file, as is the 15×1 vector $g_{mr} = C_{DD}^{-1} y_{mr,D}$ and the scalar μ_{mr}. FTIMS then adjusts the five predicted responses at each node, if necessary, to incorporate the knowledge that the true responses are nonnegative and nondecreasing. (We do not expect this adjustment to be needed very often, since the predictions interpolate data that satisfy these requirements. In the test case that we report below, the adjustment was needed at only two of the 469 nodes.) Monotonicity is enforced in a straightforward way, based on the notion that, of the five responses at node m, $\hat{y}_{m3}$ (i.e., the time to 50% filling) is generally the most reliable. This response is therefore left unchanged, and $\hat{y}_{m2}$ and $\hat{y}_{m4}$ are adjusted, if necessary, so that $\hat{y}_{m2} \le \hat{y}_{m3} \le \hat{y}_{m4}$. Keeping these three predicted responses constant, $\hat{y}_{m1}$ and $\hat{y}_{m5}$ are adjusted similarly.

To convert the five predicted responses at each node into estimates of $p(\tau)$ at the values of time desired, FTIMS again uses linear interpolation. The results are then printed in exactly the same form as the output produced by TIMS. The postprocessor that normally runs on TIMS output can then be applied to the output of FTIMS. This produces plots of the position of the flow front at various times. In a test case in which $t_1 = 0.7$, $t_2 = 0.3$, $t_3 = 0.75$, and $t_4 = 0.25$, examination of these plots showed the predicted front to be just a little ahead of the true front. On average, the predicted time to 50% filling in this case was 0.14 seconds less than the time calculated by TIMS; the root mean squared error for $\hat{y}_{m3}$ over all nodes was 0.23 seconds. In seven other randomly chosen test cases, the root mean squared error for $\hat{y}_{m3}$ over all nodes varied from 0.01 sec to 0.68 sec, with a median of 0.27 sec. In these test cases, the "true" times to 50% filling, averaged over all nodes, varied from 6.4-9.1 seconds.

The range of applications of the current version of FTIMS is obviously quite limited. Further generalizations, modifications, and tests would need to be made before it could be a considered a practical tool for optimizing this particular sheet molding process. Even at that stage, we would regard FTIMS as only an occasional replacement for TIMS, when one wants to consider many scenarios quickly and one is willing to accept an approximate result. The computing time for the run of FTIMS in the first test case described above was about 43 seconds on a Sun 3/50 Workstation, only 5 seconds of which were used to compute the predicted response vector at each node. The rest of the time was used for input and output. We have already noted that each run of TIMS takes 4-5 minutes on a Cray X-MP, so the availability of a practical and well-tested version of FTIMS would permit more extensive exploration of the effects of shape and position of the charge on the movement of the flow front.

Acknowledgements

We are grateful to Prof. Charles Tucker of the University of Illinois for allowing us to use the compression molding code (TIMS), to Dr. Alonzo Church of GenCorp Research for permission to use GenCorp's version of it, and to Dr. Daniel Fleming of GenCorp Research for sending us an executable version and helping us learn how to use it.

References

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments. *J. Amer. Statist. Assn.*, to appear.

Johnson, M., Moore, L. and Ylvisaker, D. (1990). Minimax and Maximin Distance Designs. *J. Statist. Planning and Inf.* **26**, 131-148.

Lindley, D. V. (1956), On a Measure of the Information Provided by an Experiment. *Ann. Math. Statist.* **27**, 986-1005.

Mitchell, T. J. (1974). An Algorithm for the Construction of 'D-Optimal' Experimental Designs. *Technometrics* **16**, 203-210.

Osswald, T.A. and Tucker, C.L. (1990). Compression Mold Filling Simulation for Non-Planar Parts. *Int. Polymer Processing* **5**, 79-87.

Sacks, J., Schiller, S.B., and Welch, W.J. (1989). Designs for Computer Experiments. *Technometrics* **31**, 41-47.

Sacks, J., Welch, W.J., Mitchell, T.J., and Wynn, H.P. (1989). Design and Analysis of Computer Experiments. *Statist. Sci.* **4**, 409-422. Comments and Rejoinder: 423-435.

Shewry, M. C. and Wynn, H. P. (1987). Maximum Entropy Sampling. *J. Appl. Stat.* **14**, 165-170.

Drug Design : Examining Large Experimental Designs

S. Stanley Young

Glaxo Inc.
Research Triangle Park, NC 27709

Abstract

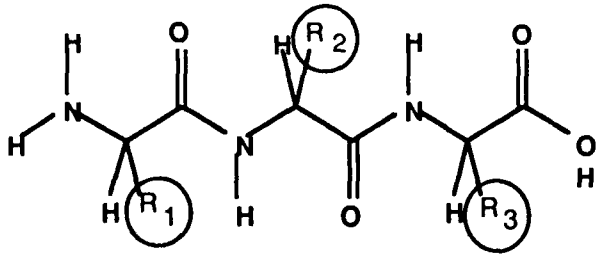
In the course of designing a new drug, thousands of candidate structures could be made and examined by empirical testing. Medicinal chemists would prefer some way of selecting a diverse subset from a list of candidates. Our statistical approach is to use experimental design technology for the selection process and to use computer visualization techniques for examination of the resulting design. A small peptide case is used as an example. The emphasis of this paper is on the value of visualization techniques in understanding the design and in explicating the design to Medicinal Chemists.

Introduction

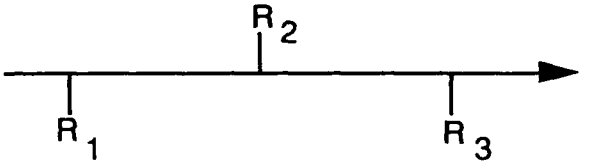
There are countless numbers of molecules that could be made for testing as potential drugs. Ten million different molecules have been made and registered; for most of these molecules that have the characteristics of typical drugs, there are millions of possible modifications. Since it is impossible to make all these molecules, there is a need to create diverse sets of molecules that span the range of possible structures. Hopefully, the "gaps" between the compounds in the design set will be small enough that important compounds are not missed. Our idea is to describe molecules numerically, use statistical experimental design software to create a design set, and examine the resulting design using 3D rotating scattergraph techniques. The process is illustrated using tripeptides.

What is a Tripeptide?

A tripeptide is a linear, directed sequence of three amino acids. There are three variable regions, called side groups, joined in sequence by amide linkages.



There is a beginning amide group, $-NH_2$, and a terminal carboxyl group, COOH. There are three variable regions denoted by R_1, R_2, and R_3 and there is a direction to the molecule. The following diagram captures these features.

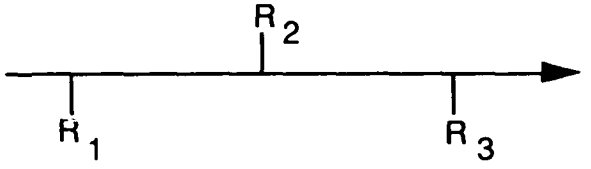


There are 20 naturally occuring amino acids, so there are 20x20x20=8,000 possible tripeptides. The cost of making enough compound for testing is about $500, so it would cost about four million dollars to make all possible tripeptides. Because this cost is too high and the process would take a long time to complete, it was decided to make a small, diverse set of tripeptides in the hope that a more cost effective discovery process would result.

Numerically Characterize a Tripeptide

Each of the variable regions of a peptide can be described using three numbers. The size can be measured as volume or surface area. Electronic properties can be measured. Also the lipophilicity of the side group can be measured. Lipophilicity is the propensity to dissolve in a water or oil environment. The blood is a water environment, as is the interior of a cell. Between the two is an oily cell membrane. Drugs typically have to pass from blood to the interior of cells so the water/oil relative solubility is important.

To numerically describe a tripeptide we combined these three numerical measures of side group properties across the three positions using linear scales.



Mean	1	1	1	Total
Linear	-1	0	+1	Gradient
Quad	-1	2	-1	Width

Note the three positions from left to right. For each of the three numerical descriptors, size, electronics, and lipophilicity, we created three scores, mean, linear, and quadratic. These scores have physical interpretations. For example, if one adds up the size of each side group at each of the three positions, then the score reflects the total size of the tripeptide. As the tripeptide is directed, the linear component measures a gradient along the tripeptide. Because the R_2 group is typically on the opposite side of the tripeptide from the R_1 and R_3 groups, the size quadratic score measures the width of the tripeptide.

There are three measures of properties of side groups and there are three scores determined for each so there are nine numerical measures of tripeptide properties. In addition to these scores, we computed various interactions among the nine scores to give a total of 34 descriptive variables, ie each of the 8,000 tripeptides was characterized with a vector of 34 numerical descriptors. The problem was to select about 100 tripeptides from the 8,000 so that the resulting set was as diverse as possible.

Experimental Design

There are about 10^{232} ways to select 100 objects from 8,000. We chose to use statistical experimental design software to make this selection. Our problem was much bigger than problems typically attempted using statistical experimental design software, so we had to improvise using various commercially available and internally developed software.

Experimental Design Software

1.	EChip	PC
2.	ACED	VAX or IBM
3.	OPTEX	IBM3090
4.	Inhouse Fortran	IBM3090

Because EChip on the PC would handle only relatively small problems, various iterative strategies were used. For example, one can select a trial design from a small random set of points, say 100 out of 800, do this several times, then make a final selection from the "winners" of each of the trial designs. Solutions on the PC took days to compute. ACED code was obtained from Dr. W. Welch of the University of Waterloo and modified to handle our large problems. We increased memory allocations and in certain instances compiled for a vector processor. We were able to obtain solutions in hours on our mainframes. Vector processing greatly speeded up the selection process. After much effort we were able to obtain a good 82 point design. This design had 55 percent G-optimality. Several designs consisting of 82 randomly selected points were checked. These random designs typically had G-optimality of 1 to 2 percent.

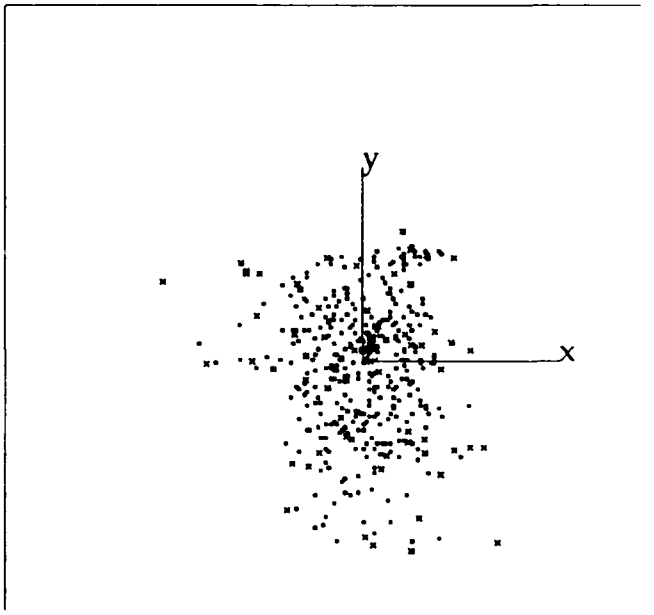
Several comments are in order. Orthogonal polynomials "fold" a dimension. For example, a tripeptide that has a large, small, large R-group in the three positions will be intermediate in size score for the mean polynomial and hence not selected as a vertex, but it will be large for the quadratic polynomial and will be selected as a vertex. The quadratic polynomial folded the size space moving a center point to an extreme point. D-optimal design software selects points that are vertices in a space. An obvious strategy is to select extreme points in the various dimensions as starting points for a design. We are attempting to saturate a low dimension space and do it by creating a higher dimension space that has the right vertices for the lower dimension space.

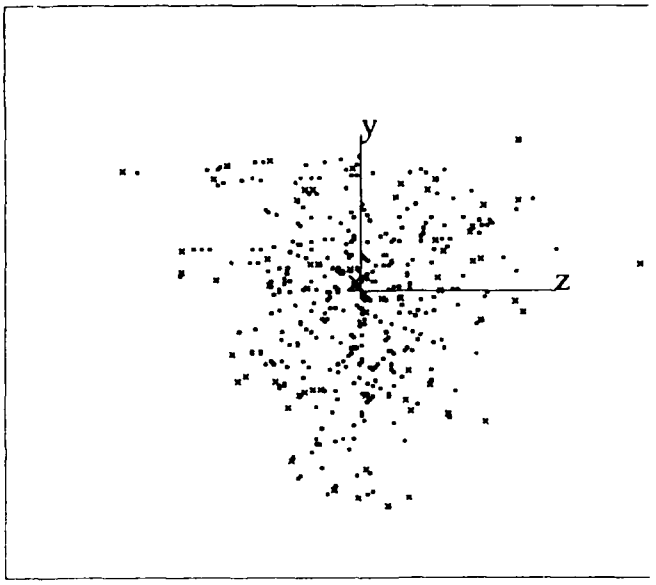
Software for Visualization

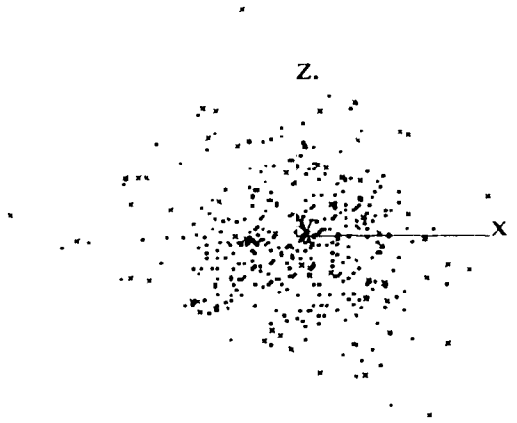
The experimental design software produces an analytical solution to the selection of representative tripeptides. Our resulting design had 82 points in a 34D space. To evaluate this design we used various 3D rotating scattergraph programs. This work was done on a Macintosh and we used MacSpin, Data Desk, and JMP. All three software packages were effective, although each had different features that helped in the visual evaluation of the design.

Our evaluation proceeded as follows. First we selected a random set of 800 points from the 8,000. This was necessary as rotation speed was a function of data set size Next, we added the 82 design points to the data set and marked them with color and/or a distinctive symbol. We then proceeded to look at various 3D projections of the random and design points.

The following figures shows three 90 degree views of the first three dimensions of the data.







In MacSpin we could slice through the cloud of points to examine the number and spacing of design points in planes of the data cloud.

Note that there are about 6,000 ways to select three dimensions from the 34. Also note that if a certain projection looks bad, design points are absent or poorly spaced, then there is no easy way to fix the design. Dropping one design point because it is visually close to another in a certain subspace and adding a point to fill in a void are likely to upset the design in other dimensions. The visualization is reassuring, but it does not offer an easy way to fix a perceived deficient design.

Discussion

Visualization helps assure the statistician that analytical techniques have been correctly employed. With many analytical techniques it can be difficult to detect if gross mistakes are made. It was quite assuring to the statisticians that the points of the final design seemed to saturate the 34D space. To make the 82 tripeptides cost about 50k dollars and took considerable time. Chemists and managers had to evaluate to reasonableness of the effort. Visualization was very effective in showing non-statisticians what was being proposed and some of the limitations, eg the gaps between design points, of the procedure. The collaborators in this project were chemists and Medicinal Chemists tend to think in highly visual ways. 3D rotating scattergraphs were very appealing to them.

Most of this work was done some time ago. In the meantime desktop computers have become much more powerful. Experimental design work could now be done on workstations, particularly if overnight or weekends were available.

The visualization of multiple dimensions is still a problem. With 3D rotation, color and symbols it is possible to get some feel for 4-5D, but we were working in 34D and we wanted to have good assurance of the saturation of 9D in our 34D space. After time consuming visual examination, we became comfortable that we had done a reasonable job, but it did take time and if we had found deficiencies, we would have had no recourse but to start all over again.

Computer Programs

ACED is a copyrighted program of Dr. W.J. Welch.
DataDesk is a trademark of Data Description, Inc.
EChip is a trademark of Expert in a Chip, Inc.
JMP is a trademark of SAS Institute Inc.
MacSpin is a trademark of D^2 Software, Inc.
Optex is a trademark of SAS Institute Inc.

Three graduate students from North Carolina State University worked on this project, Kim Carswell, Kris Latour, and Dan McCaffrey. Their work is gratefully acknowledged. Three professional statisticians also provided insights and software, Randy Tobias, and John Sall of SAS Institute Inc., and William J. Welch of U. of Waterloo.

Simulation Experiments For Neural Network Learning

DAVID S. NEWMAN
Applied Statistics
Boeing Computer Services
P. O. Box 24346, MS 7L-22
Seattle, Washington 98124-0346

Email: dnewman@caissa.boeing.com

Abstract

This paper investigates approaches to the design of simulation experiments for training neural networks which are to be used as classifiers. Hierarchical clustering applied to the ART1 and ART2 (ART = Adaptive Resonance Theory) neural network architectures developed by Carpenter and Grossberg [20,21] is the basis for the approach. A series of experiments based on this approach will test the performance of ART1 and ART2 as pattern classifiers against a variety of real and artificial data sets. The issues to be investigated in these experiments include the sensitivity of performance to a variety of network parameters, pattern characteristics, and pattern presentation disciplines. A background is provided for those unfamiliar with neural networks in general, and with Grossberg's approach in particular.

Some Background on Neural Networks

Neural networks are the latest super-hyped glamor technology, following hard on the heels of "artificial intelligence," and many statisticians are no doubt wondering how much substance, if any, lies behind the smoke. Many of the claims are of course exaggerated, and many techno-promoters are pushing the use of neural network algorithms where (for example) a standard linear regression analysis is sufficient to do the job. Nevertheless, neural networks which can outperform traditional statistical, signal processing and pattern recognition approaches already exist and have proved their worth in a number of applications. As with artificial intelligence, there are unresolved theoretical and practical issues of "machine learning" which on the one hand are in desperate need of statistical assistance, and on the other hand stretch both theoretical and applied statistics to their limits.

There are many motivations for investigating neural networks, and a large number of different approaches. Understanding brain function was the initial motivation for the study of neural networks, and remains the primary motive for that aspect of the subject which Arbib [1] has called "computational neuroscience." In what must be called the pioneering paper of the subject, McCulloch and Pitts [2] described how neurons with firing thresholds function as logic gates, and how interconnected groups of neurons could perform operations describable by a logical calculus. In their next paper [3], these authors proposed a totally different computational model of memory: an analog spatial map developed as a consequence of the dynamics of neural activity. While the analog approach has since predominated in neuroscience, the two models are not contradictory, but complementary, a point emphasized by Von Neumann [4] and apparent in many current neural network models. Neural analogies were also central in Wiener's vision of a new discipline of cybernetics [5].

The complementary nature of logical (digital) and dynamic (analog) activity provides the second motive for investigating neural networks, that of designing massively parallel hybrid computers which mimic to some extent the architecture of the brain. The first "neurocomputer" was Rosenblatt's "perceptron" [6]. The subsequent flurry of excitement led to a brief period of heavy government funding of "brain machines" in the 1960's in an atmosphere of techno-hype that makes the current round pale by comparison. Hardware limitations, early technical failures, and the devastating impact of Minsky and Papert's [7] analysis led to a 15-year long "dark age" in which neural networks were eclipsed by "artificial intelligence."

The current revival began in 1982 with the introduction of the Hopfield network [8,9]. Since then, many neurocomputing algorithms have been proposed or revived, the most popular being the Boltzmann machine [10], which owes a great deal to the work of statisticians Geman and Geman [11], and above all, back-propagation [12,13], which has become almost synonymous in many people's minds with neural networks. Ease of implementation and some impressive, well-crafted applications of "backprop" have unfortunately overshadowed its limitations, and the

importance of the steady, ongoing developments which continued in spite of the "dark age."

Another impetus for the current revival of neural networks is the availability of new technologies for hardware implementation. Large-scale integrated circuits operating in either the familiar digital mode or in analog (subthreshold) mode based on neural architectures are already being produced, and optical processors are in the design stage.

Statistical theory is already playing a role in understanding learning performance [14,15]. In a closely related development, statistical decision theory is the latest "hot topic" in the field of machine learning, not only for neural networks but also for "conventional" artificial intelligence [16].

Learning Properties of Neural Networks

Unlike a digital computer, which performs its computations by following a series of programmed instructions, a neural network is trained by presenting it with a series of examples of the inputs for which outputs are desired. If desired outputs are presented simultaneously with inputs, and the neural net adjusts its connection weights so that its output approximates the desired output, the training is called supervised; otherwise it is unsupervised. Once the training period is over, the neural network is presented with new inputs for recognition. The parallel with statistical estimation and prediction is immediately apparent. A closer examination reveals that supervised learning resembles nonparametric regression analysis if the output is continuous, and nonparametric discriminant analysis if it is discrete, while unsupervised learning corresponds to either cluster analysis or nonparametric density estimation.

A major difference between training a neural network and applying a statistical algorithm is that the input examples are presented to the network one at a time. In this respect a neural network resembles a recursive statistical algorithm such as the Kalman filter. But what many neural networks learn depends on the order in which the patterns are presented, which is typically not the case for a statistical algorithm. It is common practice to cycle through the training set repeatedly until the network weights cease to change, or some other criterion for stability is satisfied. In some cases the order of presentation is varied randomly or systematically from cycle to cycle, in others it is not.

Despite these differences, it is clear that statistical methods will be useful in evaluating neural network performance. The usual practice at present is to set aside a portion of the training set for testing, and evaluate the performance of the trained network on the testing set. Testing corresponds to the statistical practice of cross-validation, and the issues surrounding the tradeoff between bootstrapping and cross-validation are especially complex in this context.

Grossberg's Neural Principles and Adaptive Resonance

While many authors have modeled specific aspects of brain function, the most comprehensive theory and the broadest collection of models of cognitive activity has been produced by Stephen Grossberg and his collaborators [17,18,19]. Their approach is to search for and apply general principles underlying a wide range of experimental evidence from neurophysiology and psychophysics.

Grossberg begins with a physically-based model of neural activity, leading to a system of non-linear differential equations for synapse activities and connection weights. The time constants of the connection weights, which store long-term memory, are much longer than those of the synapses which constitute short-term memory. Unlike back-propagation and other feedforward algorithms which do not always converge, global dynamic stability is built in to the structure of these equations. The details of the differential equations are highly flexible, allowing for a wide variety of architectures capable of representing important aspects of vision, speech, memory, conditioned and unconditioned responses, and even reasoning. For a more detailed overview, see the references above, especially Chapters 1 and 13 of [18]. (Chapts. 1 and 12 of [18] also appear in Anderson & Rosenfeld's collection of "classic" papers [AR] as Chapts. 24 and 19 respectively.)

Adaptive Resonance Theory (ART) refers to the neural feedback mechanisms which have been developed to ensure stable encoding of incoming stimuli within the framework of this broader theory. ART1 and ART2 are general-purpose neural network modules based on ART principles. Functionally, they provide a means for rapid, unsupervised learning and classification of incoming patterns (represented as extremely high-dimensional vectors) based on "reset" of poor matches with generalizations of previously learned patterns (templates), and "resonance" with good ones. ART1 is designed for binary ("black & white") patterns, while ART2 operates on continuous-valued ("grey-scale") patterns. Both ART1 and ART2 depend on a single parameter called the "vigilance level" which determines the fineness of the resulting classification. Higher vigilance results in a larger number of classes. Details will be found in Carpenter and Grossberg [20,21]. While most of the computing tasks performed to date by these "ART units"

involve pattern learning & recognition, this limitation is not inherent.

As with many other neurocomputing algorithms, the natural computer implementation of ART is on an integrated silicon chip, optical circuit or other physical device. But until such devices are more readily available, simulations of neural algorithms will be carried out on digital computers. Unlike some other neurocomputing algorithms, "fast learning" special cases of the ART1 and ART2 algorithms are easily coded, and many experiments and practical applications of ART can be implemented in a digital computing environment.

Order Dependence of Neural Network Learning: A Remedy for ART1

The fast learning versions of ART1 and ART2 may be regarded as clustering algorithms for the purpose of understanding training. In some contexts the order of pattern presentation may be meaningful, and it should be allowed to affect the resulting classification. But for many technological applications, the presentation order of the training patterns is irrelevant, and the dependence of the classification of training patterns and templates on input order is undesirable. Furthermore it is a sign that the resultant classification is statistically inconsistent.

Under these conditions it would be desirable to find a way to train the network to find a classification of the training patterns which is free from this problem. For ART1 it is possible to do so, and it appears likely that it will also be possible for ART2. The resulting classification may be described as a "canonical" one (it is not clear that it is unique). If the nodes of the ART1 unit are encoded with the templates corresponding to this canonical classification, each training pattern will "resonate" automatically with the correct node, regardless of the order of presentation.

This result depends on the fact that the ART1 and ART2 algorithms may be characterized by similarity measures of the type used in cluster analysis. Furthermore, as a result of a 1978 theorem of Grossberg (Chapter 12 of [18] or Chapter 19 of [AR]), ART2 will correctly identify pattern classifications which are sufficiently separated from one another. The author of this paper has shown a similar result for ART1 [22], although the similarity measure proposed in that paper must be slightly modified. By putting these two facts together, it is possible to show that a single-linkage hierarchical cluster analysis of the training patterns based on the similarity measure will produce a nested family of canonical classifications corresponding to a family of ART1 units with vigilance levels ranging from zero to one. (Vigilance zero places all patterns in one class, while a value of one creates a separate class for each pattern). More specifically, the open interval (0,1) is broken into open subintervals: each subinterval is a range of vigilance level values which will give a specific classification. Relating learning to single-linkage cluster analysis may prove critical in establishing the statistical consistency of the learning process; see Hartigan [23]. The mathematical demonstration of these results will be presented in a future publication.

Thus the order-dependence of the learned result of training has been eliminated by a statistical algorithm which "simulates" the neural network, handling the training elements simultaneously instead of one at a time. It may be feasible to characterize other neural network classification algorithms by a similarity measure, and use the same approach to achieve order-independent training of the network.

Simulation Experiments with Neural Network Training

In many applications, an important part of the simulation of ART1 learning and recognition performance will involve the use of hierarchical clustering to remove the effect of training order on what is learned. This presupposes, of course, that the investigator has no predetermined classification in mind. If he does, and this classification agrees with a canonical one, his problem is solved (and ART1 is an extremely appropriate architecture for his problem!). When this is not the case, a certain amount of classification error may be tolerated (as it generally is in discriminant analysis).

If the training elements are very high-dimensional, very complex, or very numerous, this approach may not be computationally feasible. Even for applications where it is practical, many other statistical issues must be resolved. The need to test the performance of the network raises the issue of "bootstrapping vs. cross-validation" noted earlier. For example, how does one compare an assortment of hierarchical clusterings arising from bootstrapped or cross-validated training/testing samples? The answer will depend in part on whether or not memory (templates in the case of ART1) will be "frozen" in the implementation.

References

[1] M. Arbib (1987), *Brains, Machines, and Mathematics, 2nd Edition*. New York: Springer-Verlag.

[2] *W. S. McCulloch and W. Pitts (1943), "A logical calculus of the ideas immanent in nervous activity,", *Bulletin of Mathematical Biophysics 5*, 115-133.

[3] *W. Pitts and W. S. McCulloch (1945), "How we know universals: the perception of auditory and visual forms," *Bulletin of Mathematical Biophysics 9*, 127-147.

[4] John von Neumann (1958), *The Computer and the Brain.* New Haven: Yale University Press.

[5] N. Wiener (1948), *Cybernetics: or Control and Communication in the Animal and the Machine.* Wiley; 2d edition, 1961, The MIT Press.

[6] *F. Rosenblatt (1958), "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review 65*, 386-408.

[7] M. Minsky and S. Papert (1969), *Perceptrons: An Introduction to Computational Geometry.* Cambridge, Mass.: The MIT Press.

[8] *J. J. Hopfield (1982), "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. 79*, 2554-2558.

[9] *J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons, *Proc. Nat. Acad. Sci. 81*, 3088-3092.

[10] *D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, (1985), "A Learning Algorithm for Boltzmann Machines," *Cognitive Science 9*, 147-169.

[11] *S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transations on Pattern Analysis and Machine Intelligence PAMI-6*, 721-741.

[12] *D. E. Rumelhart, G. E. Hinton and R. J. Williams (1986), "Learning Internal Representations by Back-propagating Errors," *Nature 323*, 533-536.

[13] *D. E. Rumelhart, G. E. Hinton and R. J. Williams (1986), "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*," Vol. I, D. E. Rumelhart and J. L. McClelland, eds. Cambridge, Mass, MIT Press, 318-362.

[14] A. R. Barron and R. L. Barron (1988), "Statistical Learning Networks: A Unifying View," *Proceedings of the 20th Symposium on the Interface Between Computer Science and Statistics*, 192-203.

[15] H. White (1989), "Some Asymptotic Results for Learning in Single Hidden-Layer Feedforward Network Models, *Journal of the American Statistical Association 84*, 1003-1013.

[16] E.B. Baum and D. Haussler (1988), "What Size Net Gives Valid Generalization?" *IEEE International Symposium on Information Theory, Kobe,* Japan. To appear in *Neural Computation.*

[17] S. Grossberg (1982), *Studies of Mind and Brain.* Dordrecht, Holland: Reidel Publishing Co.

[18] S. Grossberg, ed. (1987), *The Adaptive Brain (2 Vols.).* Amsterdam: North-Holland/Elsevier.

[19] S. Grossberg, ed. (1988), *Neural Networks and Natural Intelligence.* Cambridge, Mass.: The MIT Press.

[20] G. A. Carpenter and S. Grossberg (1987), "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Processing 37*, 54-115.

[21] G. A. Carpenter and S.Grossberg (1987), "ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns," *Applied Optics 26, No.23*, 4919-4930.

[22] D. S. Newman (1990), "Characterization of some learning properties of ART1 machines," presented at the Conference on Neural Networks for Automatic Target Recognition, Wang Institute of Boston University.

[23] J. A. Hartigan (1981), "Consistency of Single Linkage for High-Density Clusters," *Journal of the American Statistical Association 78*, 388-394.

* All papers above marked with an asterisk, and many other important advances in neural networks up to 1987, appear (with useful prefaces by the editors) in

[AR] James A. Anderson and Edward Rosenfeld, editors (1988), *Neurocomputing: Foundations of Research.* Cambridge, Mass., MIT Press.

CLASSIFICATION BY EM-TRAINED DYNAMIC ARTIFICIAL NEURAL NETS BASED ON HIDDEN PERCEPTRONS.

ARTHUR NÁDAS
Speech Recognition Group, Computer Science Department
IBM T.J. Watson Research Center, Box 704, Yorktown Heights NY 10598

ABSTRACT

We propose to classify points in $\mathbb{R}^d$ by functions related to two-layer (a single hidden layer) feedforward artificial neural nets (ANNs). These functions, dubbed dynamic ANNs (DANNs), arise in a rather natural way from probabilistic and also statistical considerations. We treat the binary classification problem and outline an approach to the n-ary classification problem. There are two key ideas. The probabilistic idea is that DANNs are conditional probabilities in certain mixture models. The statistical idea is that these models, and hence the DANNs defined by them, are conveniently trainable by an expectation - maximization (EM) algorithm.

INTRODUCTION

Consider classification of points $x \in \mathbb{R}^d$ (feature vectors) by using continuous functions of x for the probabilities of classes. For binary classification this means any continuous function $\mathbb{R}^d \to [0, 1]$ and for n-ary classification any continuous function from $\mathbb{R}^d$ to the $n-1$ dimensional simplex in $[0,1]^n$. In this paper we focus on binary classification and merely sketch the generalization to n-ary classification.

Cybenko (1989) has shown that any continuous function defined on a compact set in $\mathbb{R}^d$ can be uniformly approximated by a two-layer (= one hidden layer) artificial neural net (ANN) f_C, i.e. a function of the form

$$(1) \qquad f_C(x) = \sum_{j=1}^{k} \alpha_j \sigma\left(\sum_{i=1}^{d} w_{ij} x_i + w_{0j}\right)$$

where k is a sufficiently large integer, $\sigma: \mathbb{R} \to [0,1]$ is a sigmoid and the α_i, w_{ij} are constants. Barron (1991) has given a bound on the number k of of summands required for approximation within prescribed precision.

A different class of approximating functions, which we call dynamic artifical neural nets (DANNs) will be introduced. Unlike (1), these functions are based on a probabilistic description of the classification process and hence enjoy certain properties which we shall exploit. A DANN $f: \mathbb{R}^d \to [0, 1]$ is defined as a conditional expectation

$$(2) \qquad f(x) \equiv E[U(Y) \mid X = x]$$

where U is the unit step function

$$(3) \qquad U(y) = \begin{cases} 0 & \text{if } y < 0 \\ 1 & \text{if } y \geq 0 \end{cases}$$

and Y is jointly distributed with X. If Y were a linear function of X then U would be a perceptron, in fact a 'threshold logic unit'. We shall construct Y by adding noise to a randomly chosen linear function of X. We cannot observe the identity of such a linear function, hence the 'hidden perceptron' of the title. This construction produces a joint distribution P for (X,Y) in which

$$(4) \qquad f(x) = \sum_{j=1}^{k} \pi_j(x) \sigma\left(\sum_{i=1}^{d} w_{ij} x_i + w_{0j}\right).$$

The form in (4) differs from f_C in (1) only in that the real constants α_j are replaced by certain hidden state probability functions $\pi_j(x)$. We shall argue elsewhere that, as is the case for the ANN functions f_C, the class of DANN functions f can uniformly approximate on a compact set any function which is continuous there. It will become clear that DANNs are not ANNs except in the degenerate case of where the π_j are constant in x; this corresponds to a certain statistical independence in our model. Conversely an ANN with one or more negative α_j cannot be a DANN so neither class contains the other.

The motivation for this approach to classification is both theoretical and practical. On the one hand we wish to use statistical optimization criteria, such as ML or Bayes, for training the classifier, and at the same time we wish to accomplish the training with the simplest numerical algorithm. For this reason we complete the choice of the form of the joint distribution of (X,Y) so as to also allow the construction of an EM algorithm (Dempster et al.(1977), Meilijson (1989)) for estimating its parameters. The EM algorithm for learning the distribution P thus becomes an indirect but simple training algorithm for DANNs $f(x)$. By the way of contrast: the standard current approaches using ANNs consist of some curve fitting ('back

propagation" etc.) methods. These typically vary the constants defining f so as to minimize the sum of squared errors in predicting the unit sum bitstring z which encodes (see the discussion below) the index of the class corresponding to x by its expectation $f(x)$.

THE CLASSIFICATION PROBLEM

A probabilistic version of the classification problem is this: given a completely specified joint distribution P of the random pair (X,I) with $X \in \mathbb{R}^d$ an d $I \in \{0, \ldots, n-1\}$, find a classifier function $\Psi: \mathbb{R}^d \to \{1, \ldots, n\}$ which minimizes the probability of misclassification $P(\Psi(X) \neq I)$. The well known solution is to let $\Psi(x)$ be the least (say) nonnegative integer which achieves $\max_{1 \leq i \leq n} P(I = i | X = x)$.

In our approach to classification a coded version Z of the class index I is introduced. Corresponding to a random feature vector X let f.Z memberof lbrace 0,1 rbrace sup m be a one-to one encoding of the class index I as a bitstring. For the sake of concreteness the reader may wish to regard Z as the binary expansion of the class index I and in this case m is the least integer $m \geq = \log_2 n$. The more popular encoding consists of encoding the event $I = i$ as the bitstring associated with the $i-th$ vertex of the n dimensional simplex; in this case $m = n$. It is obvious that from the probabilistic point of view it does not matter which one-to-one encoding one chooses. Contrast this with the statistical point of view, i.e. the typical practical situation wherein the joint distribution of (X,I) is not specified completely; in this case one has some training data

$$T = \{(x_t, i_t) | t = 1, \ldots, N\} \tag{5}$$

to work with instead. We assume that T is a random sample from the distribution of (X,I). The usual statistical approach, which (for the lack of a better idea) we also adopt, is to estimate the joint distribution P by a distribution P^T and thereafter ignore the error of the estimate. Since some functions are easier to estimate than others, it is no longer clear that different encodings are equally good. We do not pursue the encoding issue but simply assume that some encoding is specified. It is likely that ultimately some problem dependent encoding will will be preferred to either of the two simple encodings mentioned above; an encoding chosen to optimize the performance of the trained classifier.

BINARY PERCEPTRON AND ITS DANN

We now fix $m = 1$ so Z is just a random bit. A joint distribution for the input-output pair $(X,I) \in \mathbb{R}^d \times \{1, \ldots, n\}$ implies a joint distribution for the coded version (X,Z). The probability element for (X,Z) has the form

$$g_d(x) p(z|x) \tag{6}$$

where g_d is a density on $\mathbb{R}^d$ and for fixed x, $p(z|x)$ is a probability on $\{0,1\}$. In the terminology of the EM algorithm, a sample from the distribution of the observable pair (X,Z) is the 'incomplete data' and $g_d(x)p(z|x)$ is the incomplete data model. When this is parametrized as $g_d(x|\theta)p(z|x,\theta)$ then the corresponding incomplete data likelihood function is

$$l_{(\theta)} = \sum_{t=1}^{T} \log g_d(x_t|\theta) + \log p(z_t|\theta) \tag{7}$$

A direct numerical approach attempts to maximize (7). Instead, the EM algorithm iteratively maximizes a different function which however has a maximum at the same θ.

In order to model the generation of the data and to construct an EM algorithm, we now introduce the complete data model. The idea is to make local models of the joint distribution of the feature vector X and a noisy locally linear function Y whose only purpose is to define the classifying bit Z. Locality is achieved through the use of a mixing variable $J \in \{1, \ldots, k\}$ with $P(J = j) = \alpha_j$. Let

$$(X,Y,J) \quad X \in \mathbb{R}^d, \quad Y \in \mathbb{R}^m, \quad J \in \{1, \ldots, k\} \tag{8}$$

denote the complete data. Conditionally on $J = j$ the density of (X,Y) is $d+1$ dimensional Gaussian with mean vector

$$\begin{pmatrix} \mu_j \\ \nu_j \end{pmatrix} = \begin{pmatrix} \mu_{1j} \\ \cdot \\ \cdot \\ \cdot \\ \mu_{dj} \\ \nu_j \end{pmatrix} \tag{9}$$

and covariance matrix

$$\Gamma_j = \begin{pmatrix} \Gamma_{11j} & \Gamma_{12j} \\ \Gamma_{21j} & \Gamma_{22j} \end{pmatrix}. \tag{10}$$

Observe that X has a Gaussian mixture distribution describing the feature space and Y is the noisy signed distance to a hyperplane determined by the coefficients of the random linear function $E(Y|X,J)$. (Actually the Gaussian assumption is not necessary; any tractable d-dimensional kernel will do here. The conditional distribution of Y given both $X = x$ and $J = j$ can also be replaced by any tractable non-Gaussian distribution but the latter must have a location parameter which is

linear in x. The $d+1$- dimensional Gaussian assumption automatically satisfies this condition.)

Without loss of generality we can parametrize the rest of Γ_j as follows:

$$\Gamma_{12j} = \Gamma'_{21j} = \Gamma_{11j}\beta_j \tag{11}$$

where β_j are the regression coefficients of the regression of Y on X given $J=j$ and the variance of Y given $J=j$ is

$$\Gamma_{22j} = \gamma_j^2 + \beta'_j \Sigma \beta_j \tag{12}$$

where γ_j^2 is the conditional variance of Y given not only $J=j$. but also $X=x$ (residual variance). We now put

$$Z = U(Y) \tag{13}$$

where U is the unit step. Then

$$\begin{aligned} E(Z|X=x) &= \sum_{j=1}^{k} P(J=j|X=x)P(Z=1|X=x,J=j) \\ &= \sum_{j=1}^{k} \pi_j(x)P(Y>0|X=x,J=j) \\ &= \sum_{j=1}^{k} \pi_j(x)\Phi\left(\frac{1}{\gamma_j}\left\{v_j + \sum_{l=1}^{d}\beta_{lj}(x_l - \mu_l)\right\}\right) \end{aligned} \tag{14}$$

where $\pi_j(x) = \pi_j(x)(x) = P(J=j|X=x)$ and Φ is the standard normal integral. Setting $w_{lj} = \frac{\beta_{lj}}{\gamma_j}$ and $w_{0j} = \frac{v_j - \sum_{i=1}^{d}\beta_{ij}}{\gamma_j}$, and choosing the sigmoid to be standard Gaussian CDF

$$\sigma(y) = \Phi(y) \equiv \int_{-\infty}^{y} \phi(u)du \tag{15}$$

with $\phi(u) = (2\pi)^{-.5}e^{-\frac{1}{2}u^2}$, we see that the DANN in (4) is precisely the conditional expectation

$$f(x) = E(Z|X=x) \equiv P(Z=1|X=x). \tag{16}$$

THE TRAINING ALGORITHM

Let θ denote a vector whose components form a list of all the unknown parameters of the distribution

$$\theta = ((\alpha_j, \mu_j, v_j, \Gamma_j)|j = 1, \ldots, k). \tag{17}$$

After initializing $\theta = \theta^0$ the EM algorithm for exponential family (our setup) iterates two steps, (E): get the conditional expected values of all complete data sufficient statistics given the incomplete data and (M): use these to estimate their unobservable versions and hence to estimate the complete data log likelihood. The latter is then (trivially) maximized to get $\theta^{(r)}$ in the r-th iteration.

THE E-STEP OF TRAINING

The E-STEP in the usual Gaussian mixture problem estimates all the unobservable complete data sufficient statistics. These are unobservable because J, the mixture index, is hidden. Our problem is similar but differs from this in that in addition to the unavailability of J, the r.v. Y. is also hidden except for its sign. Thus the conditional expectations required here are based on less information than in the usual mixture problem. Let $\delta(A)$ be 1 or zero as A occurs or not. In our setup the complete data sufficient statistics are

$$\begin{aligned} N_j &= \sum_{t=1}^{T} \delta(J_t = j) \\ SX_j &= \sum_{t=1}^{T} \delta(J_t = j)X_t \\ SXX'_j &= \sum_{t=1}^{T} \delta(J_t = j)X_t X'_t \\ SYX'_j &= \sum_{t=1}^{T} \delta(J_t = j)Y_t X'_t \\ SYY_j &= \sum_{t=1}^{T} \delta(J_t = j)Y_t^2 \\ SY_j &= \sum_{t=1}^{T} \delta(J_t = j)Y_t \end{aligned} \tag{18}$$

The corresponding conditional expectations are

$$\begin{aligned} \overline{N_j} &= \sum_{t=1}^{T} p_j(X_t, Z_t) \\ \overline{SX_j} &= \sum_{t=1}^{T} p_j(X_t, Z_t)X_t \\ \overline{SXX'_j} &= \sum_{t=1}^{T} p_j(X_t, Z_t)X_t X'_t \\ \overline{SY\bar{X}'_j} &= \sum_{t=1}^{T} p_j(X_t, Z_t)\bar{Y}_t(j)X'_t \\ \overline{SY\bar{Y}_j} &= \sum_{t=1}^{T} p_j(X_t, Z_t)\overline{Y^2}_t(j) \\ \overline{SY_j} &= \sum_{t=1}^{T} p_j(X_t, Z_t)\bar{Y}_t(j) \end{aligned} \tag{19}$$

where

$$p_j(x_t, z_t) = P(J_t = j|X_t = x_t, Z_t = z_t; \theta),$$

$$\bar{Y}_t(j) = E(\delta_j(J)Y|X, Z; \theta), \tag{20}$$

$$\overline{Y^2}_t(j) = E(\delta_j(J)Y^2|J=j, X=x; \theta)$$

with $\theta = \theta^{r-1}$. It is easily checked that $p_j(x,1)$ is given by

$$(21)\qquad \frac{\pi_j(x)\Phi\left(w_{0j}+\sum_{l=1}^{d}w_{lj}x_l\right)}{\sum_{i=1}^{k}\pi_i(x)\Phi\left(w_{0i}+\sum_{l=1}^{d}w_{li}x_l\right)}$$

and where, similarly, $p_j(x,0)$ is given by

$$(22)\qquad \frac{\pi_j(x)\Phi\left(-w_{0j}-\sum_{l=1}^{d}w_{lj}x_l\right)}{\sum_{j=1}^{k}\pi_i(x)\Phi\left(-w_{0i}-\sum_{l=1}^{d}w_{li}x_l\right)}$$

We still need to define $\overline{Y}(j)$, $\overline{Y^2}(j)$, and $\overline{XY}(j)$. Since $E(XY|X,Z) = X\,E(Y|X,Z)$, we need only the first two. We have for $r = 1,2$:

$$(23)\qquad \begin{aligned}&E(\delta_i(J)Y^r|X=x,Z=z)\\&\quad=p_i(x,z)E(Y^r|X=x,Z=z,J=i).\end{aligned}$$

The last expectation may be evaluated as follows. Writing

$$(24)\qquad \xi_i=\xi_i(x)=v_i+\sum_{j=1}^{d}\beta_{ji}(x_j-\mu_j)$$

we have

$$(25)\qquad \begin{aligned}&E(Y^r|X=x,\ Y>(\le)0,J=i)\\&\quad=E((\xi_i(x)+\gamma_iY_{01})^r|Y_{01}>(\le)-\frac{\xi_i(x)}{\gamma_i})\end{aligned}$$

where Y_{01} is a standard scalar Gaussian r.v. with mean zero and variance one; its two conditional moments are not hard to obtain in closed form and we omit them.

THE M-STEP OF TRAINING

Assemble the results of the E-STEP to form estimates of the k mean vectors and the k covariance matrices of the model and extract the required regression coefficients β_j and residual variances γ_j^2. Compute the neural net weights and thresholds w_{ij} after the last iteration.

NUMERICAL EXPERIMENTS

We trained various versions of the model on data generated by the model itself. In these experiments we verified that the model behaves as the theory predicts; in particular we were in each case able to recover the parameters of the generating model with reasonable accuracy. In addition, we trained the model on 50 dimensional speech data belonging to the phones 'M' and 'N'. We tested the model on an independent set of test data from the same phones. Our best results were obtained with 5 hidden states yielding an error rate of 7.5 percent on the test data and 3.0 percent on the training data. This is virtually identical with results, on the same data, that had been obtained by training and testing comparable neural nets with the usual fixed weights.

THE n-CLASS PROBLEM.

Suppose that the bitstring encoding $Z \in \{0,1\}^m$ is given and (X,Z) has some joint distribution. Define the complete data model by

$$(26)\qquad (X,Y,J)\quad X\in\mathbb{R}^d,\quad Y\in\mathbb{R}^m,\quad J\in\{1,\dots,k\}.$$

and set $Z_i = U(Y_i)\quad i=1,\dots,m$. In this case conditionally on $J=j$ the density of (X,Y) is chosen to be $d+m$-dimensional Gaussian. For convenience in computing $P(J=i|X=x)\equiv P(Z=z|X=x)$ we take the conditional covariance matrix of Y given both $X=x$ and $J=j$ to be diagonal. Then $P(Z=z|X=x)$ is given by

$$(27)\qquad \sum_{j=1}^{k}\pi_j(x)\prod_{i=1}^{m}p_{ij}(x)^{z_i}(1-p_{ij}(x))^{1-z_i}$$

where

$$(28)\qquad p_{ij}(x)=P(Y_i>0|X=x,J=j).$$

We shall argue elsewhere that as in the case of $m=1$ the Bayes classifier based on the true joint distribution of (X,Z) can be uniformly approximated with such forms. The EM algorithm is again applicable; the only new object is the conditional covariance between components of Y given both X and J. While this is zero by construction, enforcing this constraint in the M-step requires some care.

REFERENCES

1. Barron, A.R. (1991), 'Universal Approximation Bounds for Superposition of a Sigmoidal Function', to be presented at the IEEE Information Theory Symposium, Budapest, June 1991.
2. Cybenko, G. (1989), 'Approximation by Superpositions of a Sigmoidal Function', Math. Control Signals Systems, 2, 303-314.
3. Dempster, A.P., Laird, N.M., Rubin, D.B. (1977) 'Maximum Likelihood from Incomplete Data via the EM Algorithm', J.Roy.Stat.Soc., B, 39,1-38.
4. Meilijson, I. (1989), 'A Fast Improvement to the EM Algorithm On Its Own Terms', J.Roy.Stat.Soc., B, 51, 127-138.

COMPARING MATHEMATICAL AND ALGORITHMIC MODELING IN BIOLOGY

G. Arthur Mihram, Ph. D.
P. O. Box No. 1188
Princeton, NJ 08542-1188

Danielle Mihram, Ph.D.
Doheny Memorial Library
University of Southern California
Los Angeles, CA 90089

Abstract

The paper uses several examples to illustrate a distinctive difference between alternative models of biological systems: those of the mathematical vs. those of the algorithmic format. Primary among these comparisons are the models of researchers dealing with neural networks versus those of artificial intelligence [AI] researchers who predicate their work on the cognitive sciences. We show how the literature of biology itself reveals why one approach to the modelling of biological systems is more likely to succeed than the other. We compare historically the acclaimed successes of non-mathematical biologists [e.g., Darwin's ORIGIN OF THE SPECIES and Lorenz's paper, "Fashionable Fallacy of Dispensing with Description "].

We include in the paper a review of the literature dealing with the principles for conducting the design and analysis of experiments with computerised stochasic models, applicable whether their dynamics are 'controlled' within the computer mathematically or, alternatively, algorithmically. Exemplary models of AI systems are the current software packages being implemented throughout the research and university communities: viz., bibliographic retrieval progammes which, e.g., include statistical analyses for the purpose of suggesting alternative subject-search strategies.

1 Introduction

For the past four decades [since, e.g., McCulloch and Pitts (1943)], researchers in AI have become very slowly aware of the distinctive advantage which algorithmic models possess over those other computerised models of the strictly mathematical format. Quite recent authors [e.g., Amit (1989)] persist, particularly in the literature of neural networks, with their fascination with mathematical modelling, as though the success of the mathematically-expressed Newtonian models (of physics) will automatically be conferred on their own work.

On the other hand, Mihram (1973) noted that philosophers Sayre and Crosson (1963) had been struggling with the non-mathematical ("non-formalized") nature of computer programming as it might affect the modelling of mind, a mental struggle being conducted as well in the context of computerised modelling of social systems in that same decade by the mathematician Kemeny (1969).

Completely generalizing this struggle to biological systems, including not only neural networks/organs but also socio-political organizations, was the 1975 Ludwig von Bertalanffy Lecturer, J.G. Miller (1978). Miller notes that there are seven levels of living systems, from the cell to the 'supra-national society', and that at any level there are nineteen functional subsystems, the central one of which is the system's decider. Since any algorithm is a recipe for a decision-making process, Miller unknowingly [cf. Mihram (1979)] had uncovered the preference for algorithmic, as opposed to mathematical, models among biologists, sociologists, and sociobiologists as well.

2 The Algorithm

Wheatley and Unwin (1972) made quite explicit what Mihram (1970) had suggested quite strongly: viz., that algorithmic modelling is distinctly different from models written in the language of mathematics:

> *An algorithm is a mathematical recipe. From this, its meaning has been extended to cover a recipe in any field of activity.*
>
> Wheatley/Unwin (1972)

This distinction between the algorithm and mathematics is, however, quite grammatical [cf. Mihram (1973)]: the algorithm is a second-person expression, or command, whereas a mathematical statement is expressed in the third person (e.g., F =mXa).

The pertinence of the distinction to biologists, however, lies in Miller's revelation (1978) that every living system, no matter how small or complex, contains as its central subsystem its decider:

> *the executive which receives information from all the other subsystems and transmits to them information outputs that control the entire organization.*
>
> Miller (1978)

Thus, if one is to capture the dynamics of any living system in terms of a computerised model, one would do well to employ

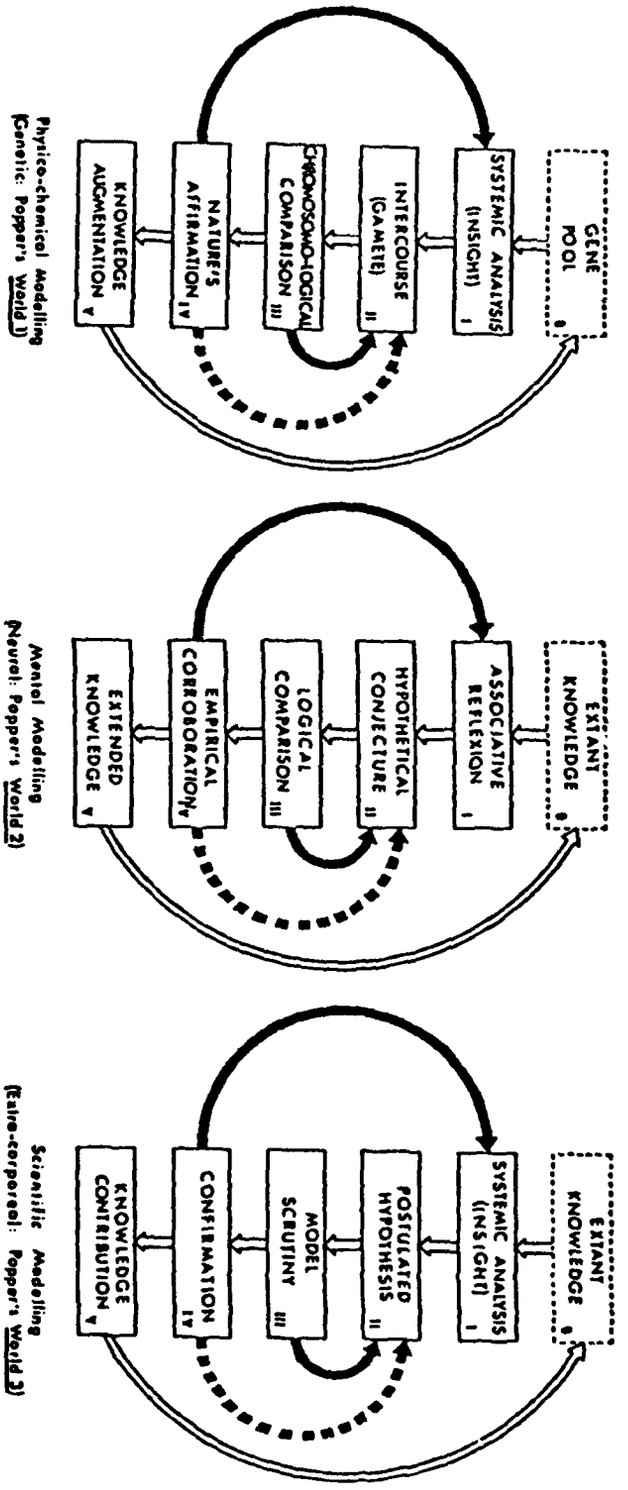


Figure 1. EVINCIBLE REASON: MAN'S SEARCH FOR TRUTH IS NATURE'S PROCESS FOR SURVIVAL.

the algorithmic (as opposed to mathematical) construction. The algorithm is ideally suited for capturing the dynamics of any living system because it can precisely describe the conditions under which a change is made, a decision or choice is enacted.

3 Exemplary Systems

Many researchers in AI take, nonetheless, the mathematical approach: e.g., researchers dealing with neural networks [cf., e.g., Newman's paper in these 1991 proceedings and Gagliano et al (1991)] express their models in mathematics, then use computer algorithms to exercise a particular solution to these mathematical relationships.

This is the same approach used by the authors (e.g., Forrester and the Meadows-es) of the once-highly-touted "world models": viz., describe the world's economic development in terms of differential, or difference, equations, then go solve (arithmetically evaluate) this 'system' of time-dependent equations on a computer [Mihram (1974a)]. Unfortunately, here the underlying algorithms mime the passage of time by: (a) computing, from the present status, the status at the next step of time; and (b) advance time by one unit; then, (c) using the same algorithms, re-compute the next status, ...

Unfortunately, such an approach fails to capture the quite erratic dynamics of any living system: one needs to write an algorithm which, like the particular living system which it describes, is activated not regularly but, rather, if and when required.

The algorithmic, as opposed to the mathematical, among computerised models is thus far better suited to capture with scientific credibility the dynamics of any living system (or, of any system containing at least one living component).

The researchers dealing with neural networks via their mathematical models typically are describing motor activities of the living system; however, artificial intelligence researchers, attempting to capture the decision-making capabilities of a living organism, are finding that the algorithm is much better suited to their task than is mathematics, notwithstanding the negativistic approach of writers like Winograd/Flores [cf. Mihram, 1989].

As a further example, consider the currently increasing use of bibliographic retrieval systems in major research libraries. These software packages, or computer programmes, are in actuality simulation models of a librarian-researcher team seeking pertinent literature citations on a specified logical combination of subjects. The models become, in effect, an AI model of a librarian or researcher at his/her task. They are not mathematical, but they do describe the reason why algorithmic models are much better suited for capturing the dynamics of any living system than is mathematics: the decisions are described precisely by algorithms, not by mathematical expressions.

4 Concluding Remarks

The history of science actually reveals that one need not use mathematics in order to qualify as a scientist. Newton may well have given mathematics an esteemed place among languages used by scientists, and the French philosophers/mathematicians/scientists of the early nineteenth century only enhanced this image [cf., e.g., Mihram, 1991] when they virtually 'institutionalized' the notion of scientific method as being no more than the theorem-proving mechanism of mathematicians.

Ampere and these other early nineteenth-century scientists were in actuality only serving to confirm the correctness of Newton's laws: they first accepted/assumed that Newton was correct, then assumed (like the geometry student in quest of the terminating 'QED') that matter is particulate in its character, and then by mathematical argumentation derived results (such as the inverse-square laws of electricity and magnetism).

However, scientists (and biologists, particularly) should recall the success (also in the nineteenth century) of Charles Darwin. His ORIGIN OF THE SPECIES, if it were not for the editorial insertion of the pagination sequence, contains virtually no mathematics. As importantly, they should heed the message of Nobel Laureate Konrad Lorenz:

> *The Fashionable Fallacy [Today] of Dispensing with Description [in Favour of Mathematics]" ...."I have never in my life published a book or a paper with either a table or a graph in it.*
>
> Lorenz, 1973

Scientists who convey their model of the reality which they have observed may choose a natural language (the first-person format: a la Darwin), the language of mathematics (the third-person format: a la Newton), or computer programming (the second-person format). The decision/choice must not be a mere predisposition, but, rather, a result of a reflexion [cf. Mihram and Mihram, 1984; Mihram, 1974b] on the intrinsic character of the natural phenomenon, or system of phenomena, being studied/observed. Are deciders to be mimed?

References

Amit, D.J. (1989), *World of Attractor Neural Networks*, Cambridge U. Press, London.

Gagliano, R. et al (1991), Pre- versus Post-Synaptic Long-term Potentiation in Neural Circuits, *Modeling & Simulation* **22**: to appear, 1991.

Kemeny, J.G. (1969), Mathematical & Computer Models of Large Systems, *Cybernetics & the Management of Large Systems*, E.M. Dewan, ed., Amer Soc Cyber., Washington,

DC, pp. 65-74.

Lorenz, Konrad Z. (1973), Fashionable Fallacy of Dispensing with Description, *Naturwissenschaften* **60**: 1-9.

McCulloch, W. and Pitts, W. (1943), A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bull. Math. Biophysics*, **5**,115-133.

Mihram, G. Arthur (1970), *Simulation:* Statistical *Foundations & Methodology*, Academic Press, Orlando, FL, 1972.

Mihram, G. Arthur (1973), Simulation: Methodology for Decision Theorists, *Role & Effectiveness of Theories of Decision in Practice*, D.J. White and K.C. Bowen, eds., Hodder/ Stoughton, London, pp. 320-327, 1975.

Mihram, G. Arthur (1974a), A Critique of World Models, *Proc, SocGen Systems Research*, University Park, Cal, 1975.

Mihram, G. Arthur (1974b), *An Epistle to Dr. Benjamin Franklin*, Exposition-University Press, Pompano Beach, FL, 1975.

Mihram, G. Arthur (1979), A Simulationist's Look at Miller's LIVING SYSTEMS, *General Systems Theory: Science, Methodology, Technology*, B.R. Gaines, ed., Soc Gen Syst Res, University Park, Calif, pp. 219-229.

Mihram, G. Arthur (1989), Artificial Intelligence Research Gone Astray: The Winograd/Flores Book, *Modeling & Simulation*, **20**, 549-555.

Mihram, G. Arthur (1991), Mathematics, Statistics, Computer Programming: Simulation Methodology in Historical Perspective, *Modeling & Simulation*, **22**, to appear, 1991.

Mihram, G. Arthur and Danielle Mihram (1984), Credibility: Every Computer Programme is a Simulation Model, *Proc 18th Ann Hawaii Int'l Conf on System Sciences*, Honolulu, 306-316,1985.

Miller, J.G. (1978), *Living Systems*, McGraw-Hill, N.Y.

Sayre, K.M. and Crosson, F.J., eds. (1963), *The Modeling of Mind*, Simon & Schuster, N.Y.

Wheatley, D.M. and Unwin, A.W. (172), *Algorithm User's Guide*, Longman, London.

Checking the Validity of the Bootstrap Analysis by Bootstrap

Hung Chen*
Department of Applied Mathematics and Statistics
State University of New York, Stony Brook, NY11794

Hung Kung Liu
Statistical Engineering Division
National Institute of Standards and Technology, Gaithersburg, MD 20899

Abstract

We describe a method for using pseudo realizations of data to check the validity of the simple bootstrap analysis considered in Efron (1979). A simulation study is performed to demonstrate the usefulness of the proposed method.

1 INTRODUCTION

Simple bootstrap analysis, as described in Efron (1979), gives nonparametric estimates of accuracy of statistic of interest. Major advantages of a bootstrap analysis are its simplicity and its use at the case when the analytic method is intractable. However, it is not clear in general whether a bootstrap analysis is valid. Refer to Bickel and Freedman (1981) for examples that the bootstrap analysis fails. In view of this, an algorithm is proposed in this article for using pseudo realization of data to check its validity. The specifics of this algorithm is given in Section 2.

Let us start with a brief review of the one-sample simple bootstrap analysis. Suppose the quantity of interest is $\theta(F)$, which is a parameter of unknown distribution F. Let $s(\mathbf{x}_n)$ be an estimate of $\theta(F)$ based on $\mathbf{x}_n$, where $\mathbf{x}_n = (x_1, \ldots, x_n)$ denotes a realization of random sample $\mathbf{X}_n = (X_1, \ldots, X_n)$ from F. We then need to assess the accuracy of $s(\mathbf{X}_n)$ as an estimator of $\theta(F)$. In this article, the measure of accuracy, ϕ, will always be referred to as the kth percentile of the distribution of $\sqrt{n}[s(\mathbf{X}_n) - \theta(F)]$ when the distribution of $\sqrt{n}[s(\mathbf{X}_n) - \theta(F)]$ is nondegenerate. However, the proposed algorithm is also applicable to other measure of errors, such as standard error.

In this case, the bootstrap estimate of ϕ is the corresponding kth percentile of $\sqrt{n}[s(\mathbf{X}_n^*) - \theta(F_n)]$. Here F_n is the usual empirical distribution function based on $\mathbf{X}_n$ and $s(\mathbf{X}_n^*)$ is the corresponding estimate based on the bootstrap sample $\mathbf{X}_n^* = (X_1^*, \ldots, X_n^*)$, which is a random sample of size n from F_n.

The proposed algorithm is motivated by the following argument. Suppose that we have two observed samples of size n from F. Denote them by $\mathbf{x}_{n1}$ and $\mathbf{x}_{n2}$, respectively. Although ϕ is unknown, it is a fixed number. When a bootstrap analysis is a valid one, the two bootstrap estimates of ϕ based on $\mathbf{x}_{n1}$ and $\mathbf{x}_{n2}$, respectively, should not be too different. In other words, the variability of bootstrap estimate of ϕ over realizations of $\mathbf{X}_n$ should be "small" compared to ϕ when the bootstrap analysis is valid.

In summary, a bootstrap analysis is not valid if the bootstrap estimate of ϕ varies "dramatically" over realizations of $\mathbf{X}_n$. Therefore, the accuracy or the sensitivity of bootstrap estimate of ϕ over realizations of $\mathbf{X}_n$ should be analyzed before reporting the bootstrap statistics.

However, a major hurdle in observing the variability of bootstrap statistics over $\mathbf{X}_n$ is that the statistician has available only one realization of $\mathbf{X}_n$, $\mathbf{x}_n$. Hence we propose to generate "pseudo" realizations of $\mathbf{X}_n$ based on a smoothed estimate of F to get an estimate of the variability of the bootstrap estimate of ϕ. This algorithm can be called smooth bootstrap-after-bootstrap according to Efron (1990b). This idea is, strictly speaking, not new. It is just another application of the bootstrap. This

*Research supported by the National Science Foundation under Grant No. DMS-8901556.

problem is also considered in Efron (1990b). It suggestes to use the jackknife method to estimate the variability of the bootstrap estimate of ϕ. This leads to the so-called jackknife-after-bootstrap method.

2 Proposed Algorithm

According to the discussions in Section 1, the following algorithm is proposed to assess the "accuracy" of the simple bootstrap analysis. This algorithm proceeds in three steps:

Step 1. Construct a smoothed estimate of F, F_{sn}. (See Section 3 for discussions on the construction of F_{sn}.)

Step 2. Draw B random sample of size n from F_{sn}, say for $1 \leq b \leq B$

$$X_{sbi} = x_{sbi}, X_{sbi} \sim_{ind} F_{sn} \quad i = 1, \ldots, n.$$

Call these the test bootstrap samples, $\mathbf{X}_{sb} = (X_{sb1}, \ldots, X_{sbn})$, and $\mathbf{x}_{sb} = (x_{sb1}, \ldots, x_{sbn})$.

Step 3. For each test bootstrap sample $\mathbf{x}_{sb}$, find its bootstrap estimate of ϕ, and then study the variability among those B bootstrap estimates of ϕ.

When a "significant" variation among the B bootstrap estimates of ϕ is found, it indicates that the result of bootstrap analysis is dubious. Let F_{nb} be the empirical distribution function based on $\mathbf{x}_{sb}$. Since F_{nb} lies in a neighborhood of F_{sn}, the found "significant" variation means the lack of uniformity over the above mentioned neighborhood. Hence, we may cast doubt on the usefulness of bootstrap analysis since F_{sn} lies in a small neighborhood of F. No specific recipe on measuring variation is given in this article. See Sections 3 and 4 for further discussions in this regard.

If F_{sn} is replaced by F_n at Step 1 of the proposed algorithm, the implementation of the proposed algorithm is almost identical to the implementation of nested double bootstrap algorithm in Beran (1987) and others. However, these two algorithms are proposed with totally different rationale. The nested double bootstrap is proposed to improve the bootstrap estimate of ϕ when the bootstrap works, but the proposed algorithm is used to estimate the variability of the bootstrap estimate of ϕ.

3 Discussion

An algorithm is proposed for evaluating the variability of a bootstrap analysis over realizations of $\mathbf{X}_n$. A practical disadvantage of this algorithm is that it is computationally expensive. As a rough guide, the execution time of the proposed algorithm is roughly equal to the execution time of evaluating the bootstrap method by a Monte Carlo experiment. Through various reports in the literature and the greater availability of fast computer, the computational cost should not be a big problem in today's computing environment.

In the implementation of proposed algorithm, four issues are needed to be addressed. Namely,

1. the prescription of F_{sn} (in Step 1),
2. the choice of B (in Step 2),
3. the computation of bootstrap statistics (in Step 3), and
4. the variability of bootstrap statistics (in Step 3).

For the first issue, there are various methods to construct a smooth probability density in the density estimation literature. The smooth probability distribution F_{sn} can be then obtained by an appropriate integration. Two natural questions are then raised. They are *smooth distribution function versus empirical distribution function* and *the choice of smoothing scheme*. For the first question, refer to Hall, DiCiccio, and Romano (1989) and references therein. As a remark, the use of smooth probability distribution may not be appropriate if X is a discrete random varible. For the second question, we are investigating the smoothing scheme based on the logspline density estimate in Stone and Koo (1986). The result will be reported elsewhere. An advantage of logspline density estimate over other smoothing schemes is that most widely used density functions are of the form log-spline.

For the second issue, we suggest to let B be around $n(n-1)/2$ based on the following reason. When the jackknife-after-bootstrap method is used, the accuracy measure of bootstrap statistics is obtained by repeatedly deleting a single observation. On the other hand, $\mathbf{x}_{sb}$ may contain any number of x_i with different probabilities among those B pseudo realizations of $\mathbf{x}$ if F_n is in place of F_{sn} in the proposed algorithm. Furthermore, Theorem 6.1 of Efron (1982) attempts to view the jackknife as a linear approximation to the bootstrap. As it is known, the jackkbife method may have trouble for markedly nonlinear statistics. To avoid the proposed algorithm to be reduced to the jackknife-after-bootstrap method, we would like to choose B large enough to guarantee that these pseudo realizations should include some of the "delete-many" samples.

For the third issue, it is known that one cannot usually compute analytically the bootstrap statistics except in special cases or in small samples. A viable alternative is to approximate the bootstrap distribution numerically by means of a Monte Carlo sampling. When this method is used, the number of Monte Carlo sampling will be constrained by the available computing power. It then raises the question on how many bootstrap replications must be taken to insure that the observed variability at Step 3 does not come from the randomness added by the Monte Carlo sampling. Note that a bootstrap sample is the same as a random sample of size n drawn with replacement from the actual sample $\mathbf{x}_n$. For simple statistics, some obvious estimate on the variability of Monte Carlo sampling can be obtained based on Berry-Esséen type bound. It turns out that a large number of replications may be needed. However, Efron (1990a) suggested that between 1000 and 2000 replications are to be appropriate. Currently, we are investigating this error bound along the line of Efron (1990b).

For the last issue, the variability can be revealed by various available exploratory data analysis tools such as the one used in Section 4. Formal tests similar to these proposed in Nair (1982) can also be used.

4 Simulation Study

A Monte Carlo study was performed to demonstrate the usefulness of the proposed algorithm in Section 2. For simplicity, we consider the estimation of population mean by sample mean based on 50 observations. The two cases considered for F are Normal and Cauchy, and the measure of accuracy is various percentiles. The specific percentiles considered here are 5th, 10th, 16th, and 32nd. Since the density function of Normal is of the form log-spline but the density function of Cauchy is not, we replace F_{sn} in Step 1 by F_n in this Monte Carlo study to avoid a possible bias toward the proposed algorithm. Therefore, the algorithm used here is the bootstrap-after-bootstrap method instead of the smooth bootstrap-after-bootstrap method as described in Section 2. Also, a Monte Carlo algorithm is used in Step 3 to find the bootstrap estimate of ϕ.

For each realization of 50 observations, B (in Step 2) is set to be 1000 and the number of bootstrap replication (in Step 3) is 2000. Results are then summarized in Figures 2 and 4 for the first realization. In these figures, the

Figure 1: Normal, Bootstrap

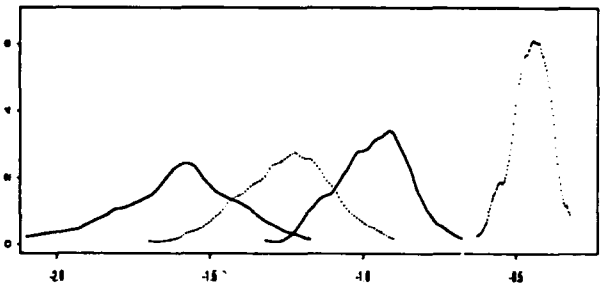


Figure 2: Normal, Bootstrap-After-Bootstrap

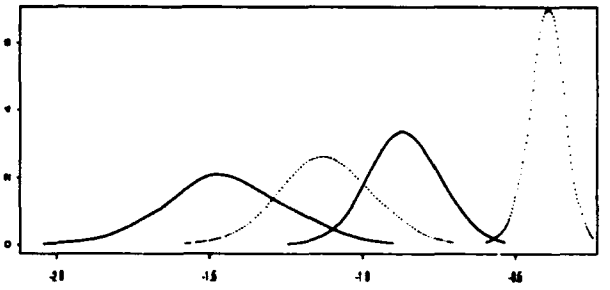


Figure 3: Cauchy, Bootstrap

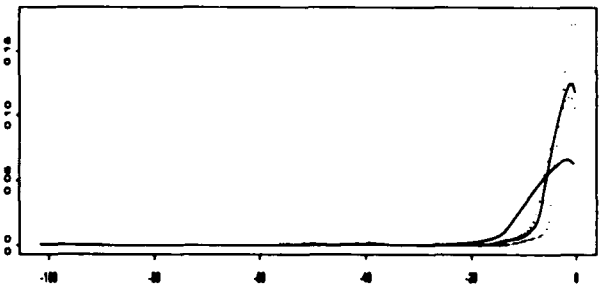
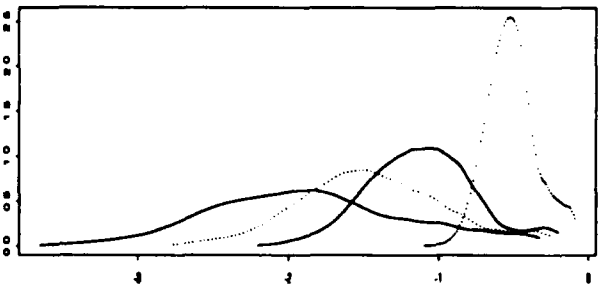


Figure 4: Cauchy, Bootstrap-After-Bootstrap



plotted curves from left to right are the estimated density functions for those four percentiles arranged in ascending order. Each curve is obtained by applying the kernel smoother over 1000 bootstrap percentiles. For kernel smoother, a triangular kernel is used and the bandwidth is set to be one-quarter of the sample range of these 1000 bootstrap percentiles. However, Figure 4 is constructed without the normalizing factor $\sqrt{50}$. This experiment is then repeated for another 99 times. The characteristics of all figures from the next 99 realizations are similar to Figures 2 and 4 correspondingly. Here the characteristics refers to the amount of overlapping among these four density functions and the shape of the density functions. However, the "center" of these curves does vary. For example, the median of the estimated 5th percentile density function over these 100 experiments ranges over $[-1.119, -2.078]$ when F is Normal.

In order to check whether the estimate of the variability of the bootstrap estimate of ϕ obtained from bootstrap-after-bootstrap is close to the variability of the bootstrap estimate of ϕ, we compute bootstrap statistics based on 2000 replications for 100 realizations of 50 observations. Figures 1 and 3 summarize the result from those 100 realizations. They are constructed in the same fashion of as Figures 2 and 4. Figure 3 shows clearly that the four estimated density functions have a concentration around $[-10, 0]$ and spread over a wide range of values. These just reflect the fact that there are a few wild outliers presented in most realizations of 50 observations.

Based on Bickel and Freedman (1981) and Knight (1989), the bootstrap analysis is useful for Normal but is not good for Cauchy. Figures 2 and 4 confirm it. The dissimilarity between Figure 3 and Figure 4 suggests that the estimate of the variability of the bootstrap estimate of ϕ obtained from bootstrap-after-bootstrap is not necessary equal to the variability of the bootstrap estimate of ϕ.

In summary, the proposed algorithm has the potential of revealing whether a bootstrap analysis is a valid one. But the proposed estimate of the measure of accuracy is not necessary close to the unknown measure of accuracy, ϕ. This again confirms that the bootstrap analysis may fail sometimes although it is a quite useful method.

Reference

Beran, R. (1987). Prepivoting to reduce level error of confidence sets. *Biometrika* **74**, 457-468.

Bickel, P.J. and Freedman, D.A. (1981). Some asymptotic theory for the bootstrap. *Ann. Statist.* **9**, 1196-1217.

Efron, B. (1979). Bootstrap methods: another look at the jackknife *Ann. Statist.* **7** 1-26.

Efron, B. (1982). *The Jackknife, The Bootstrap, and Other Resampling Schemes.* SIAM. CBMS-*Natl. Sci. Found. Monogr.* **38** Philadelphia.

Efron, B. (1990a). More efficient bootstrap computations. *J. Amer. Statist. Assoc.* **85** 79-89.

Efron, B. (1990b). Jackknife-After-Bootstrap Standard Errors and Influence Functions. Technical Report #339, Department of Statistics, Stanford University.

Hall, P., DiCiccio, T.J. and Romano, J.P. (1989). On smoothing and the bootstrap. *Ann. Statist.* **17**, 692-704.

Knight, K. (1989). On bootstrap of the sample mean in the infinite variance case. *Ann. Statist.* **17** 1168-1175.

Nair, V.J. (1982). Q-Q plots with confidence bands for comparing several populations. *Scand. J. Statist.* **9**, 193-200.

Stone, C.J. and Koo C.-Y. (1986). Logspline density estimation. In *AMS Contemporary Math. Ser.* **29** 1-15. Amer. Math. Soc., Providence.

QUASI-RANDOM RESAMPLING FOR THE BOOTSTRAP

Kim-Anh Do
Statistical Sciences Division
Centre for Mathematics and Its Applications
Australian National University
Canberra, A.C.T. 2601
Australia

Abstract

Quasi-random sequences are known to give efficient numerical integration rules in many Bayesian statistical problems where the posterior distribution can be transformed into periodic functions on the n-dimensional hypercube. From this idea we develop a quasi-random approach to the generation of resamples used for Monte Carlo approximations to bootstrap estimates of bias, variance and distribution functions. We demonstrate a major difference between quasi-random bootstrap resamples, which are generated by deterministic algorithms and have no true randomness, and the usual pseudo-random bootstrap resamples generated by the classical bootstrap approach. Various quasi-random approaches are considered and are shown via a simulation study to result in approximants that are competitive in terms of efficiency when compared with other bootstrap Monte Carlo procedures such as balanced and antithetic resampling.

1. Introduction

Let $\mathcal{X} = \{X_1, \ldots, X_n\}$ denote a random sample of size n, write $\hat{T}$ for a function of these data, and let $\hat{T}^*$ represent the same function of the data in a sample $\mathcal{X}^* = \{X_1^*, \ldots, X_n^*\}$ drawn randomly from $\mathcal{X}$, with replacement. Thus, $\mathcal{X}^*$ is a uniform resample. The bootstrap estimate of $t = E(\hat{T})$ is $\hat{t} = E(\hat{T}^* | \mathcal{X})$. In the event that the X_i's are vectors, assume that we can write $T = g(\bar{X})$ for a smooth function g, where $\bar{X}$ denotes the mean of $\mathcal{X}$. Let bracketed superscripts denote indices of vector elements, and put $g_j(x) = \partial g(x)/\partial x^{(j)}$, $G(X_i) = \Sigma_j X_i^{(j)} g_j(\bar{X})$. We begin by describing an algorithm for constructing a quasi-Monte Carlo approximation to $\hat{t}$.

First, sort the n data values in $\mathcal{X}$, obtaining $\mathcal{X} = \{X_{(1)}, \ldots, X_{(n)}\}$ where $G(X_{(1)}) \le \ldots \le G(X_{(n)})$. (Alternatively, we could ask that $G(X_{(1)}) \ge \ldots \ge G(X_{(n)})$. If the sample $\mathcal{X}$ is univariate then we may order the sample values directly, and not pass to the function $G(X_{(i)})$.) Let B denote the number of bootstrap resamples and let $\mathbf{u}_b = (u_b^{(1)}, \ldots, u_b^{(n)})$, $1 \le b \le B$, represent B points in the n-dimensional hypercube $C_n = [0, 1)^n$ generated by a quasi-random algorithm, which we shall describe in section 2. Transform the $\mathbf{u}_b$'s into a set of index vectors $\mathbf{i}_b = (i_b^{(1)}, \ldots, i_b^{(n)})$, $1 \le b \le B$, by

$$i_b^{(j)} = [1 + n u_b^{(j)}] \quad b = 1, \ldots, B; \quad j = 1, \ldots, n,$$

where $[x]$ denote the largest integer not exceeding x. Then each $i_b^{(j)}$ is an integer between 1 and n. Conditional on $\mathcal{X}$, let $\mathcal{X}_1^\dagger, \ldots, \mathcal{X}_B^\dagger$ denote quasi-random resamples defined by

$$\mathcal{X}_b^\dagger = \{X_{(i_b^{(1)})}, \ldots, X_{(i_b^{(n)})}\}, \quad b = 1, \ldots, B.$$

Thus by using the idea of selecting points according to a deterministic scheme that is well-suited for numerical integration, we develop a quasi-random approach to bootstrap resampling. Our contributions are twofold: (i) we expand the scope of usefulness of quasi-random methods to other computer-intensive areas, in particular we familiarize "bootstrappers with this school of thought; (ii) we explore possible efficiency gains (over pseudo-random resampling) in using different types of quasi-random resampling.

2. Quasi-random sequences

The terminology given here is not always standard but has been found to be the easiest for distinguishing the nature of quasi-random sequences. We shall consider *regular* quasi-random sequences generated by

$$\mathbf{u}_{b+1} = \mathbf{u}_b + \boldsymbol{\alpha} (\mathrm{mod} \mathbf{1}), \tag{1}$$

where $\mathbf{u}_1$ is a fixed or random point in the n-dimensional hypercube. Note that the jth coordinate in $\mathbf{u}_b$ is the fractional part of the jth coordinate in $\mathbf{u}_b + \boldsymbol{\alpha}$. Regular sequences are distinguished as *rational* or *irrational* according as $\boldsymbol{\alpha}$ is a vector consisting of only rationals or only irrationals. We also consider *irregular* or *quasi-random* sequences generated by other forms of algorithm and include pseudo-random sequences. Assessment of how "good" a deterministic sequence is can often be expressed in terms of its discrepancy. The discrepancy measure provides a bound to the integration (i.e. expectation) error in numerical integration, provided the function to be integrated is of bounded variation. In the bootstrap framework,

approximation of bias, variance, and distribution functions are generally based on well-behaved functions. Therefore, it is anticipated that low-discrepancy sequences for integration rules will provide bootstrap approximants with high accuracy.

To construct low-discrepancy sequences, Hlawka (1962) used the so-called *method of good lattice points* which takes account of the regularity of the function f, in addition to the bounded variation property of f. Hlawka considered the case of B being prime and $\mathbf{k}$ a point with integral coordinates. Let $R(\mathbf{k})$ be the set of all non-zero vectors $\mathbf{h}$ such that $\mathbf{k} \cdot \mathbf{h} \equiv 0 (\text{mod} B)$. Define

$$r(\mathbf{h}) = \prod_{j=1}^{n} \max(1, |h_j|); \quad \rho(\mathbf{k}) = \min_{\mathbf{h} \in R(\mathbf{k})} r(\mathbf{h}).$$

Hlawka called the lattice point $\mathbf{k}$ *good modulo* B if

$$\rho(\mathbf{k}) \geq B(8 \log B)^{-2},$$

and proved the existence of good lattice points modulo any prime B. Zaremba (1966) showed that in the case $n = 2$, even "better" lattice points corresponding to a larger $\rho(\mathbf{k})$ can be obtained where B does not need to be prime. Niederreiter (1977) improved on Zaremba's result for an arbitrary dimension n. Shaw (1988) considered another measure of distance defined as

$$\nu(\mathbf{k}) = \min_{\mathbf{h} \in R(\mathbf{k})} \sum_{j=1}^{n} |h_j|,$$

where ν is an upper bound to the minimum number of parallel $(n-1)$-dimensional hyperplanes covering the sequence $\mathbf{u}_1, \ldots, \mathbf{u}_B$. We shall discuss below the construction of several different types of quasi-random sequences and their properties.

• **Rational sequences**

SEQUENCE 1. Here the $\mathbf{u}_b$'s are as defined in (1), where

$$\boldsymbol{\alpha} = \left(B^{-1}, B^{-1}k, B^{-1}k^2 \text{mod} B, \ldots, B^{-1}k^{n-1} \text{mod} B\right).$$

The construction of this sequence is based on the method of good lattice points. When $n = 2$, it is possible to explicitly construct good lattice points by using continued fractions.

• **Irrational sequences.** These can be generated using a method closely related to that of rational sequences. Here the $\mathbf{u}_b$'s are as defined in (1) where $\boldsymbol{\alpha}$ is an irrational point of the form $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ and $1, \alpha_1, \ldots, \alpha_n$ are linearly independent over the rationals. Davis (1963, pp.356-457) proved the equidistribution property for these sequences. Let $p_1, p_2, \ldots$ be the sequence of prime numbers $1, 3, 5, 7, 11, \ldots$ and p be some prime. A number x is said to have order $y \text{mod} z$ if $x^y \equiv 1 \text{mod} z$ and $x^k \not\equiv 1$ for $1 \leq k < y$.

In our bootstrap simulation study, we consider equidistributed irrational sequences by using $\boldsymbol{\alpha}$ as described below.

SEQUENCE 2

$$\boldsymbol{\alpha} = \left(\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_n}\right),$$

SEQUENCE 3

$$\boldsymbol{\alpha} = \left(\xi, \xi^2, \ldots, \xi^n\right) \text{ where } \xi = p^{n+1},$$

SEQUENCE 4

$$\boldsymbol{\alpha} = \left(2\cos\frac{2\pi}{p}, 2\cos\frac{4\pi}{p}, \ldots, 2\cos\frac{2\pi n}{p}\right),$$

where $p \geq 2n+3$ and satisfies either (i) 2 has order $p-1 \text{mod} p$ or (ii) 2 has order $(p-1)/2 \text{mod} p$ and $p = 7 \text{mod} 8$.

• **Irregular sequences.** We focus attention specifically on three irregular sequences that have been used successfully in integration problems.

SEQUENCE 5. (Haber sequence)

$$\mathbf{u}_b = \left(\frac{b(b+1)}{2}\sqrt{p_1}, \ldots, \frac{b(b+1)}{2}\sqrt{p_n}\right) \ (\text{mod} 1).$$

SEQUENCE 6. (Hammersley sequence)

$$\mathbf{u}_b = \left(B^{-1}b, \phi_{p_1}(b), \ldots, \phi_{p_{n-1}}(b)\right),$$

where $p_1, \ldots, p_n$ are the first $n-1$ primes and $\phi_p(b)$ is the *radical inverse function* of b to the base p (a rigorous definition is given below).

SEQUENCE 7. (Halton sequence)

$$\mathbf{u}_b = \left(\phi_{p_1}(b), \phi_{p_2}(b), \ldots, \phi_{p_n}(b)\right).$$

The function $\phi_p(b)$ is the *rational inverse function* of b to the base p, obtained by taking the p-ary representation of the number b and reflecting the digits about the decimal point.

3. Simulation Study

In this section we summarize the results of a simulation study of the performance of quasi-random resampling relative to uniform resampling. We applied our method to the problems of estimating bias and variance when $T(\mathcal{X}) = \bar{X}^2$ or $T(\mathbf{X}) = \sqrt{|\bar{X}|}$, and of estimating the distribution of the Studentized mean. Let T be the numerical value of the statistic of interest calculated from the original sample, and let T_b^* be the corresponding value calculated from the bth bootstrap resample. Bias, variance and the distribution function $\hat{F}(x) = P(T^* \leq x | \mathcal{X})$ can be estimated by

$$\begin{aligned}
\widehat{\text{bias}} &= T^* - T, \\
\widehat{\text{var}} &= B^{-1} \sum_{b=1}^{B} \left(T_b^* - T_b^*\right)^2, \\
\hat{F}(x) &= B^{-1} \sum_{b=1}^{B} I\left(T_b^* \leq x\right),
\end{aligned}$$

where $T^* = B^{-1}\Sigma_b T_b^*$.

Consider the problem of estimating $\hat{F}(x)$. We calculated $\hat{F}(x)$ using 100,000 uniform resamples. Let $F_{t\cdot}(x)$ and $F_Q(x)$

denote our approximations to $\hat{F}(x)$ using B uniform resamples and B quasi-random resamples respectively. We computed $D_Q = \{F_Q(x) - \hat{F}(x)\}^2$ and computed the average, D_U, of $\{F_U(x) - \hat{F}(x)\}^2$ over $M = 100$ independent repeats of the uniform resampling scheme, for a given sample. Note that we do not need to average D_Q over M repeats since there is only one deterministic quasi-random sequence for each sample. We then averaged D_Q and D_U over $N = 250$ independent samples, obtaining d_Q and d_U, say; and finally, took the ratio $r = d_U/d_Q$. This gave a measure of the efficiency of quasi-random resampling relative to uniform resampling in estimation of distribution functions. The case of bias and variance estimation can be treated similarly, with obvious analogues for d_U and d_Q. It was observed that quasi-random resampling does not perform better or worse than quasi-random resampling in the problem of bias estimation. Therefore the tables presented in this paper will concentrate only on efficiencies in variance and distribution estimation.

In the problem of distribution estimation, we have restricted our considerations to rational sequences only. Rational sequences perform better than straight random sequences at all quantile values. They exhibit the common pattern of better performance towards the centre of the distribution. However, efficiency gains at the tails are still impressive and surpass those obtained from balanced and antithetic resampling, especially when the parent population generating $\mathcal{X}$ is exponential.

4. Conclusions

Regular sequences are no more difficult to implement than pseudo-random sequences and usually exhibit consistent trends in efficiency gains. Bootstrap resampling based on Haber sequences is rather disappointing due to their erratic behaviour, but quasi-random resampling based on radical inverse functions such as the Hammersley and Halton sequences can yield significant efficiency gains for large B. The behaviours observed here for irregular sequences are in close agreement with results in Shaw (1988) and Warnock (1972), who concentrated on efficient numerical integration rules. It should be emphasised that the problems of variance and distribution estimation are usually of more practical importance than bias estimation, since bias is generally small relative to standard deviation. Therefore, even though quasi-random resampling does not provide an improvement over pseudo-random resampling in problems of bias estimation, quasi-random sequences remain attractive in the bootstrap context because of their superior performance in variance and distribution estimation. We suggest that regular sequences be applied quite generally in bootstrap resampling problems, although greater caution is recommended for irregular sequences. A more rigorous and detailed version of this paper is available from the author.

References

Davis, P.J. (1963). *Interpolation and Approximation*. Ginn (Blaisdell), Boston, Massachusetts.

Davis, P.J. and Rabinowitz, P. (1984). *Methods of Numerical Integration*. 2nd ed. Academic, Orlando, Fla.

Haber, S. (1966). A modified Monte Carlo quadrature. *Math. Comp.* **20**, 361–368.

Halton, J.H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.* **2**, 84–90.

Hammersley, J.M. (1960). Monte Carlo methods for solving multivariate problems. *Ann. New York Acad. Sci.* **86**, 844–874.

Hlawka, E. (1962). Zur angenäherten Berechnung mehrfacher Integrale, *Monatsh. Math.* **66**, 140–151.

Niederreiter, H. (1977). Pseudo-random numbers and optimal coefficients. *Adv. Math.* **26**, 99-181.

Niederreiter, H. (1978). Quasi-Monte Carlo methods and pseudo-random numbers. *Bull. Amer. Math. Soc.* **84**, 957–1042.

Shaw, J.E.H. (1988). A quasi-random approach to integration in Bayesian statistics. *Ann. Statist.* **16**, 895–914.

Warnock, J.T. (1972). Computational investigations of low-discrepancy point sets. *Applications of Number Theory to Numerical Analysis* (S.K. Zaremba, ed.), Academic Press, New York, pp.319–343.

Zaremba, S.K. (1966). Good lattice points, discrepancy, and numerical integration. *Ann. Mat. Pura Appl.* **73**, 293–317.

Table 1
Efficiencies for variance estimation using rational sequences

n	B	k	Distribution	$T(\mathcal{X}) = \mathcal{X}^2$	$T(\mathcal{X}) = \sqrt{\lvert\mathcal{X}\rvert}$
10	237	10	Normal N(1,1)	2.25	1.81
			Exponential	2.35	2.17
			Folded Normal	2.35	2.36
10	342	17	Normal N(1,1)	2.15	1.75
			Exponential	2.23	2.06
			Folded Normal	2.23	2.76
10	610	23	Normal N(1,1)	2.15	1.87
			Exponential	2.24	2.19
			Folded Normal	2.36	2.29

Table 2
Efficiencies for variance estimation using irrational sequences

n	B	Distribution	Seq. 2	Seq. 3	Seq. 4
10	100	Normal N(1,1)	1.34	1.33	1.03
		Exponential	1.55	1.37	1.05
		Folded Normal	1.62	1.39	1.08
10	200	Normal N(1,1)	1.39	1.38	1.00
		Exponential	1.75	1.41	1.02
		Folded Normal	1.79	1.44	1.09
10	300	Normal N(1,1)	1.89	1.61	1.11
		Exponential	1.93	1.69	1.24
		Folded Normal	1.97	1.75	1.26
10	500	Normal N(1,1)	1.96	1.88	1.39
		Exponential	2.30	2.00	1.38
		Folded Normal	2.41	1.97	1.42

Table 3
Efficiencies for variance estimation using Haber sequences

n	B	Distribution	$T(\mathcal{X}) = X^2$	$T(\mathcal{X}) = \sqrt{\lvert X \rvert}$
10	100	Normal N(1,1)	10.31	2.20
		Exponential	4.43	5.80
		Folded Normal	7.67	5.19
10	200	Normal N(1,1)	4.46	1.37
		Exponential	1.91	5.74
		Folded Normal	2.84	4.25
10	300	Normal N(1,1)	3.22	1.82
		Exponential	2.87	14.86
		Folded Normal	3.9	6.58

Table 4
Efficiencies for variance estimation using Hammersley (S6) and Halton (S7) sequences

			$T(\mathcal{X}) = X^2$		$T(\mathcal{X}) = \sqrt{\lvert X \rvert}$	
n	B	Distribution	S6	S7	S6	S7
10	500	Normal N(1,1)	2.37	0.46	1.87	1.89
		Exponential	0.89	0.21	8.86	6.75
		Folded Normal	1.54	0.35	5.61	6.03
10	1000	Normal N(1,1)	3.16	0.53	2.13	2.31
		Exponential	1.11	0.33	2.25	3.79
		Folded Normal	2.41	0.51	2.38	4.59
10	2000	Normal N(1,1)	5.99	0.65	3.78	4.01
		Exponential	2.19	0.39	3.15	4.50
		Folded Normal	2.60	0.47	3.00	4.71

Table 5
Efficiencies for distribution estimations using rational sequences

				α:	0.90	0.95	0.975
n	B	k	Distribution	z_α:	1.282	1.645	1.96
10	237	10	Normal N(1,1)		2.55	2.23	1.88
			Exponential		2.57	2.34	2.23
			Folded Normal		2.59	2.23	1.53
10	342	17	Normal N(1,1)		2.51	2.10	1.97
			Exponential		2.47	2.35	2.15
			Folded Normal		2.49	2.06	2.01
10	237	10	Normal N(1,1)		2.42	2.06	1.74
			Exponential		2.41	2.25	2.15
			Folded Normal		2.40	2.07	1.95

Table 6
Efficiencies for bias, variance and distribution estimations using rational sequences in comparison to balanced and antithetic resampling

Resampling method	$T(\mathcal{X}) = \sqrt{\lvert \bar{X} \rvert}$		$T(\mathcal{X}) = \sqrt{n}(\bar{X}^* - \bar{X})/s^*$		
			α:	0.90	0.95	0.975
	Bias	Var	z_α:	1.282	1.645	1.96
Quasi-random using rational sequence $(n, B, k) = (10, 237, 10)$	1.00	2.17		2.57	2.34	2.23
Balanced $(n, B) = (10, 500)$	1.35	0.70		1.36	1.11	1.04
Antithetic $(n, B) = (10, 500)$	2.31	1.12		1.23	1.06	1.00

Bootstrapping with Constraints: Analysis of Scattering Asymmetry for Polarized Beam Studies

Kevin J. Coakley
National Institute of Standards and Technology
Statistical Engineering Division
Gaithersburg, MD 20899

1 Abstract

In polarized beam studies, an asymmetry statistic of physical interest is an estimate of the ratio of the difference and the sum of the Poisson rate parameters for two scattering processes. Typically, an additive background signal contributes to measurements of each scattering process. Background is measured in a third experiment. Data is corrected by subtracting measured background. When the measured background is larger than one of the other measurements, the asymmetry computed from the background corrected data is nonsensical. For such cases, true asymmetry and an associated conservative interval are estimated using a bootstrap procedure. Bootstrap replications of the observed data satisfy a constraint that insures physically meaningful results.

2 Introduction

In many areas of research, asymmetry statistics are of physical interest. For example, in atomic collision physics, asymmetry statistics computed from the scattering of spin-polarized electrons from atoms carry information about atomic structure (McClelland, et. al. 1989). In materials science studies, maps of magnetic microstructure are made based on the polarization of secondary electrons emitted from the material after it is bombarded by an energetic beam of electrons (Scheinfein, et. al. 1990). To estimate these polarizations, asymmetry statistics are computed.

Many of the experiments in which asymmetries are of interest involve the counting of electrons or other particles. Generally, streams of pulses (assumed to be Poisson distributed) are counted in two experiments, one for each orientation of the spins in the system. The number of counts measured in each experiment is associated with an intensity for each of the two spin orientations. The asymmetry is estimated by taking the ratio of the difference and background corrected sum of the two intensities.

Suppose that the number of scattering events for two different spin orientations are measured in two independent experiments. Further, assume that each experiment lasts the same amount of time t. This assumption can be relaxed with out loss of generality. The first observation N_1 can be expressed as the sum of two unobservable quantities as follows.

$$N_1 = N_1^* + N_{BG,1}^* \tag{1}$$

Above, N_1^* represents what would have been observed if there had been no background. The number of counts due to the background is $N_{BG,1}^*$. The terms on the right hand side of Eq. 1 are realizations of Poisson processes with parameters $\lambda_1 t$ and $\lambda_{BG} t$. The second measurement is expressed as

$$N_2 = N_2^* + N_{BG,2}^* \tag{2}$$

where the two terms on the right side of Eq. 2 are independent realizations of Poisson processes with parameters $\lambda_2 t$ and $\lambda_{BG} t$. The goal is to estimate the asymmetry term

$$R = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}. \tag{3}$$

Note that since true asymmetry R lies between -1 and +1, so should any estimate of asymmetry as well as the endpoints of any confidence interval for asymmetry.

In order to estimate the asymmetry, experimenters measure background in a third independent experiment. Suppose that this experiment also lasts time t. Further, assume that the experimental conditions for the background measurement are the same as for the other experiments. The number of detected background counts $N_{BG,3}$ is modeled as a realization of a Poisson process with parameter $\lambda_{BG} t$. With this third measurement, experimenters typically estimate asymmetry as

$$\hat{R} = \frac{N_1 - N_2}{N_1 + N_2 - 2\,N_{BG,3}}. \tag{4}$$

As the duration of the experiment increases, $\hat{R}$ converges to R. However, for short experiments, $\hat{R}$ may be far from R. Moreover, for short enough experiments, the measured background can be greater than one of the other two signals and the asymmetry computed from the data is not between -1 and 1. That is, the above estimate is outside the physically meaningful range. For such cases, it would seem as though the experiment was a failure. Here, useful information is extracted from such data using the bootstrap.

3 Bootstrap Approach

Using a parametric bootstrap (Efron, 1982) approach, replications of the observed data (N_1, N_2, $N_{BG,3}$) are obtained by simulating Poisson random variables with means N_1, N_2 and $N_{BG,3}$. To insure physically meaningful results, replications for which simulated background is larger than either of the other signals are discarded. Also, replications for which twice background equals the sum of the other signals are discarded. This second condition insures that computed asymmetry is well defined. Thus, the k^{th}bootstrap replication of the observed data satisfies the following constraint.

$$N_{BG,3}{}^k \leq N_1{}^k \quad (5)$$

$$N_{BG,3}{}^k \leq N_2{}^k \quad (6)$$

$$2\, N_{BG,3}{}^k < N_1{}^k + N_2{}^k \quad (7)$$

Because of this constraint, the three simulated signals are correlated with one another. The true asymmetry is estimated by the mean of the bootstrapped asymmetry statistics. A confidence interval is also computed from the histogram of the bootstrapped asymmetry statistics.

4 Applications

4.1 High R

First, the Poisson parameters for the data were set to $(240, 60, 50)$. For this case, true asymmetry is 0.9. One thousand data sets, where simulated background is larger than one of the other two signals, were simulated. For each data set, 10,000 bootstrap replications were simulated as described earlier. In Figure 1, the histogram of bootstrapped asymmetry statistics for one of simulated data sets, $(N_1, N_2, N_{BG,3}) = (220, 59, 65)$, is shown. For this particular data set, the mean of the 10,000 bootstrapped asymmetry statistics was 0.924. Hence, the bootstrap estimate of asymmetry $\hat{R}$ is 0.924. This is very close to the true value of 0.9 ! Intuitively, the method worked well because N_1 was much larger than N_2. The fact that the two measurements are far apart is telling us that asymmetry is high even though background is larger than N_2.

In Table 1, bootstrap estimates of asymmetry are listed for ten simulated data sets. The average of the bootstrap estimates for all 1000 data sets was 0.938. The standard error of this average value is only 0.0004. Thus, the bootstrap estimate of asymmetry is slightly biased. Although slightly biased, root mean square prediction error (RMS) was only 0.039.

A confidence interval for true asymmetry is computed from the histogram of bootstrapped statistics as follows. If the asymmetry estimated from the observed data $\hat{R}$ is larger than unity, i.e. $N_1 > N_2$, a one-sided confidence interval is computed. The upper endpoint of the interval is unity. The lower endpoint is the 5% percentile of the bootstrap histogram. If i.e. $N_2 > N_1$, the lower endpoint is set to -1 and the upper endpoint is the 95% percentile of the bootstrap histogram. If $N_1 = N_2$, the confidence interval endpoints are the 2.5% and 97.5% percentiles.

In Table 1, confidence intervals are listed for the ten data sets. For the 1000 simulated data sets, true asymmetry 0.9 was outside the computed bootstrap confidence interval 9 times out of 1000. That is, coverage was 99.1%. All the upper endpoints were unity. Hence, for this case, the bootstrap method gave conservative 95% confidence intervals.

Table 1. High Asymmetry.

N_1	N_2	$N_{BG,3}$	$\hat{R}$	c.i.
220	59	65	0.924	(0.803,1.0)
243	63	64	0.916	(0.796,1.0)
227	50	53	0.928	(0.817,1.0)
224	50	56	0.935	(0.829,1.0)
236	48	60	0.953	(0.867,1.0)
230	40	62	0.968	(0.899,1.0)
238	44	51	0.947	(0.856,1.0)
219	66	72	0.915	(0.779,1.0)
236	50	61	0.948	(0.853,1.0)
261	50	51	0.936	(0.840,1.0)

4.2 Intermediate R

The same kind of analysis done above was repeated for the case where the true Poisson parameters were assumed to be $(460, 420, 400)$. Here, true asymmetry is 0.5. In Table 2, the bootstrap estimate of asymmetry and a confidence interval are listed for ten data sets. For the data set (433,393,394), the histogram of bootstrapped asymmetry statistics is shown in Figure 2. Note that this histogram is more dispersed than the one for the data set (220,59,65).

Table 2. Intermediate Asymmetry.

N_1	N_2	$N_{BG,3}$	$\hat{R}$	c.i.
433	393	394	0.416	(-0.217,1.0)
435	421	423	0.148	(-0.692,1.0)
461	420	444	0.415	(-0.368,1.0)
487	430	439	0.532	(-0.014,1.0)
447	378	394	0.626	(0.171,1.0)
479	391	406	0.689	(0.304,1.0)
493	390	391	0.703	(0.368,1.0)
484	395	401	0.673	(0.298,1.0)
481	408	410	0.604	(0.170,1.0)
434	441	437	-0.074	(-1.0,0.793)

The mean value of the 1000 bootstrap estimates of asymmetry is 0.497. The standard error of this average is 0.007. Although the bootstrap estimate is not significantly biased, root mean square error is larger than before. Here. $RMS = 0.215$ whereas before, i.e. for the high asymmetry case, RMS was over five times less.

The true value of the asymmetry fell in the confidence interval constructed from the bootstrapped asymmetry statistics 988 out of 1000 times (98.8%). Seven times the lower endpoint was greater than 0.5. Five times the upper endpoint was less than 0.5. Thus, the bootstrap confidence interval is again conservative.

4.3 Interval Width

In Figure 3, the width of the confidence interval for each of the 2000 simulated data sets from the high and intermediate asymmetry study are plotted versus $\| \frac{N_1 - N_2}{N_1 + N_2} \|$. This ratio is a measure of how close N_1 and N_2 are to one another. In general, the intervals are broadest when N_1 and N_2 are closest.

4.4 Background Study

In order to study how background affects the accuracy of the bootstrap estimate, the Poisson parameters were set to be $(\lambda_1, \lambda_2, \lambda_{BG}) = (100 + x, 5 + x, x)$ where $x = 5, 10, 20, 50, 100, 200, 500, 1000$. Asymmetry is 0.905 for each value of x. For each set of parameters, 1000 data sets, where background exceeds one of the other signals, were simulated. A confidence interval and an estimate for asymmetry were computed for each data set. In Table 3, the average of the estimates with the standard deviation of the estimates in parentheses, root mean square error, coverage fraction and average length of the confidence intervals are listed.

Table 3. $(\lambda_1, \lambda_2, \lambda_{BG}) = (100 + x, 5 + x, x)$.

x	Ave.$\hat{R}$	RMS	Coverage	|c.i.|
5	0.963(0.010)	0.059	0.854	0.108
10	0.949(0.013)	0.046	0.974	0.142
20	0.932(0.018)	0.033	1.000	0.184
50	0.893(0.026)	0.029	1.000	0.272
100	0.853(0.035)	0.062	1.000	0.359
200	0.796(0.059)	0.124	1.000	0.478
500	0.702(0.098)	0.225	1.000	0.678
1000	0.601(0.145)	0.338	1.000	0.896

For $x \leq 20$, the asymmetry estimate was biased high. For larger backgrounds, the estimate was biased low. This downward bias for high background is plausible because the difference between N_1 and N_2, in units of standard deviations of either one, diminishes as background increases. As the standardized difference between signals tends to zero, confidence in claiming that true asymmetry is close to unity diminishes.

As background increases, both the variability of $\hat{R}$ and the average confidence interval length increase. However, RMS does not increase monotonically since RMS depends on both bias and variability. The coverage of the bootstrap confidence intervals for low background was less than 95%. This probably is due to both the shortness of the intervals and the bias of the estimate. At larger background levels, bias is greater but the confidence intervals are longer and coverage is 100%.

4.5 Other Examples

For the case $(\lambda_1, \lambda_2, \lambda_3) = (600, 505, 500)$, the average bootstrap estimate of asymmetry was 0.702 whereas true asymmetry was 0.905 (Table 3). When the second parameter is changed to 510, true asymmetry drops to 0.818 from 0.905 (Table 4). However, the expected value of $\hat{R}$ was almost the same and the variability of $\hat{R}$ diminished only slightly. This is reasonable; if the difference between λ_2 and λ_{BG} is very slight, the expected value of the bootstrap estimate will depend mostly on λ_1 and λ_{BG}.

Table 4.

λ_1	λ_2	λ_{BG}	R	Ave.$\hat{R}$	RMS
600	505	500	0.905	0.702(.098)	0.225
600	510	500	0.818	0.704(.087)	0.144
573.5	531.5	500	0.400	0.473(0.236)	0.247
563	542	500	0.200	0.295(0.319)	0.333

In two other examples, the first and second parameter were both adjusted so that true asymmetry was 0.4 and 0.2. However, the sum of the two parameters was invariant. Thus, the difference between λ_2 and λ_{BG} is increased as the difference between λ_1 and λ_2 is diminished. For these cases, the variability of $\hat{R}$ and RMS were greater

than for the high asymmetry example. However, bias was less. The results are summarized in Table 4. Note that in parentheses, the standard deviation of $\hat{R}$ is indicated. The coverage of the bootstrap confidence intervals for the second, third and fourth examples in Table 4 were 100%, 98.4% and 95.8%.

A simulation study was also done for the example $(\lambda_1, \lambda_2, \lambda_3) = (605, 499, 475)$. For this case, true asymmetry is 0.688. The average of the bootstrap estimate of asymmetry for a 1000 data set study was 0.739(.078), RMS was 0.092 and coverage was 100%.

5 Conclusion

For cases where the observed background signal was larger than either of the other signals, the asymmetry computed from the data is nonsensical. Using a bootstrap approach, true asymmetry and an associated confidence interval were estimated. In the bootstrap method, replicated data sets satisfied a constraint that insured physically meaningful results. For all cases except very low background signal cases, the bootstrap 95% confidence intervals were conservative. For some cases, the bootstrap estimate for asymmetry had very small prediction error. The prediction error of the bootstrap estimate was greatest for cases where the background was very large relative to the other signals. The bootstrap estimate of asymmetry, i.e. the mean of the bootstrapped asymmetry statistics, was biased in general. The magnitude of the bias was greatest for cases where background was very high and asymmetry was was close to unity (0.905). For other high background cases where asymmetry was less extreme, bias was less but RMS was larger.

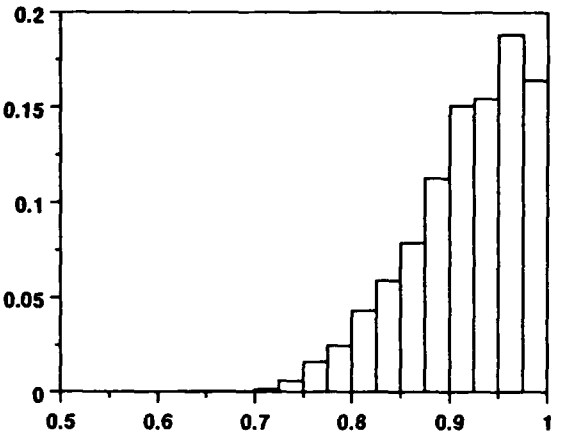


Figure 1. Bootstrap replications of asymmetry statistic for data set (220,59,65)

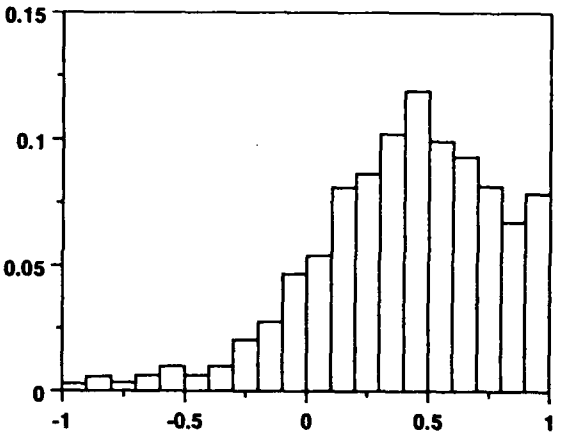


Figure 2. Bootsrap replications of asymmetry statistic for data set (433,393,394)

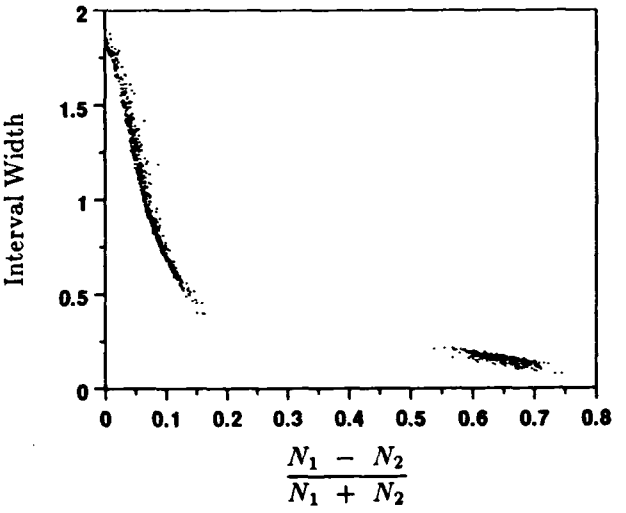


Figure 3. Length of Bootstrap confidence intervals.

6 Acknowledgements

Conversations with Stefan Leigh, Keith Eberhardt and Charles Hagwood were useful.

References

[1] B. Efron, "The Jacknife, the bootstrap and other resampling plans," *CBMS, SIAM-NSF,* 1982

[2] J.J. McClelland, M.H.Kelley, and R.J. Celotta, "Superelastic scattering of spin-polarized electrons from Sodium," *Phys. Rev. A,* Vol. 40, No. 5, pp. 2321-2329, 1989

[3] M.R. Scheinfein, J. Unguris, M.H. Kelley, D.T. Pierce and R.J. Celotta, "Scanning electron microscopy with polarization analysis (SEMPA)," *Review of Scientific Instruments,* Vol 61, No. 10, pp. 2501-2526, 1990

On Constructing Confidence Intervals for Functions of a Multinomial Parameter

Robert Koyak*
Antitrust Division
U.S. Department of Justice
555 Fourth Street, N.W.
Washington, D.C. 20001

Abstract

We consider the problem of constructing confidence intervals for possibly "messy" functions of a multinomial parameter. The number of categories can be large and the sample size small, meaning that the problem of sparseness must be confronted. Thus, standard asymptotics based on the delta method will often prove unsatisfactory. Alternatives to the delta method include: (1) Madansky's method, based on constrained maximum likelihood; (2) the bootstrap; and (3) intervals derived from the brute force (Monte Carlo) calculation of exact confidence regions. These approaches are discussed and contrasted in the context of an empirical problem.

1 Introduction

Let π denote the vector parameter of a multinomial distribution, and let $\theta = \theta(\pi)$ be a "smooth" (i.e., differentiable) scalar-valued function of π. Suppose that a random sample of size N is taken from Multin(π), from which we wish to construct a confidence interval for θ. We can approach this problem in a variety of ways, ranging from computationally intensive "exact" methods, to the bootstrap, to less computationally intensive but approximate methods based on asymptotic arguments. But, how workable are these methods in a particular instance where both the dimensionality of π and N are large, but N is not large enough to justify faith in the validity of asymptotic approximations?

This paper is a summary of ongoing research motivated by a problem arising from estimating the degree of concentration of an economic market. An important concern of the Antitrust Division of the U.S. Department of Justice is to maintain competitive economic markets. Markets that are the least concentrated—that is, those that are not dominated by a small number of firms—tend to be the most responsive, other things being equal, to the discipline of competition. Over time, a given market will become more concentrated if existing firms fail or exit the picture; if a few highly successful competitors gain larger market shares; or, if mergers occur among existing firms. The first two of these often arise from market forces themselves, and are seldom amenable to or appropriate for regulatory control. Mergers, however, can be and often are contested by the Antitrust Division in the interest of keeping markets competitive.

The decision to contest a merger depends on a complex analysis, including the level of concentration in the market both before and after the merger. While there is no single right way of measuring concentration, the Herfindahl Index has emerged as a favorite in much of the microeconomic literature, and in the Antitrust Division since 1982. If a market has K firms, with market shares $\pi_1, \ldots, \pi_K$, where $\pi_k > 0$ and $\sum_{k=1}^{K} \pi_k = 1$, the Herfindahl Index is is defined by $H = 10,000 \times \sum_{k=1}^{K} \pi_k^2$. It is easy to see that H assumes its smallest possible value of $10,000/K$ when the market is least concentrated (equal market shares for all firms), its largest possible value of $10,000$ when it is most concentrated (monopoly), and is increased if two or more of the firms merge. One of the great virtues of the Herfindahl Index is its simplicity: even nonmathematically astute judges, lawyers, and jurors can understand it, and multiplication by $10,000$ eliminates the need for fractions.

2 Estimation

Occasionally, Herfindahls are estimated when market shares are imputed from a "random sample" of con-

*This paper does not purport to represent the policy or views of the U.S. Department of Justice

sumers. The implied model is that a randomly sampled consumer responds in favor of firm k with probability equal to its market share. Suppose that N consumers are sampled, and that $\hat{\pi}_k$ are the sampled proportions of the firms. The naïve estimate of H is then $\hat{H} = \sum_{k=1}^{K} \hat{\pi}_k^2$ (we will from here on drop multiplication by 10,000). The mean and variance of $\hat{H}$ are easily found to be

$$E(\hat{H}) = \frac{1}{N} + \frac{N-1}{N} H$$

and

$$Var(\hat{H}) = \frac{A_1}{N} + \frac{A_2}{N^2} + \frac{A_3}{N^3}, \qquad (2.1)$$

where $A_1 = 4(T - H^2)$, $A_2 = 6H + 10H^2 - 16T$, $A_3 = 6(2T - H - H^2)$, and $T = \sum_{k=1}^{K} \pi_k^3$. Using the fact that $H^2 \leq T \leq H$ gives

$$Var(\hat{H}) \leq H(1-H) r_N, \qquad (2.2)$$

where $r_N = 4N^{-1} + 6N^{-2} + 6N^{-3}$. Appealing to the asymptotic normality of $\hat{H}$, an asymptotically conservative $100(1-2\alpha)\%$ confidence interval for H can be derived as follows:

$$\frac{\hat{\theta}_{\alpha,N} \pm \sqrt{\hat{\theta}_{\alpha,N}^2 - 4\left(\hat{H} - \frac{1}{N}\right)^2 \left[\left(\frac{N-1}{N}\right)^2 + z_\alpha^2 r_N\right]}}{2\left[\left(\frac{N-1}{N}\right)^2 + z_\alpha^2 r_N\right]} \qquad (2.3)$$

where $\hat{\theta}_{\alpha,N} = 2\left(\hat{H} - \frac{1}{N}\right)\left(\frac{N-1}{N}\right) + z_\alpha^2 r_N$ (see Bickel and Doksum [1977], p. 160). Replacing z_α by a Chebyshev inequality bound (i.e., $\sqrt{20}$ for 1.96 at $\alpha = .025$) gives a confidence interval with guaranteed coverage for every N.

Suppose that firms 1 and 2 propose to merge. This would increase the Herfindahl index by an amount equal to $\Delta H = 2\pi_1\pi_2$. Using arguments similar to those above, exact expressions for the mean and variance of $\Delta\hat{H}$ can be given, and a confidence interval similar to (2.3) can be derived. In particular, $E(\Delta\hat{H}) = \left(\frac{N-1}{N}\right)\Delta H$, from which it is seen that the estimated Herfindahl tends to be biased *upward*, but that the estimated change due to a merger tends to be biased *downward*.

Usually, the market shares $\{\pi_k\}$ are regarded as known quantities, and H is not the subject of statistical inference. Even with this knowledge, however, the issue of inference may arise if a more detailed analysis is desired. Suppose that the universe of consumers is partitioned into J strata, with $\pi_{k,j}$ representing the share of the market belonging to firm k and stratum j, and $\pi_{k,*}$ and $\pi_{*,j}$ denoting marginalized market shares. A "weighted" Herfindahl Index is given by

$$H^{(w)} = \sum_{j=1}^{J} \pi_{*,j} \sum_{k=1}^{K} \left(\frac{\pi_{k,j}}{\pi_{*,j}}\right)^2, \qquad (2.4)$$

which is simply a convex combination of the stratum-specific Herfindahls. As before, let $\hat{\pi}_{k,j}$ denote the estimated cell shares based on a random sample of N consumers. *We will assume that the marginal firm shares $P_k = \pi_{k,*}$ are known*, giving $\tilde{\pi}_{k,j} = \hat{\pi}_{k,j} P_k / \hat{\pi}_{k,*}$ as the MLE of $\pi_{k,j}$. Take $\tilde{\pi}_{*,j} = \sum_{k=1}^{K} \tilde{\pi}_{k,j}$. Substituting these estimates into (2.4) gives an estimated weighted Herfindahl that we will denote $\tilde{H}^{(w)}$, with $\Delta\tilde{H}^{(w)}$ obtained in a similar manner.

In one particular problem, $K = 10$, $J = 5$, $N = 400$, and many cells in the $\hat{\pi}_{k,j}$ table were empty. Some of these zeros were surely structural, but others were plausibly induced by sampling. It seems intuitive that the weighted estimates should exhibit greater bias than their unweighted counterparts, and simulations have borne this out. One might ask whether taking advantage of knowing the firm shares P_k is more trouble than it is worth: for instance, an estimate of $H^{(w)}$ taking the form

$$\hat{\theta}^{(w)} = \sum_{j=1}^{J} \hat{\pi}_{*,j} \left(\hat{H}_j - \frac{1}{N\hat{\pi}_{*,j}}\right)\left(\frac{N\hat{\pi}_{*,j}}{N\hat{\pi}_{*,j} - 1}\right), \qquad (2.5)$$

is unbiased conditioned on $N\hat{\pi}_{*,j} > 1$, with $\hat{H}_j$ denoting the estimated Herfindahl for stratum j using the observed proportions. We used the constrained estimates because they were proposed by the parties contemplating the merger, who could argue that the unconstrained estimates failed to take advantage of known information, leading to discrepancy measures that unfairly worked against them.

3 Confidence Intervals

Between the blunt edges of asymptotics and brute computing force lie a variety of methods for deriving confidence intervals. We discuss several in the context of estimating Herfindahl indices. Our conclusions are based on simulation exercises taking $K = 10$, $J = 5$, and $N = 400$; $(P_1, \ldots, P_{10}) = (.20, .20, .20, .15, .10, .05, .03, .03, .03, .01)$; equal stratum probabilities of .2; and, independence between stratum and firm. Figure 1 shows histograms of 1000 simulated values of $\tilde{H}^{(w)}$ (truth = .1578) and $\Delta\tilde{H}^{(w)}$ (truth = .08).

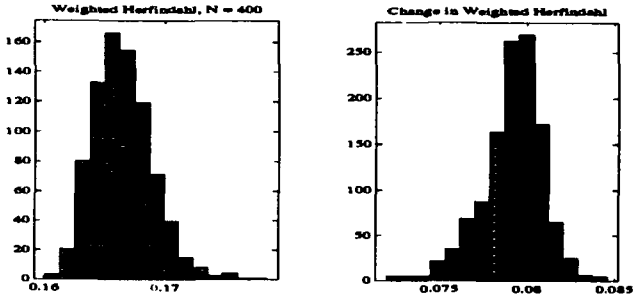


Figure 1: Simulation Histograms

3.1 The Delta Method

Both $\tilde{H}^{(w)}$ and $\Delta\tilde{H}^{(w)}$ are consistent and asymptotically normal estimates, because they are "regular" functions of the observed cell proportions. Thus, asymptotic standard errors can be derived by evaluating the gradients of these functions at the observed proportions, and using the facts that $Var(\hat{\pi}_{k,j}) = N^{-1}\pi_{k,j}(1-\pi_{k,j})$ and $Cov(\hat{\pi}_{k,j}, \hat{\pi}_{r,s}) = -N^{-1}\pi_{k,j}\pi_{r,s}$. This is tedious and ultimately not satisfying, because (1) the standard error estimates are poor; (2) the estimates are substantially biased; and (3) the estimates have sampling distributions that are not very normal-like. It is interesting to note that all of the simulated values of $\tilde{H}^{(w)}$ exceeded the truth, while $\Delta\tilde{H}^{(w)}$ exhibited less bias. These results are summarized below:

	Bias		SE	
	Truth	Mean	Simulation	Theory
$\tilde{H}^{(w)}$	.1578	.1666	.0025	.0009
$\Delta\tilde{H}^{(w)}$	.0800	.0793	.0017	.0023

The "simulation SE" is the standard deviation of the simulated values. The "theory SE" refers to the average, across simulations, of the estimated standard error obtained from the delta method.

It is often worthwhile to apply a variance-stabilizing transformation, if known, when deriving confidence intervals. This cannot be done exactly in the present context, due to the presence of "nuisance parameters." In the case of $\hat{H}$, (2.2) suggests that an arcsine transformation may come close to doing the job, and it may be a good first guess for $\tilde{H}^{(w)}$ as well.

3.2 Confidence Regions

If $\hat{C}_\alpha$ is a $100(1-2\alpha)\%$ confidence region for the multinomial parameter π, the minimum and maximum values of $\theta(\cdot)$ evaluated over the region can be called a $100(1-2\alpha)\%$ confidence interval for $\theta(\pi)$. There are many ways to choose $\hat{C}_\alpha$, and not all of them will produce good (narrow) confidence intervals for $\theta(\pi)$. In fact, the best confidence *regions* will not generally produce the best confidence *intervals*. Intuition suggests that desirable confidence regions will follow the contours of $\theta(\cdot)$ as closely as possible, and will concentrate as much of their mass as possible between them.

3.3 Constrained MLE

A likelihood-based confidence interval can be obtained for $\theta(\pi)$ in the following manner. Let

$$L(\pi; X) = \binom{N}{X} \prod_{j=1}^{J} \prod_{k=1}^{K} \pi_{k,j}^{x_{k,j}}, \qquad (3.1)$$

where $X = [x_{k,j}]$ is the matrix of observed cell frequencies. Let $S_{K,J}$ denote the region of the unit simplex in $\mathcal{R}^{K\times J}$ that has marginal firm shares equal to the known quantities. Fix a value θ_0, and let $S_{K,J}(\theta_0)$ denote the subset of $S_{K,J}$ for which $\theta(\pi) = \theta_0$. The MLE of π over $S_{K,J}$ and $S_{K,J}(\theta_0)$ will be denoted $\tilde{\pi}$ and $\tilde{\pi}(\theta_0)$ respectively.

Consider a test of the hypothesis $\mathcal{H}(\theta_0)$: $\theta(\pi) = \theta_0$ based on the statistic

$$R(\theta_0; X) = \frac{L(\tilde{\pi}; X)}{L(\tilde{\pi}(\theta_0); X)}. \qquad (3.2)$$

For $\pi \in S_{K,J}(\theta_0)$, $2\log R(\theta_0; X)$ is asymptotically distributed as chi-square with one degree of freedom. This can be used to test $\mathcal{H}(\theta_0)$, with the set of θ_0 for which $\mathcal{H}(\theta_0)$ is accepted giving an asymptotically valid confidence interval for $\theta(\pi)$. Recently, Owen (1990) has extended this classical idea to a nonparametric context.

One difficulty with this approach is the computation of the constrained MLE $\tilde{\pi}(\theta_0)$. Treating it as a Lagrange multiplier problem requires the simultaneous solution of a large set of nonlinear equations, a numerically difficult problem for which no method is guaranteed safe and sure. Projected gradient methods may offer the best hope, provided that good starting values are available, which is often the case. We have used projected gradients with success only in lower dimensional problems.

If g is a 1-1 function over the range of θ, applying g^{-1} to a confidence interval obtained for $g \circ \theta(\pi)$ gives a confidence interval for $\theta(\pi)$. Sometimes working with $g \circ \theta$ offers computational advantages over working with θ. One particular choice for g that often seems to work well is the Lagrange multiplier attached to the constraint

$\theta(\pi) = \theta_0$, an idea which is due to Madansky (1965). For example, for $\theta = H$ (unweighted Herfindahl), it can be shown that θ_0 and the Lagrange multiplier λ are in a 1-1 decreasing relationship, and that solving the constrained "normal equations" reduces to finding the fixed point of a contraction mapping when λ is positive.

The reliance on asymptotics can be obviated if the exact finite sample distribution of $R(\theta_0; X)$ is used. This is usually not feasible analytically, but it can be done via simulation within the context of a projected gradient problem. We have not tried this, so the extent to which it pays off to expend the additional effort is not clear.

3.4 The Bootstrap

An advantage of using the bootstrap to construct confidence intervals is its ease of implementation, which in its simplest form is basically the same for all problems. We have used the bootstrap to construct confidence intervals for H, ΔH, $H^{(w)}$ and $\Delta H^{(w)}$. Our conclusion is that the simplest use of the bootstrap produces disappointing results, but that it offers a fertile area for experimentation.

The "simple" bootstrap proceeds by taking B random samples from a Multin $(\hat{\pi})$ distribution, computing estimates of the desired quantity from the B bootstrap samples, and choosing quantiles of the bootstrap estimates corresponding to the desired confidence level. This technique can produce good confidence intervals if certain conditions are satisfied. One condition—that the estimates be asymptotically normal and consistent—is in our view not too severe. The estimates should also exhibit little or no finite sample bias, and the effect of not knowing the values of any nuisance parameters that may be present should be negligible. These last two conditions are more troublesome, and embellishments to the simple bootstrap have been made to deal with them: various bias-correction schemes and pivoting, respectively. They, and bootstrapping in general, are discussed in Efron (1982).

Our best results were obtained for ΔH using $\Delta \hat{H}$, with $B = 1000$, no bias correction, and no pivoting. Based on 250 replications of the model described above, the estimated lower and upper tail violation probabilities for a 95% confidence interval were 1.2% and 2.8% respectively. Other quantities fared much worse, with much larger-than-nominal violation probabilities, and badly unbalanced intervals. Bias correction is obviously needed, but the usual quantile-adjustment methods have not worked well because the bias tends to be of a much larger order than the standard error.

One problem with bootstrapping large-order multinomials is sparseness: the bootstrap samples contain at least as many empty cells as the root sample, and often more. Thus, if the quantity of interest is sensitive to sparseness, the bootstrap may produce disappointing results. One outcome of this that we have observed is that the variance within bootstrap samples can be much smaller than the variance between, which bodes ill for staying true to nominal coverage levels. A possible corrective measure would be to "smooth out" sparseness before bootstrapping, using either a Bayesian or a non-Bayesian argument. None of these problems detract from the asymptotic validity of the bootstrap, but they do underscore the need to carefully study its small sample behavior.

4 Discussion

While producing good standard error estimates is usually easier than producing good confidence intervals, many of us prefer the latter. In the context of estimating Herfindahls, we felt that it was important to show that a variety of states of nature could plausibly explain a given set of sample results. However, even with a full menu of options to choose from, constructing good small-sample confidence intervals is not an easy problem, and there is much room for further experimentation.

References

Bickel, P.J. and Doksum, K.A. (1977). *Mathematical Statistics*, Holden-Day, Oakland.

Efron, B. (1982). *The Jacknife, the Bootstrap, and Other Resampling Plans.* CBMS No. 38, SIAM, Philadelphia.

Madansky, A. (1965). *Technometrics* **(7)** 495–503.

Madansky, A. and Olkin, I. (1969). Approximate confidence regions for constraint parameters. *Multivariate Analysis II* 261–286 (P.R. Krishnaiah, ed.) Academic Press, New York.

Owen, A. (1990). *Ann. Math. Statist.* **(18)** 90–120.

RANDOMIZED NEWTON-RAPHSON AND ANIMAL SEARCH

A. Levine and J. Liukkonen
Department of Mathematics & Statistics
Tulane University, New Orleans, LA 70118, U.S.A.

ABSTRACT. Adding "systematic noise" to the step term of the Newton-Raphson (NR) root finding algorithm permits "expected q-linear convergence" and convergence almost surely to the root for a larger class of functions and larger starting sets than those for which NR converges "deterministically." These results have application not only to a wide range of optimization problems but also to understanding the behavioral repertory of animals undertaking pheromone induced search. It is shown that the "search" reduces in many cases to finding the root of a function of two or three dimensions. In cases (as in the search of the gypsy moth for its mate) where the animal cannot simply travel in the direction of increasing signal (scent) randomized NR gives insight into the search behavior required to discover the signal source.

1. INTRODUCTION. The numerical determination of a global maximum or minimum of a function $g : R^m \to R^n$ where R^t represents Euclidean t-space is commonly accomplished through an iterative algorithm $x_k = H_{k-1}(x_{k-1}, \ldots, x_0)$, $k = 1, 2, \ldots, x_0 \in E \subset R^m$, where E is the set of initial solution estimates and $\{x_k\}$, $\{H_k\}$ represent, respectively, a sequence of solution estimates (which we call the *path*) and operators on the estimates. It has long been recognized that numerical algorithms are subject to "unacceptable convergence;" i.e., non-convergence to a solution or convergence too slow to yield practical results. Paths in two or more dimensions are particularly subject to traps, to being caught in ridges or to cycling. Joseph, et al (1990) present examples where this type of non-convergence occurs for one dimensional paths as well. To ameliorate these problems, the authors introduced *randomized* Newton-Raphson (RNR) in which a random element is injected into the Newton-Raphson algorithm in order to allow cycles to be broken or to permit large jumps along the path towards the solution, thereby increasing the speed of convergence. (See Joseph, et al (1991) for an application in two dimensions to a problem in seismic exploration.) We examine RNR in Sec.2 and in Sec.3 introduce some new results related to it.

Of additional interest here is that RNR serves as a source for conceptual models of animal search, especially where chemical systems provide the dominant means of communication. This matter is discussed in Secs.2 and 3 in association with the analytical issues raised there. Applications are discussed in Sec.4.

2. RANDOMIZED NEWTON-RAPHSON (RNR): The Newton-Raphson (NR) algorithm for $g : R \to R$, W a compact interval in R, g' the derivative of g, E the set of initial values, $x_0 \in E \subset W$ is

$$\begin{aligned} x_{k+1} &= x_k - y_k,\ x_k \in W \\ y_k &= g(x_k)/g'(x_k) \qquad \text{(step term)} \end{aligned} \tag{2.1}$$

with some stopping rule. NR is a root finding algorithm where $\{x_k\}$ is a sequence of *iterates* which, under certain conditions, converges to the root p of g. We use NR as a paradigm because of its optimality properties (Ortega and Rheinboldt (1970)). Furthermore, its use permits a concise presentation while at the same time making clear the methods by which the results may be extended to other root finding algorithms.

In "animal search" we assume that the prey (or potential mate) omits a signal (such as a gas) which produces a continuous spatial distribution having a unique maximum $h(p)$ at the location p of the source at any given time. We also assume that h has no local minima. (We will subsequently weaken these assumptions.) Under these conditions, there are many ways of changing the animal search problem into a root finding problem. For example, we can assume the animal "knows" the threshold value of $h(p)$. Defining $g(x) = h(x) - h(p)$, the problem becomes one of finding the *unique* root of $g(x)$. Another method of changing the problem to root finding is to let $g(x) = h'(x)$. Then the position p of the source can be found by solving $g(x) = 0$. We shall assume for the moment that the "maximization" problem of the animal can be transformed into finding the root p of a function. We call the location p, the *target* of the algorithm.

In Sec.4, we shall motivate the introduction of RNR into the issue of animal search, but for the moment we present RNR as capable of resolving areas of "unacceptable convergence" as described in Sec.1.

To define RNR in one-dimension, the step term in (2.1) is now a random variable $Y_k = (g(x_k) + Z_{1k})/(g'(x_k) + Z_{2k})$ where Z_{1k} is a random variable having a density and zero expectation; Z_{2k} is either an independent, continuous random variable with the same sign

as $g'(x_k)$ or is a constant with the same sign as $g'(x_k)$. The RNR algorithm is

$$X_{k+1} = x_k - Y_k \quad \text{when } x_k \in W,\ x_k \text{ is a realization of } X_k$$
$$X_{k+1} = x_{k-1} \quad \text{when } x_k \notin W \text{ (re-set)}. \tag{2.2}$$

The stopping rule for the algorithm is: for a specified t, and $S_t = \{x : |g(x)| < t\} \subset W$, stop when $X_{k+1} = x_k \in S_t$. (Note, the re-set condition in (2.2) can be replaced by the re-start condition: X_{k+1} is uniform over W when $x_k \notin W$.) Joseph, et al (1990) obtained results for the convergence (in a probabilistic sense) of RNR to the root. We do not repeat them here since we develop stronger results in the next section.

Animal search goes on in more than one-dimension. The generalization of (2.2) to higher dimensions can be found in Joseph, et al (1990). For conceptual purposes, our discussion mainly will be in one-dimension but the results are easily generalized to higher dimensions.

The algorithm (2.2) permits steps of arbitrary size, steps which can overshoot the target by a greater amount than is permissible in any animal tracking problem. To avoid unrealistic step sizes, in all that follows, we fix Z_{2k} as a constant, the size of which depends on each particular tracking problem and the time frame permitted for each iteration.

It is important to emphasize that the random variables $\{Z_{ik}\}$ are injected into the system by the algorithm. These "noise" terms are not introduced externally as is done with "Robbins-Monro" which attempts to extract the signal $g(x)$ from the noise. In RNR the purpose is to add "noise" in a controlled way to the NR algorithm to obtain convergence in some cases where NR does not yield acceptable convergence.

3. STRONG CONVERGENCE.

We present here some new results on the convergence of RNR to a unique root which are both of general interest in numerical computation and to the issue of animal search. The question, what conditions on the density of the random variable Z_{1k} are required to insure convergence, is treated in Th.1. What happens to the search if these conditions are (mildly) not met is treated in Ex.1.

Lemma 1. (Dinwoodie). Let $\{X_k\}$ be a sequence of random variables such that $E(|X_{k+1}|) \le cE(|X_k|)$ for all k and for some positive $c < 1$. Then $X_k \to 0$, a.s..

Proof: Clearly, $\sum_{i=1}^{\infty} E|X_k| < \infty$. Since, for every $\varepsilon > 0$ and n, $\varepsilon P(|X_k| \ge \varepsilon) \le E(|X_k|)$ so $\sum P(|X_k| \ge \varepsilon) < \infty$ as well. By the Borel-Cantelli lemma, for every $\varepsilon > 0$, $P(|X_k| \ge \varepsilon, i.o.) = 0$. The conclusion follows using Chung (1974), pg.73, Th.4.2.2.

Theorem 1. Suppose $\{X_k\}$ is a Markov process in a compact interval W. Let p represent the target and $\varphi(d) = E(|X_{k+1} - p| \mid X_k - p = d)$ for all $d \in W - \{p\}$. Suppose: (a) for all d in a neighborhood of 0, $\varphi(d) \le c|d|$ for some positive $c < 1$; (b) the conditional density f of X_{k+1} is bounded below as follows: $f_{x_{k+1}|x_k}(y \mid w) \ge b(w)$ for all $y, w \in R$, where the bound b is a continuous function of w, positive except possibly for $w = p$. Then $X_k \to p$ a.s.

Proof: By conditions (a) and (b), there are values r, $a > 0$ such that (i) for $|d| \le r$, $E(|X_{k+1} - p| \mid X_k - p = d) \le c|d|$ and (ii) for $|d| \ge r$, $f_{x_{k+1}|x_k}(y \mid d + p) \ge a$ for all $y \in R$. For each k, let $Y_k = |X_k - p| \wedge r$. Then $\{Y_k\}$ is a sequence of non-negative, uniformly bounded random variables. For $0 \le t \le r$, using (i), $E(Y_{k+1} \mid Y_k = t) \le ct$. When $t = r$, using (ii), $E(Y_{k+1} \mid Y_k = t) \le c't$ for some positive constant $c' < 1$. Consequently, $E(Y_{k+1}) \le c''E(Y_k)$ for some positive constant $c'' < 1$. The result follows from Lemma 1.

This result suggests that the design of the density of the injected random variable Z_{1k} in the algorithm (2.2) is crucial to insuring almost sure convergence to the target. Specifically, this theorem demands that the concentration of the density of X_k about the target p increases "rapidly" as X_k approaches p. The theorem gives an indication of how rapidly this concentration must take place. Moreover, the density of Z_{1k} must be conditionally bounded below (condition (b) of the theorem). This condition insures there are no other points of concentration. It can be relaxed, if condition (a) is correspondingly changed.

Example 1. Suppose we seek the root of $g(x) = 2x - x^2$. Using RNR, $X_{k+1} = x_k - (2x_k - x_k^2 + Z_{1k})/(2 - 2x_k + Z_{2k})$ where Z_{1k}, Z_{2k} are independent random variables; $E(Z_{1k}) = 0$, $Z_{2k} \ge 0$, a.s. (Z_{2k} could be a positive constant.).

(a) Observe that $E(|X_{k+1}| \mid X_k) \le [2 - 2x_k]^{-1}[x_k^2 + |x_k|E(Z_{2k}) + E(|Z_{1k}|)]$. For $|x_k|$ small, if we had designed the density of Z_{1k} so that $E(|Z_{1k}| \mid x_k) \le .9|x_k|$ and the density of Z_{2k} so that $E(Z_{2k} \mid x_k) \le .9$ (or Z_{2k} is a constant less than .9), then $E(|X_{k+1}| \mid x_k) \le .95|x_k|$ ($|x_k|$ small). From our results, we expect a.s. convergence. We performed a simple simulation on a hand calculator using $Z_{2k} \in U(0, 1.8)$, $Z_{1k} \in U(-1.8|x_k|, 1.8|x_k|)$. Stopping when $|g(x)| < 10^{-5}$, and using $x_0 = .75$, we arrive quickly at $x_6 = 3 \times 10^{-6}$.

(b) Note that

$$E(X_{k+1}^2 \mid x_k) = x_k^2 E\left(\frac{x_k^2 + Z_{2k}(Z_{2k} - 2x_k) + x_k^{-2}Z_{1k}^2}{(2 - 2x_k + Z_{2k})^2}\right). \tag{3.1}$$

If $0 \leq Z_{2k} \leq 1$, and we design the density of Z_{1k} so that $V(Z_{1k} \mid x_k) = |x_k|/4$, then the term on the right of (3.1) is greater than or equal to $(x_k^2/25)\left[(4|x_k|)^{-1} + E(Z_{2k}^2 - 2Z_{2k}x_k)\right]$. For small $|x_k|$, $E(X_{k+1}^2 \mid x_k)/x_k^2$ is large. Thus, we do not expect convergence. In fact, letting $Z_{1k} \sim U\left(-\sqrt{3|x_k|/4},\ \sqrt{3|x_k|/4}\right)$ and $Z_{2k} \sim U(0, |x_k|)$ after 2500 iterations, the smallest value of $g(x)$ was .008; the sequence x_k largely oscillated between $-.1$ and .1. There appears to be a barrier to convergence.

The following example treats the question of convergence barriers more generally.

Example 2: Let $\varphi : [0,1] \to [0,.5]$ be any continuous function taking on the value 0 only at zero. Define $s(x,t) = [2\varphi(t)]^{-1} I_{[0,2\varphi(t)]}(x)$ to be the transition kernel of a Markov chain; i.e., with X_0 having an arbitrary density on $[0,1]$ and given the density $f_k(x)$ of X_k, the density $f_{k+1}(x)$ of X_{k+1} is given by

$$f_{k+1}(x) = \int_0^1 s(x,t) f_k(t)\, dt\ .$$

By direct calculation, $E(X_{k+1} \mid X_k = t) = \varphi(t)$. If $\varphi'(0) < 1$, then by Th.1, $X_n \to 0$, a.s. On the other hand, if $\varphi'(0) > e/2 = 1.36$, then X_n does not even converge in probability to 0. In fact, we state without proof, that there exists a nontrivial distribution F_0 as close to the constant 1 as we wish, such that if $F_{X_k}(x) \leq F_0(x)$ for every x, then the same is true of $F_{X_{k+1}}$. Thus, there exist "stable barriers" to convergence.

With respect to the rate of convergence, using the notion of q-linear convergence (Dennis and Schnabel (1983)) we say that we have "expected q-linear convergence" if condition (b) of Th.1 is satisfied. In this sense, Th.1 gives us both the "expected rate of convergence" and the certainty of it.

4. ANIMAL SEARCH. Investigators have observed that animal search often appears random. Their reports suggest thereby that the search is not purposefully directed. As an example, Wilson (1963) observes that the male gypsy moth detecting the "faintly tinted air" produced by the female (perhaps thousands of meters distant) cannot fly in the direction of increasing scent because the "attractant is distributed almost uniformly after it has drifted a few meters from the female." Wilson then describes the path of the moth: "... they simply fly upwind and thus inevitably move toward the female. If by accident they pass out of the active zone, they either abandon the search or fly about at random until they pick up the scent again. Eventually as they approach the female, there is a slight increase in the concentration of the chemical attractant and this can serve as a guide for the remaining distance." The random flying about depicted by Wilson does not suggest purposefully directed behavior. Some consideration of the problem from the standpoint of the moth makes clear that this random flying about is an integral part of the solution. In fact, it is not correct that simply flying upwind the moth "inevitably moves toward the female." As Wilson shows in a figure on page 103, the wind forms a plume from the gas the female emits; unless the moth happens to happily be flying along the "line of sight" to the female, flying upwind must inevitably bring him to the edge of the plume. At this point the moth must use derivative information (decrease in intensity) to return to the plume, not totally along the downwind direction but with some motion along the line perpendicular to the wind's path. The search for the plume cannot be "totally random" for this would suggest that the moth's motion could be modelled by a "random walk" model in two dimensions. Such a model results in infinite expected time to return to the plume. Hence, some deterministic component depending on $g(x)$ (a function inversely proportional to intensity) and $g'(x)$ must be included in the "random" algorithm dictating the moth's motion. These observations are supported by Wilson who notes that increasing scent is a "guide" to the moth. This algorithm must be efficient in order that the moth has a chance for success before it exhausts itself.

Another type of animal search is found in the invisible odor trails fire ant workers leave to guide their colleagues to a food source. The trail consists of a pheromone laid down by workers returning to their nest after finding a source of food. The signal consists of intermittent "hot" spots which decrease in density as the distance from the source increases. Again, one observes a "random" motion of the ant as it hits one "hot" spot and searches for the next. Again, the path of the ant cannot be patterned after a random walk. The function $g(x)$ must be inversely proportional to a *cumulative* sum of "hot spots."

What can go wrong? As the theory of the last section suggests, it is possible for an algorithm not "finely tuned" to produce erratic behavior even near the source. Anyone who has watched an exhausted retriever try to find a source (such as a familiar tennis ball buried in deep grass) will observe that the retriever at times simply steps over the source without ever focusing upon it. It appears that exhaustion has distorted the algorithm into producing a "stable barrier" to convergence.

5. FINAL COMMENTS: We have only here touched upon the connection between animal search and randomized algorithms. We have also investigated a number of models, such as "hot spots" distributed over lattices and

run a number of simulations which appear to demonstrate the utility of RNR type "purposeful random models" in describing the convergence (or lack of it) of animals to a target.

We have shown that a random element is vital to the success of searches based on low intensity or intermittent signals. It would be interesting to investigate the biological mechanisms which produce and regulate random elements. In the case of higher mammals, "anxiety" appears to be such a mechanism.

One area which requires additional research is that in which the signal is contaminated by additive noise in such a way that local minima and maxima are produced thereby violating the assumption of a unique maximum. It appears that the external noise can be added in the algorithm to the "noise" Z_{1k} produced by the animal (or algorithm). Near the source, external noise should have little effect while far from the source, it may be necessary to "design" Z_{1k} so that it dominates the random process.

Another area which demands further investigation because it appears central to gaining insight into the "algorithmic" mechanism employed in animal search is the speed of convergence. Efforts are now being made to find the expected number and variance of the number of steps to solution under general conditions.

REFERENCES

Chung, K. L., 1974, *A Course in Probability Theory*, Academic Press, N. Y.

Dennis, J. E. and R. B. Schnable, 1983, *Numerical Methods for Unconstrained Optimization and Non-Linear Equations*, Prentice-Hall, Englewood Cliffs, N.J.

Joseph, G. J., Levine, A. and Liukkonen, J., 1990, Randomized Newton-Raphson, J. App. Numerical Math., 6, 459–469.

Joseph, G. J., A. Levine, and J. Liukkonen, 1991, A. Stochastic Method for Estimating the Thickness of Subsurface Strata, Proceedings 1991 Offshore Technology Conference, Vol. 1, 491–499.

Ortega, J. M. and W. C. Rheinboldt, 1970, *Iterative Solution of Non-linear Equations in Several Variables*, Academic Press, N. Y.

Wilson, E. O., 1963, Pheromones, Scientific American, 100–114, N. Y.

Note on Learning Rate Schedules for Stochastic Optimization

Christian Darken and John Moody
Yale Computer Science, P.O. Box 2158, New Haven, CT 06520
Email: moody@cs.yale.edu

Abstract

We present and compare learning rate schedules for stochastic gradient descent, a general algorithm which includes LMS, on-line backpropagation and k-means clustering as special cases. We introduce "search-then-converge" type schedules which outperform the classical constant and "running average" $(1/t)$ schedules both in speed of convergence and quality of solution.

Introduction: Stochastic Gradient Descent

The optimization task is to find a parameter vector W which minimizes a function $G(W)$. In the context of learning systems typically $G(W) \equiv \mathcal{E}_X E(W, X)$, i.e. G is the average of an objective function over the exemplars, labeled E and X respectively. The stochastic gradient descent algorithm is

$$\Delta W(t) = -\eta(t) \nabla_W E(W(t), X(t)).$$

where t is the "time", and $X(t)$ is the most recent independently-chosen random exemplar. For comparison, the deterministic gradient descent algorithm is

$$\Delta W(t) = -\eta(t) \nabla_W \mathcal{E}_X E(W(t), X).$$

While on average the stochastic step is equal to the deterministic step, for any particular exemplar $X(t)$ the stochastic step may be in any direction, even uphill in $\mathcal{E}_X E(W(t), X)$. Despite its noisiness, the stochastic algorithm may be preferable when the exemplar set is large, making the average over exemplars expensive to compute.

The issue addressed by this paper is: which function should one choose for $\eta(t)$ (the learning rate schedule) in order to obtain fast convergence to a good local minimum? The schedules compared in this paper are the following (Fig. 1):

- **Constant:** $\eta(t) = \eta_0$

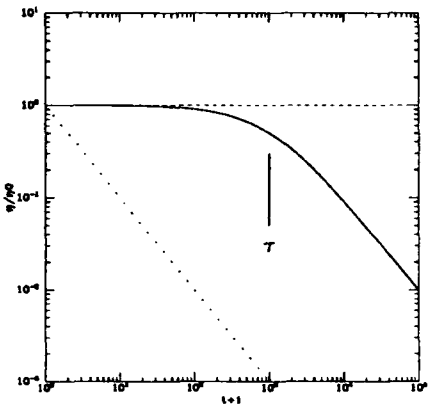


Figure 1: Comparison of the shapes of the schedules. Dashed line = constant, Solid line = search-then-converge, Dotted line = "running-average"

- **"Running Average":** $\eta(t) = \eta_0/(1 + t)$

- **Search-Then-Converge:** $\eta(t) = \eta_0/(1 + t/\tau)$

"Search-then-converge" is the name of a novel class of schedules which we introducein this paper. The specific equation above is merely one member of this class and was chosen for comparison because it is the simplest member of that class. We find that the new schedules typically outperform the classical constant and running average schedules. Furthermore the new schedules are capable of attaining the optimal asymptotic convergence rate for any objective function and exemplar distribution. The classical schedules cannot.

Adaptive schedules are beyond the scope of this short paper (see however Darken and Moody, 1991). Nonetheless, all of the adaptive schedules in the literature of which we are aware are either second order, and thus too expensive to compute for large numbers of parameters, or make no claim to asymptotic optimality.

Example Task: K-Means Clustering

As our sample gradient-descent task we choose a k-means clustering problem. Clustering is a good sample problem to study, both for its inherent usefulness and its illustrative qualities. Under the name of vector-quantization, clustering is an important technique for signal compression in communications engineering. In the machine learning field, clustering has been used as a front-end for function learning and speech recognition systems. Clustering also has many features to recommend it as an illustrative stochastic optimization problem. The adaptive law is very simple, and there are often many local minima even for small problems. Most significantly however, if the means live in a low dimensional space, visualization of the parameter vector is simple: it has the interpretation of being a set of low-dimensional points which can be easily plotted and understood.

The k-means task is to locate k points (called "means") to minimize the expected distance between a new random exemplar and the nearest mean to that exemplar. Thus, the function being minimized in k-means is $\mathcal{E}_X \|X - M_{nrst}\|^2$, where M_{nrst} is the nearest mean to exemplar X. An equivalent form is $\int dX P(X) \sum_{\alpha=1}^{k} I_\alpha(X) \|X - M_\alpha\|^2$, where $P(X)$ is the density of the exemplar distribution and $I_\alpha(X)$ is the indicator function of the Veronois region corresponding to the αth mean. The stochastic gradient descent algorithm for this function is

$$\Delta M_{nrst}(t) = -\eta(t_{nrst})[M_{nrst}(t) - X(t)],$$

i.e. the nearest mean to the latest exemplar moves directly towards the exemplar a fractional distance $\eta(t_{nrst})$. In a slight generalization from the stochastic gradient descent algorithm above, t_{nrst} is the total number of exemplars (including the current one) which have been assigned to mean M_{nrst}.

As a specific example problem to compare various schedules across, we take $k = 9$ (9 means) and X uniformly distributed over the unit square. Although this would appear to be a simple problem, it has several observed local minima. The global minimum is where the means are located at the centers of a uniform 3x3 grid over the square. Simulation results are presented in figures 2 and 3.

Constant Schedule

A constant learning rate has been the traditional choice for LMS and backpropagation. However, a constant rate generally does not allow the parameter vector (the "means" in the case of clustering) to converge. Instead, the parameters hover around a minimum at an average distance proportional to η and to a variance which depends on the objective function and the exemplar set. Since the statistics of the exemplars are generally assumed to be unknown, this residual misadjustment cannot be predicted. The resulting degradation of other measures of system performance, mean squared classification error for instance, is still more difficult to predict. Thus the study of how to make the parameters converge is of significant practical interest.

Current practice for backpropagation, when large misadjustment is suspected, is to restart learning with a smaller η. Shrinking η does result in less residual misadjustment, but at the same time the speed of convergence drops. In our example clustering problem, a new phenomenon appears as η drops—metastable local minima. Here the parameter vector hovers around a relatively poor solution for a very long time before slowly transiting to a better one.

Running Average Schedule

The running average schedule ($\eta(t) = \eta_0/(1+t)$) is the staple of the stochastic approximation literature (Robbins and Monro, 1951) and of k-means clustering (with $\eta_0 = 1$) (MacQueen, 1967). This schedule is optimal for $k = 1$ (1 mean), but performs very poorly for moderate to large k (like our example problem with 9 means). From the example run (Fig. 2A), it is clear that η must decrease more slowly in order for a good solution to be reached. Still, an advantage of this schedule is that the parameter vector has been proven to converge to a local minimum (MacQueen, 1967). We would like a class of schedules which is guaranteed to converge, and yet converges as quickly as possible.

Stochastic Approximation Theory

In the stochastic approximation literature, which has grown steadily since it began in 1951 with the Robbins and Monro paper, we find conditions on the learning rate to ensure convergence with optimal speed [1].

From (Ljung, 1977), we find that $\eta(t) \rightarrow At^{-p}$ asymptotically for any $1 \geq p > 0$, is sufficient to guarantee convergence. Power law schedules may work quite well in practice (Darken and Moody, 1990), however from (Goldstein, 1987) we find that in order to converge at an optimal rate, we must have $\eta(t) \rightarrow c/t$ asymptotically, for c greater than some threshold which depends

[1] The cited theory generally does not directly apply to the full nonlinear setting of interest in much practical work. For more details on the relation of the theory to practical applications and a complete quantitative theory of asymptotic misadjustment, see (Darken and Moody, 1991).

on the objective function and exemplars [2]. When the optimal convergence rate is achieved, $\|W - W^*\|^2$ goes like $1/t$.

The running average schedule goes as η_0/t asymptotically. Unfortunately, the convergence rate of the running average schedule often cannot be improved by enlarging η_0, because the resulting instability for small t can outweigh the improvements in asymptotic convergence rate.

Search-Then-Converge Schedules

We now introduce a new class of schedules which are guaranteed to converge and furthermore, can achieve the optimal $1/t$ convergence rate without stability problems. These schedules are characterized by the following features. The learning rate stays high for a "search time" τ in which it is hoped that the parameters will find and hover about a good minimum. Then, for times greater than τ, the learning rate decreases as c/t, and the parameters converge.

We choose the simplest of this class of schedules for study, the "short-term linear" schedule ($\eta(t) = \eta_0/(1 + t/\tau)$), so called because the learning rate decreases linearly during the search phase. This schedule has $c \equiv \tau\eta_0$ and reduces to the running average schedule for $\tau = 1$.

Conclusions

We have introduced the new class of "search-then-converge" learning rate schedules. Stochastic approximation theory indicates that for large enough τ, these schedules can achieve optimally fast asymptotic convergence for any exemplar distribution and objective function. Neither constant nor "running average" (1/t) schedules can achieve this. Empirical measurements on k-means clustering tasks are consistent with this expectation. Furthermore asymptotic conditions obtain surprisingly quickly. Additionally, the search-then-converge schedule improves the observed likelihood of escaping bad local minima.

As implied above, k-means clustering is merely one example of a stochastic gradient descent algorithm. LMS and on-line backpropagation are others of great interest to the learning systems community. Due to space limitations, experiments in these settings will be published elsewhere (Darken and Moody, 1991). Preliminary experiments seem to confirm the generality of the above conclusions.

Extensions to this work in progress includes application to algorithms more sophisticated than simple gradient descent, and adaptive search-then-converge algorithms which automatically determine the search time.

[2] This choice of asymptotic η satisfies the necessary conditions given in (White, 1989).

Acknowledgements

The authors wish to thank Hal White for useful conversations and Jon Kauffman for developing the animator which was used to produce figure 2. This work was supported by ONR Grant N00014-89-J-1228 and AFOSR Grant 89-0478.

References

C. Darken and J. Moody. (1990) Fast Adaptive K-Means Clustering: Some Empirical Results. In *International Joint Conference on Neural Networks 1990*, **2**:233-238. IEEE Neural Networks Council.

C. Darken and J. Moody. (1991) Learning Rate Schedules for Stochastic Optimization. In preparation.

L. Goldstein. (1987) Mean square optimality in the continuous time Robbins Monro procedure. Technical Report DRB-306. Department of Mathematics, University of Southern California.

L. Ljung. (1977) Analysis of Recursive Stochastic Algorithms. *IEEE Trans. on Automatic Control.* **AC-22**(4):551-575.

J. MacQueen. (1967) Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Math. Stat. Prob.* **3**:281.

H. Robbins and S. Monro. (1951) A Stochastic Approximation Method. *Ann. Math. Stat.* **22**:400-407.

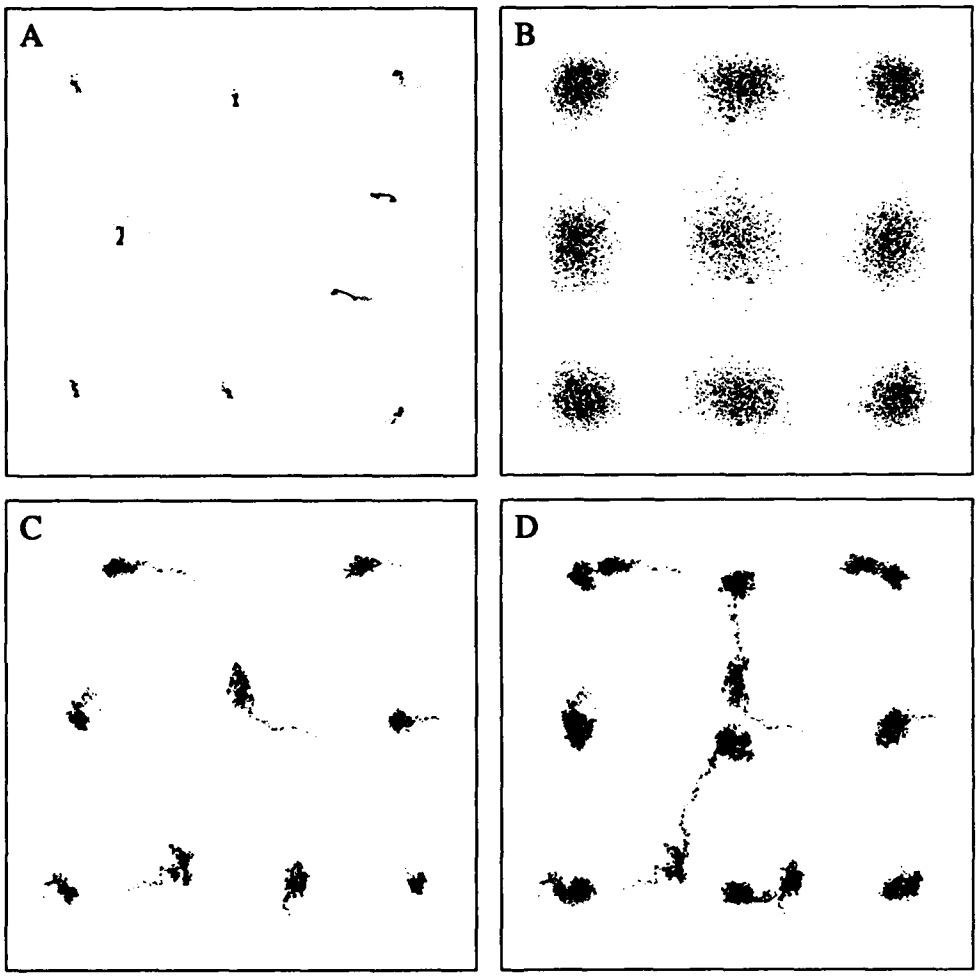


Figure 2: Example runs with classical schedules on 9-means clustering task. Exemplars are uniformly distributed over the square. Dots indicate previous locations of the means. The triangles (barely visible) are the final locations of the means. (A) "Running average" schedule ($\eta = 1/(1+t)$), 100k exemplars. Means are far from any minimum and progressing very slowly. (B) Large constant schedule (η=0.1), 100k exemplars. Means hover around global minimum at large average distance. (C) Small constant schedule (η=0.01), 50k exemplars. Means stuck in metastable local minimum. (D) Small constant schedule (η=0.01), 100k exemplars (later in the run pictured in C). Means tunnel out of local minimum and hover around global minimum.

Genetic Optimization for Exploratory Projection Pursuit

Stuart L. Crawford
Advanced Decision Systems
1500 Plymouth Street
Mountain View, CA 94043

Abstract

Exploratory Projection Pursuit is a technique for forming projections of a multivariate point cloud and searching for those projections that reveal the most structure. The search component is typically some variant of a steepest descent procedure and, particularly when the search space is ill-behaved, leaves open the possibility that the best projection will not be found. Genetic Algorithms are generally applicable optimization techniques, well suited for search spaces in which more traditional techniques fail. This paper describes experiments designed to ascertain the effectiveness of Genetic Algorithms as optimizers for exploratory projection pursuit.

1 Introduction

A common goal of exploratory data analysis is to find structure (clusters, hyperplanes, and the like) among a configuration of points in p-dimensional space. This is a difficult task for large p because high-dimensional space is inherently empty, and procedures that rely on interpoint distances to establish structure fall prey to the "curse of dimensionality". A typical approach to the problem is to reduce dimensionality with the hope that, in doing so, information loss is minimal. Exploratory Projection Pursuit (PP) [4, 5] is a dimension reduction technique for forming projections of a multivariate point cloud onto subspaces spanned (usually) by the first 1, 2, or 3 coordinates, and searching for those projections that reveal the most structure. The search component of PP systems is typically some variant of a steepest descent procedure and, particularly when the search space is not well behaved, leaves open the possibility that the best projection will not be found. Genetic Algorithms (GAs) [6] hold a great deal of promise as generally applicable optimization techniques [1], and are particularly suitable for search spaces in which more traditional techniques fail. This paper provides a brief overview of both PP and GAs, and describes an implementation of PP in which a GA is used to locate the most interesting projections. The power of the GA approach is illustrated by examples in which the genetic PP algorithm is applied to datasets generated by the infamous RANDU pseudo-random number generator.

2 Exploratory Projection Pursuit

When applying PP, the analyst's goal is simply to locate interesting *structure* within the high dimensional data space. The basic paradigm for PP is:

> Assume the data is unstructured in p-space
> *REPEAT:*
> 1. locate and save directions indicating the presence of structure
> 2. return to the unstructured assumption by removing any structure found in step *1*.
>
> *UNTIL:* no significant structure can be found

For the purposes of this paper, the critical issues are related to *Step 1*, namely, how the computer can recognize when a projection is "interesting", and how such projections can be located.

2.1 Evaluating a Projection

Techniques for defining evaluation functions that can measure the degree to which a given projection reveals structure are described in detail in [7]. Due to the ease with which it can be programmed, the evaluation function used in the experiments in this paper is the simple "clottedness" index described by Friedman and Tukey in [5].

The clottedness index was designed to locate projections that simultaneously maximize both the overall "spread" and the local density of the datapoints. In order to keep the notation simple, an index designed to assess the clottedness of a *one*-dimensional projection direction α is described here. Friedman and Tukey defined the clottedness of α as:

$$C(\alpha) = s(\alpha)d(\alpha) \qquad (1)$$

Here, $s(\alpha)$ is a measure of the overall variability of the data

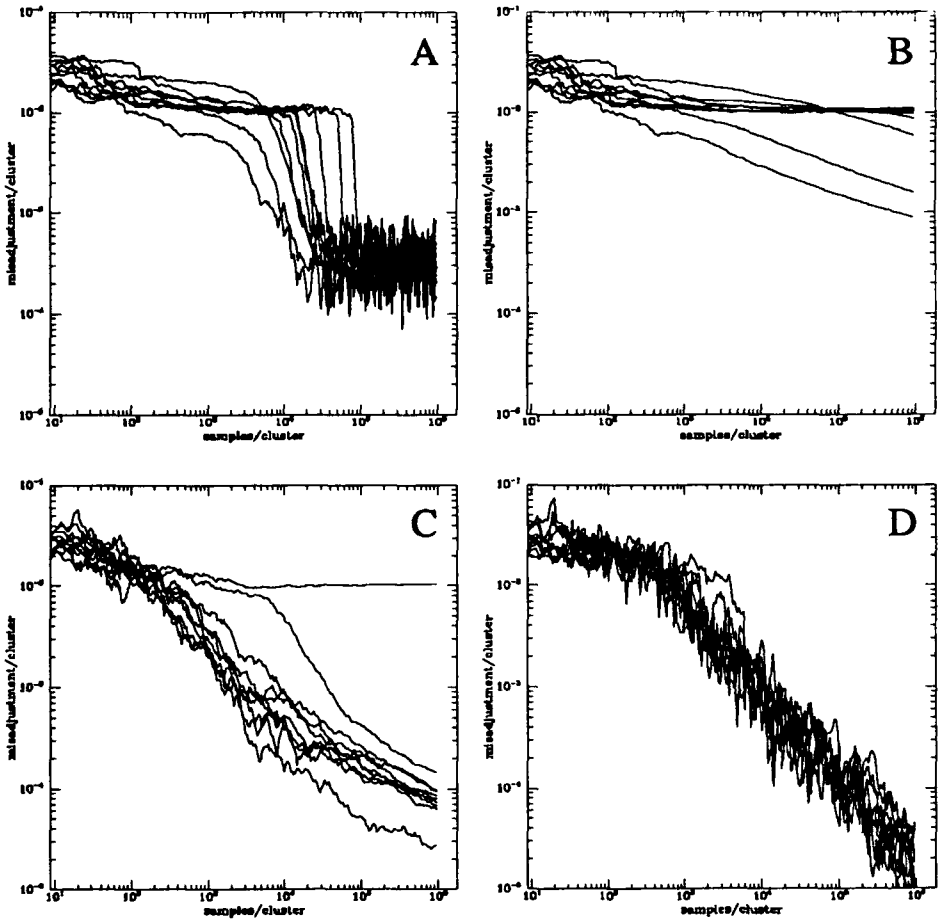


Figure 3: Comparison of 10 runs over the various schedules on the 9-means clustering task (as described under Fig. 1). The exemplars are the same for each schedule. Misadjustment is defined as $||W - W^{best}||^2$. (A) Small constant schedule (η=0.01). Note the well-defined transitions out of metastable local minima and large misadjustment late in the runs. (B) "Running average" schedule ($\eta = 1/(1+t)$). 6 out of 10 runs stick in a local minimum. The others slowly head for the global minimum. (C) Search-then-converge schedule ($\eta = 1/(1+t/4)$). All but one run head for global minimum, but at a suboptimal rate (asymptotic slope less than -1). (D) Search-then-converge schedule ($\eta = 1/(1+t/32)$). All runs head for global minimum at optimally quick rate (asymptotic slope of -1).

as projected onto direction α, and is computed as the trimmed standard deviation of the N data points, as projected onto α. Local point density is defined as

$$d(\alpha) = \sum_{i=1}^{N} \sum_{j=1}^{N} f(r_{ij}) I_{(R-r_{ij})}. \qquad (2)$$

In (2), r_{ij} is a measure of the absolute distance between any pair of points as projected onto α, and $f(r_{ij})$ is a kernel function that monotonically decreases for increasing r. A local cutoff radius, R, defines the neighborhood within which point density is measured, and an indicator function, $I(\eta)$, that evaluates to unity for $\eta > 0$, is used to identify those pairs of points no further apart that R. In words, then, the average nearness of the points along α is computed as the sum of the contributions of all pairs of points no farther apart than R, such that the closer the points, the greater their contribution to the double sum defined in (2). Locating a direction, α, that maximizes (1) therefore amounts to locating a direction that shows a configuration of well separated, dense clusters — an "interesting" projection. The extension of (1) and (2) to *two*-dimensional projections is straightforward. Data variability across the plane defined by (α, β) is defined simply as $s(\alpha)s(\beta)$ and point density is measured just as in (2) with r_{ij} defined as the Euclidean distance between pairs of points on the projection plane.

2.2 Searching for Interesting Projections

Given that any arbitrary projection can be evaluated according to its degree of interest, a mechanism must be found to locate interesting projections in the p-dimensional data space. If one imagines a two-dimensional grid encompassing all possible two-dimensional projections, then the values obtained from the clottedness function define a third dimension that is a surface over the grid of possible projections. In this context, the search for interesting projections amounts to a search for local maxima along this surface. A standard approach to problems of this sort involves choosing an initial starting point, choosing a "step" size and then varying α and β by steps until a new point, "uphill" of the previous point, is located. The application of numerical optimization procedures of this type to PP is described in some detail in [4] and [5].

2.3 Summary

PP is an effective approach for uncovering structure in multivariate data. The analyst need not specify a model in advance, estimation takes place in low-dimensional context (thus avoiding the "curse of dimensionality"), projections that reveal structure can be cheaply applied to new data, and multiple informative projections can often be found. Unfortunately, the projections located by PP can often be difficult to interpret [8]. In addition, the numerical optimization procedures might locate spurious structure [3] or fail to locate real structure. The latter possibility is the subject of this paper.

3 Motivation

Data obtained from IBM's now infamous RANDU pseudo-random number generator are often described as the kind of data to which PP techniques might be applied. The RANDU generator has the property that any three consecutively generated numbers satisfy $x_{n+2} - 6x_{n+1} + 9x_n \equiv 0 \pmod 1$, and so the triplets lie on 15 parallel planes through the unit cube. These planes are, however, visible only over a narrow "squint angle" (less than 5°) and it has been suggested that PP methods could be used to located two-dimensional projections that reveal the planes. However, Buja and Stuetzle[1], state that:

> "The RANDU planes do suggest several questions about PP. First, it seems doubtful that any version of PP would pick up the planes...even if the sample estimate of the projection index had minima at projections which show the RANDU planes, the valleys might be much too narrow to be found by *conventional* optimizers...in spite of being promoters of PP methods ourselves, we are not quite convinced that this example makes a strong case for PP. On the opposite, it might highlight some unresolved problems." (italics mine)

This statement prompted the work described in this paper—an investigation designed to ascertain whether a decidedly *unconventional* optimizer (a Genetic Algorithm) could be applied in a PP setting in order to locate two-dimensional projections that reveal the RANDU planes.

4 Genetic Algorithms

GAs typically follow a standard paradigm:

- define an encoding scheme,
- generate a starting population,
- evaluate the starting population,
- reproduce, recombine, mutate and re-evaluate until some termination criterion is met

In the context of PP, the application of GAs to the search for interesting projections involves a straightforward implementation of this paradigm. Each step is described below.

4.1 Define an Encoding Scheme

Note that a projection is usually represented algebraically as a pair[2] of linear combinations of the original p-dimensional data. For example, a starting projection might be composed of $\alpha = .32x_1 + .45x_2 - \ldots + .12x_p$ and $\beta = .56x_1 - .77x_2 + \ldots + .78x_p$. We transform each of the $2p$ parameters into

[1] On p. 486 of a discussion of Huber's Projection Pursuit review paper [2].

[2] To simplify things, we'll assume we are projecting onto the plane, so that our solutions will always be $2D$-scatterplots.

a binary string[3] of sufficient length to represent the desired range[4]. Concatenating the $2p$ bit-strings then delivers a single bit string representing the projection.

4.2 Generate a Starting Population

Instead of starting with a *single* projection and searching uphill from there, we start with *many* (hundreds, even thousands) randomly[5] selected projections. Each projection is then transformed into the bit-string representation defined above.

4.3 Evaluate the Starting Population

Associated with each bit string is an index of merit measuring how "interesting" that projection is. For our experiments, the index of merit is the clottedness function shown as equation (1). The index of merit is applied to each bit-string in the starting population.

4.4 Iterate

GAs make use of a biological metaphor in which we imagine each bit string to be a chromosome capable of combining with another chromosome and producing offspring that share the characteristics of each parent. The following five steps are repeated until some termination criterion is met:

Selection: Just as in biological evolution, natural selection is the guiding force towards adaptation. Reproduction is controlled by a biased "roulette wheel" in that the probability that a bit string will be allowed to provide a copy of itself to the next generation is proportional to its index of merit—the best projections provide multiple copies of their "genetic material" to the next generation, whereas the worst projections do not survive to the next generation at all.

Recombination: The biological metaphor is followed once again, as the collection of bit-strings form into pairs, and each member of some proportion[6] of the pairs exchanges[7] a sequence of bits with the other member. This process is called *crossover* and mimics the exchange of genetic material between biological chromosomes.

Mutation: To ensure that genetic diversity is maintained (*i.e.* premature convergence on local maxima is avoided), single bits spontaneously change state at a predefined *mutation rate*.

Restructuring: Selection, recombination and mutation serve to generate a brand new population of projections. However, unlike the starting population, the pairs of directions that form each projection are unlikely to be orthogonal. In this step, a Gram-Schmidt orthogonalization is applied to each projection to ensure that the orthogonality constraint is maintained.

[3]A special transformation called grey scale encoding is used to ensure that bit strings representing close numbers are similar.

[4]For example, $2^{11} = 2048$, so an 11-bit string can be used to represent parameters in the range $-1.024 \leq x_i \leq 1.024$.

[5]Some carefully chosen projections (*e.g.*, principal component directions) can be included in the starting population if desired.

[6]This proportion is called the *crossover rate*.

[7]An exchange point is selected at random.

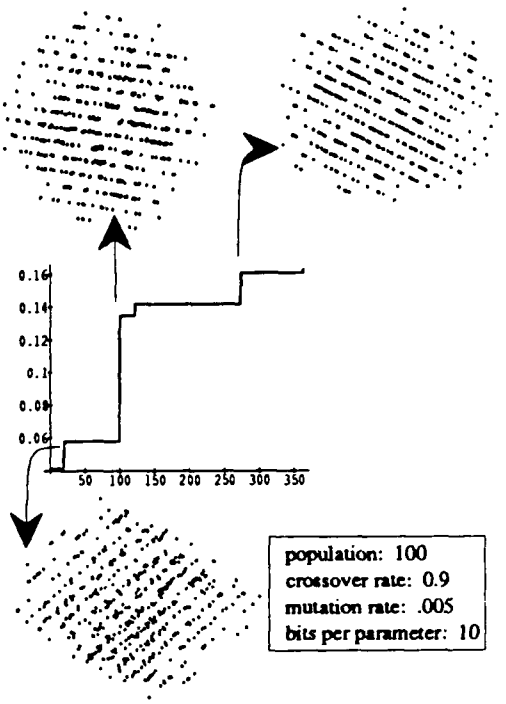


Figure 1: *Results for 3D RANDU data.*

Re-Evaluation: Each projection is evaluated and assigned an index of merit.

4.5 Summary

When used for numerical optimization, GAs differ from more traditional procedures in that they are stochastic in nature, they use an encoding of the parameter set rather than the parameters themselves, they start with a collection of points rather than a single point, and they use a simple evaluation procedure rather than computed or approximated derivatives. Because of these characteristics, GAs tend to be extremely simple to use and generate multiple, parallel search paths that tend to locate global maxima in ill-behaved (multimodal, discontinuous, noisy) search spaces where more traditional approaches fail [1]. An additional advantage of the genetic approach is that GAs can be readily implemented on fast parallel hardware when large problems must be tackled [10].

5 Results

For the first experiment, a dataset consisting of N=500 cases was generated. Each case consisted of three consecutive random variates generated by the RANDU generator. Figure 1 shows a plot with generation number on the x-axis and com-

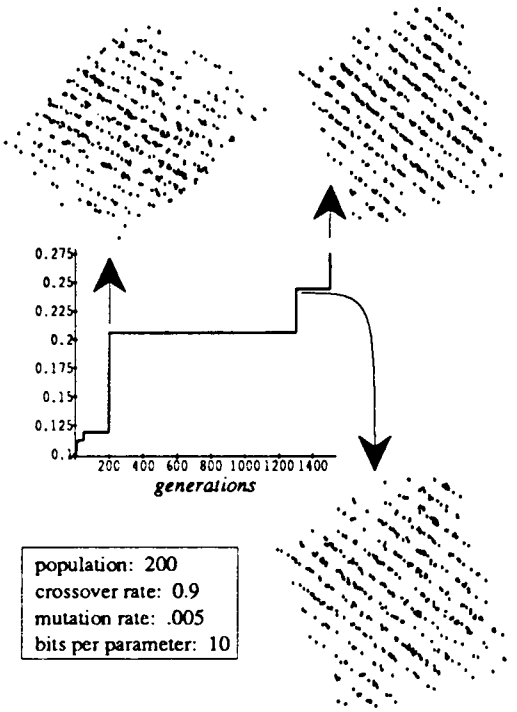


Figure 2: *Results for 4D RANDU data.*

puted clottedness index on the y-axis. The plot illustrates how the index of merit associated with the current best projection changes with time. The best projections at selected points, as well as the settings for the GA parameters[8] are also illustrated. Note that, for $3D$ data, a projection clearly revealing the RANDU planes was located after only 18 generations[9].

For the second experiment, 500 cases of $4D$ RANDU variates were generated. In addition, the size of the starting population of was increased from 100 to 200. Figure 2 illustrates that a good projection was located after 200 generations, and that subsequent generations were only slightly better.

When $5D$ RANDU data was generated, the size of the starting population was increased to 500 and the mutation rate was increased to 0.1. With these settings, the genetic PP algorithm was able to locate a good projection after 500 generations. When $6D$ data was generated, however, PP was unable to locate a good projection even after 10,000 trials[10].

[8]Guidelines for good parameter settings for optimization problems are found in [9] and were used for these exper iments.

[9]This took four minutes of CPU time on a 68040 NeXT workstation, ten seconds when N was reduced from 500 to 100.

[10]The search space is so enormous (for a squint angle of 10°, there are 10^8 2D projections!) and the number of interesting views so small that it is hard to imagine *any* optimizer doing well here.

6 Summary

The experiments described in this paper clearly demonstrate that a genetic version of PP can readily locate projections that reveal the RANDU planes—even in the exceptionally difficult $4D$ and $5D$ spaces.

GAs seem well-suited for PP not only for their good performance as general purpose optimizers, but also because they are able to generate a *collection* of interesting projections instead of the one (hopefully) optimal projection returned by the currently used search techniques. Current implementations of PP get around this problem by transforming the data to remove found structure, searching for additional structure in the transformed space, and repeating until no more structure can be found [4]. This iterative approach is not necessary when using GAs for search. Finally, when fast, parallel implementations of GAs are available, a genetic version of PP can readily be applied to very large problems.

References

[1] A.D. Bethke. *Genetic Algorithms as Function Optimizers.* PhD thesis, University of Michigan, Ann Arbor, 1981.

[2] A. Buja and W. Stuetzle. Projection pursuit. *The Annals of Statistics*, 76(376):484–490, 1981. Discussion following P.J. Huber's article.

[3] P. Diaconis. Asymptotics of graphical projection pursuit. Technical Report 14, Project ORION, Department of Statistics, Stanford University, Stanford, CA, 1982.

[4] J.H. Friedman. Exploratory projection pursuit. *The Journal of the American Statistical Association*, 82(397):249–266, 1987.

[5] J.H. Friedman and J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, c-23(9):881–889, 1974.

[6] J.H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.

[7] P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[8] S.C. Morton. Interpretable projection pursuit. In *Proceedings ASA Annual Meeting: Statistical Computing Section*, pages 63–68, Anaheim, CA, 1990.

[9] J.D. Schaffer, R.A. Caruana, L.E. Eshelman, and R. Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, 1989.

[10] R. Tanese. Distributed genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, 1989.

The Use of Genetic Algorithms in the Construction of Mixed Multilevel Orthogonal Arrays

R. B. Safadi and R. H. Wang
Olin Research Center
350 Knotter Drive
Cheshire, CT 06410

Abstract

The use of mixed multilevel orthogonal arrays in robust design has gained popularity in quality improvement areas in recent years. We have investigated the use of genetic algorithms in the construction of such arrays. This paper addresses issues encountered in formulating the problem (such as encoding and representation), as well as the results of this application. We compare this technique with simulated annealing which we published previously.

Introduction

The objective of engineering design, a major part of research and development, is to produce high quality products that meet customer requirements. Knowledge of scientific phenomena and past engineering experience with similar product designs and manufacturing processes form the basis of engineering design activity. However, a number of new decisions related to the particular product must be made regarding product specification, parameters of the product design, the process design, and parameters of the manufacturing process. A large amount of engineering effort is consumed in conducting experiments (either with hardware or by computer simulation) to generate the information needed to guide these decisions. Robust design promoted by Dr. Genichi Taguchi is an engineering methodology for improving productivity during research and development so that high-quality products can be produced quickly and at low cost (Taguchi, 1986).

Robust design draws on many ideas from statistical experimental design to plan experiments for obtaining dependable information about variables involved in making engineering decisions. Robust design makes heavy use of orthogonal arrays. Robust design adds a new dimension to statistical experimental design. It helps engineers to reduce economically the variation of a product's function in the customer's environment. Robust design also ensures that decisions found to be optimum during laboratory experiments will prove to be so in manufacturing and in customer environments.

A matrix experiment consists of a set of experiments where we change settings of the various product or process parameters we want to study from one experiment to another. Conducting matrix experiments using orthogonal arrays allows the effects of several parameters to be determined efficiently and is an important technique in robust design.

In this paper, we describe using genetic algorithms to generate mixed multilevel orthogonal arrays, without the need to resort to complex combinatorics theory. This is a continuous effort in exploring novel optimization techniques for generating general orthogonal arrays. We have reported on the use of simulated annealing for such a purpose in a previous paper (Wang and Safadi, 1990).

Genetic Algorithms: What are they and how they work.

Genes are essentially blueprints or maps that contain many segments that are responsible for the way the parts of a living species appear and function. These genes are modified during the evolutionary process by means of reproduction and mutation, where genes that are responsible for attributes in the organism that help it survive are carried over with greater probability into the next generation (survival of the fittest). Now, if one thinks of survival of the fittest as an optimization problem, with the genes mapping variables that are responsible for the value of an objective function (the fitness of the organism), then this evolution process should lead to values of these variables that optimize the objective function. A genetic algorithm is a procedure that mimics the evolution process, and uses bit-strings (strings of 1's and 0's) as genes to represent values of the independent variables, in which these bit-strings undergo the changes that genes undergo during evolution. Thus, bit-strings that represent a large (high fitness) value of the objective function will survive, eventually giving us a solution to the optimization problem. The following is a list of steps that a simple genetic algorithm could follow (Goldberg, 1989)(also see Figure 1):

1. Mapping of the variables into genes (encoding).
2. Marking of the genes with probabilities for their participation in the reproduction process based on the value they cause the objective function to have.
3. Reproduction (Collection of a gene pool for mating).
4. Mating of the genes (crossover).
5. Mutation.

In order to illustrate these steps, we shall formulate the problem of mixed-multilevel orthogonal array generation, encode it in the aforementioned binary form (genes), then apply the algorithm.

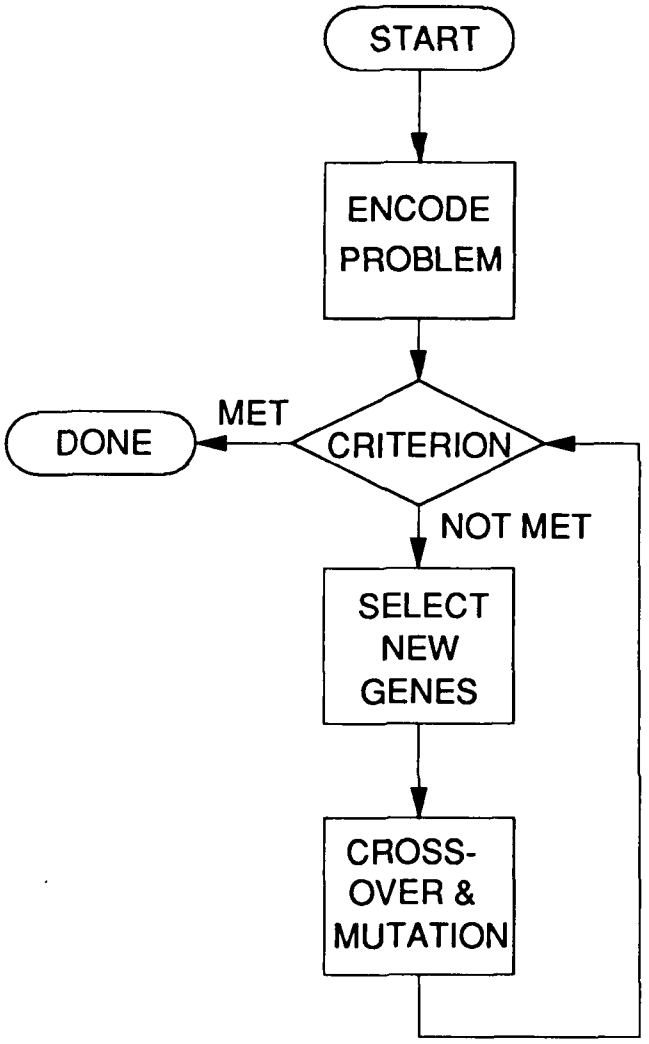


Figure 1. A general Genetic Algorithm

Orthogonal array generation

The problem statement for the generation of an orthogonal array (OA) is as follows:
Given F_1 factors at L_1 levels, F_2 factors at L_2 levels, L_2 levels, ... , F_N factors at L_N levels, generate the orthogonal (balanced) array $L_1^{F_1} \times L_2^{F_2} \times \ldots \times L_N^{F_N}$ Which is a matrix consisting of columns that contain the various levels for each factor. The orthogonality requirement is met if and only if the following is true:

1. Within each column, there must be an equal number of occurrences of each level setting.
2. Rows having a particular number of level settings in a certain column, must have an equal number of all other level settings in the rest of the columns.
3. The number of rows in the matrix should be the minimum that achieves the above conditions.

To generate the OA, we first generate an unbalanced array, then use simulated annealing to balance it. To generate the initial unbalanced array, the following steps are taken:

1. Satisfy condition 3 above. The minimum number of rows N_R is the lowest common multiplier of the following list:
   $L_i \text{ for } i = 1,n; L_i \cdot L_j \text{ for } i,j = 1,N; L_i \cdot L_i \text{ for } i = 1,N \text{ if } F_i > 1$
2. Find the number of occurrences of each level for each particular column (condition 1 above). If the number of levels in that columns is L_i then the number of occurrences for each of these levels in that column is:
   $$N_R/L_i$$
3. Fill the columns with the appropriate number of levels as calculated above.

This would give us the initial unbalanced matrix. To illustrate, consider the array $3^1 \times 2^4$, which gives the following initial (unbalanced) matrix shown in Figure 2. The minimum number of rows is lcm(2,3,4,6)=12. The balanced matrix is given in Figure 3.

To speed up the computation time, we balance the array one column at a time. First, the first column is fixed, the algorithm is performed on the second column to balance it with the first using condition 3 above. Once this column is balanced, the algorithm is performed on the next while trying to balance it with the previous two columns. This is repeated until the whole array is balanced.

1 1 1 1 1
2 2 2 2 2
3 1 1 1 1
1 2 2 2 2
2 1 1 1 1
3 2 2 2 2
1 1 1 1 1
2 2 2 2 2
3 1 1 1 1
1 2 2 2 2
2 1 1 1 1
3 2 2 2 2

Figure 2. The unbalanced $3^1 \times 2^4$ matrix

1 1 1 2 1
1 1 2 1 1
1 2 1 2 2
1 2 2 1 2
2 1 1 1 2
2 1 2 2 2
2 2 1 2 1
2 2 2 1 1
3 1 1 1 1
3 1 2 2 2
3 2 1 1 2
3 2 2 2 1

Figure 3. The balanced $3^1 \times 2^4$ matrix

The Algorithm

We will now illustrate the five steps of the algorithm on one of the columns of an array.

1. Encoding: An encoding scheme should insure that all possibilities for a column configuration can be represented by it. We designed the encoding scheme illustrated in Figure 4. The encoding gene in this case specifies a series of switching operations to be performed on a fixed initial column in order to arrive at the column that the gene encodes. In Figure 4, we have a column of 6 rows, and an encoding gene with 15 "chromosomes", each of which represents a combination of two rows in the column. To arrive at the column that the gene is actually encoding, we switch the rows of the column whose corresponding chromosomes in the encoding gene have a value of 1. For example, chromosome #1 has a value of 0, so rows 1 and 2 will in the column will not be switched. Chromosome #2, however, has a value of 1, so rows 1 and 3 will be switched. This process is repeated for all row combinations.

2. Assigning genes probabilities for reproduction: Table 1 shows a list of genes and their corresponding fitness, and based on that fitness, a segment of real numbers between 0 and 1. The idea is that the larger the fitness, the larger the corresponding segment, and the larger the probability (Figure 3) that a random number between 0 and 1 will fall in that segment, which is the way the genes are chosen for mating and reproduction. This method insures higher probabilities of reproduction for genes with higher fitness.

3. Selection of the mating pool: A random number is generated, and the gene that corresponds to the segment into which this random number fits is selected to be a member of the mating pool. This is repeated as many times as the number of genes in the initial gene pool.

4. Mating and crossover: After building the mating pool, genes from this pool are paired randomly, and crossover will take place between them. In other words, a segments of the same (random) number of chromosomes are chosen from random locations in the mating genes and exchanged (Figure 4).

5. Mutation: The function of mutation is to help prevent the algorithm from being trapped in local minima. Mutation happens infrequently and to a random chromosome in a randomly selected gene.

Table 2. Gene Fitness and Segment assignments

Gene #	Fitness	Segment	Probability of Reproduction (%)
1	4	0.00 - 0.04	4.0
2	10	0.04 - 0.14	10.1
3	15	0.14 - 0.29	15.2
4	20	0.29 - 0.49	20.2
5	50	0.49 - 1.00	50.5

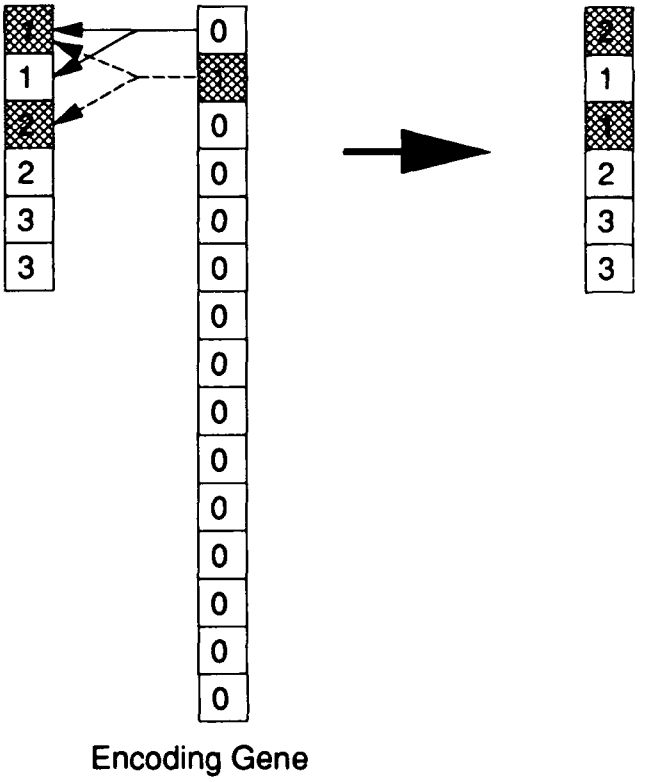


Figure 2. Gene Encoding.

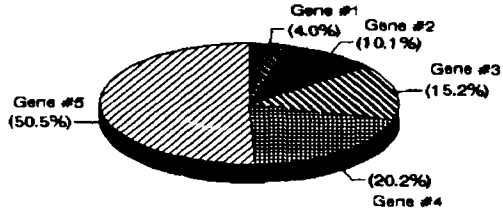


Figure 3. Gene Mating probabilities.

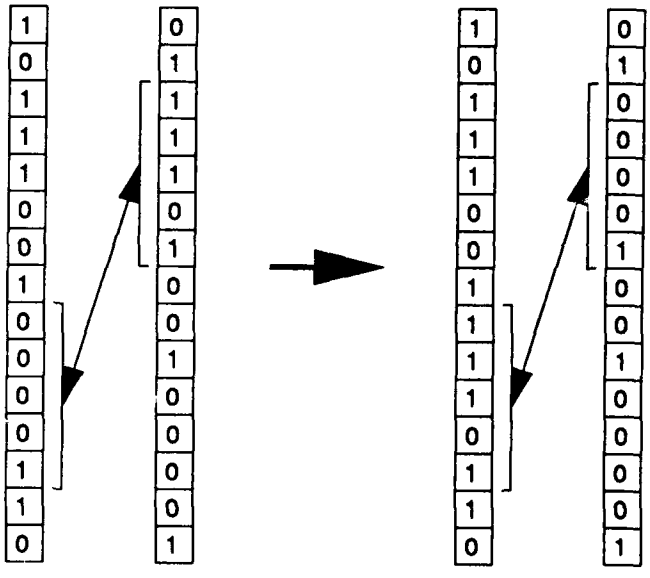


Figure 4. Crossover

Conclusion

We have used a genetic algorithm to construct mixed multilevel orthogonal arrays. These arrays are quite useful in robust design and quality improvement projects. The use of this novel search and optimization technique allows us to generate these arrays without resorting to complex combinatorial techniques which can also be restrictive.

REFERENCES

1. Taguchi, G., Introduction to Quality Engineering, Asian Productivity Organization, 1986.
2. Wang, R. H. and Safadi, R. B., "Generating Mixed Multilevel Orthogonal Arrays by Simulated Annealing", in Proceedings of Interface '90, 1990.
3. Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley: Reading, Massachusetts, 1989.

The Use of LEGO Bricks to Construct Solid 3-Dimensional Dose-Response Surfaces

William R. Greco[1]
Pharmacometrics Laboratory
Roswell Park Cancer Institute
Buffalo, New York 14263

Abstract

Three-dimensional graphs of mathematical/statistical models are useful for the understanding of many phenomena. However, even with sophisticated expensive computer hardware and software, realistic rendering of 3-dimensional graphs is still a difficult task. For scientists with a limited budget, it is close to impossible. A simple inexpensive approach is to construct 3-D surfaces with LEGO bricks. Steps include: (a) simulate data (calculate outputs, i.e., z values from a design matrix of x, y inputs) for the surface via a favorite computing language or package; (b) roundoff data to the resolution of individual bricks, and tabulate data on sheets of paper, one sheet for each brick layer; (c) construct a wooden platform with axis tic marks and labels; and (d) construct the 3-D surface with bricks. Examples of useful models which have been constructed include ones of: (a) synergism for two anticancer drugs; (b) antagonism for two anticancer drugs; (c) a composite generalized nonlinear model consisting of a logistic dose-response structural model with a binomial data variation model; and (d) a likelihood function associated with the fitting of a monoexponential pharmacokinetic model to data with two estimable parameters, illustrating profile likelihood. These 3-D LEGO models are useful for (a) studying the shape of 3-D functions; (b) gaining insight into physical phenomena; (c) explaining concepts in statistical analysis approaches; and (d) designing experiments.

Introduction

The visualization of 3-dimensional (3-D) mathematical/statistical models is important in many branches of applied and theoretical mathematics and statistics. Such visualization is very difficult however, even with expensive computer graphics capabilities. It is especially difficult to represent 3-dimensional surfaces as 2-dimensional (2-D) static images. Tricks such as shading, shadowing, motion, stereoscopic hardware and holography may provide some assistance in 3-D visualization, but none of these approaches are ideal, and most are expensive.

Driven by the need to intimately understand 3-D concentration-effect surfaces for my research in Pharmacometrics, and constrained by the limits of a small budget, I constructed four 3-D graphs with LEGO and LEGO-compatible building blocks. A picture of me with three of these models is shown in Figure 1. They are of a suitable size for classroom teaching, one-on-one tutoring, and contemplative thinking. When carefully packed in boxes with styrofoam beads, they are easily transported by car and/or plane. The approach which I used to construct these models is quite general, and should be applicable and useful to a wide variety of mathematical/statistical topics for both research and teaching purposes. This article describes the construction and use of these models.

[1]Supported by NCI grants CA46732, CA16056, and CA21071.

Figure 1. A proud man and his models.

Methods

Each 3-D mathematical/statistical function was first simulated with custom FORTRAN programs. The 3-D array of points was then printed out on a 2-D table with the values of the X and Y variables listed along the top and leftside of the table, and with the values of the Z (height) variable listed in the cells of the table. The Z values were rounded to the nearest LEGO brick.

Using the 2-D table as a guide, each of the models was constructed on a standard 10 in by 10 in LEGO base . The heights of the models varied from 11 to 22 standard LEGO bricks. (Each brick is 0.375 in high.) Models were constructed, one layer at a time of a uniform color, with each adjacent layer (or set of layers) being a different color. Black bricks were often use to highlight important contours. The judicious use of color is a great aid to the thorough understanding of the 3-D model by the viewer.

For each model, a wood base, 13 in by 13 in by 4.5 in was constructed, and covered with black laminate. (A simpler base made from one piece of 0.5 inch pressed board or plywood would be adequate.) Axis labels with tic marks were made to the proper scale, were laminated, and then glued to the wood bases. Finally, the LEGO models were glued onto the bases.

Description of Models

A. Synergism. Figure 2 shows a concentration-effect surface for two drugs, DDATHF (5,10-dideazatetrahydrofolate) and trimetrexate, and a response which is the growth of cells in a cell culture assay, expressed as a percent of control growth. The details for this *in vitro* cancer chemotherapy experiment are described elsewhere (Greco et al, 1990). Equation 1 was fit to the data in Figure 1 with iteratively reweighted nonlinear least squares. The best fit surface, shown in Figure 1, was constructed with SAS/GRAPH (SAS Institute, 1990). The mathematical/statistical details of the nature, origin and use of Equation 1 have been published elsewhere (Greco et al, 1990; Greco and Lawrence, 1988; Greco, 1989; Syracuse and Greco, 1986).

Briefly, Equation 1 allows the slopes of the concentration-effect curves for the two drugs to be unequal. A convention used in Equation 1 is that as drug concentration(s) increases, the measured response decreases; the slope parameter, m, is negative. The output, E , is the measurement from the cell growth assay; and the inputs are $[TMTX]$, $[DDATHF]$, the respective concentrations of TMTX and DDATHF. The seven estimable parameters include: $Econ$, the control or maximum response at 0 drug concentration; B, the extrapolated background response at infinite drug concentration; $IC_{50,TMTX}$, $IC_{50,DDATHF}$, median effective concentrations of TMTX, DDATHF respectively; m_{TMTX}, m_{DDATHF}, slope parameters for TMTX, DDATHF respectively; and α, the synergism-antagonism parameter. When α is positive synergism is

indicated, when α is negative antagonism is indicated, and when α is 0, no interaction or additivity is indicated, The magnitude of α is algebraically related to the degree of bowing of isobols (contours cut through the surface at specific response levels); a larger α will result in a larger degree of bowing.

The specific surface shown in Figure 2, the LEGO model in the lower left of Figure 1, the LEGO model in Figure 3, and on the computer screen in Figure 3, is the best fit surface for data from an experiment in which cells were exposed to 2 μM folic acid in addition to the drugs, trimetrexate and DDATHF. The parameter estimates were: $Econ$ = 0.787 response units; B = 0.0213 response units; $IC_{50,DDATHF}$ = 3.91 nM; m_{DDATHF} = -3.91; $IC_{50,TMTX}$ = 16.7 nM; m_{TMTX} = -2.16, and α = 4.68.

$$1 = \frac{[TMTX]}{IC_{50,TMTX}\left(\frac{E-B}{Econ-E}\right)^{1/m_{TMTX}}} + \frac{[DDATHF]}{IC_{50,DDATHF}\left(\frac{E-B}{Econ-E}\right)^{1/m_{DDATHF}}} + \frac{\alpha[TMTX][DDATHF]}{IC_{50,TMTX}\ IC_{50,DDATHF}\left(\frac{E-B}{Econ-E}\right)^{1/2m_{TMTX}}\ \left(\frac{E-B}{Econ-E}\right)^{1/2m_{DDATHF}}} \quad (1)$$

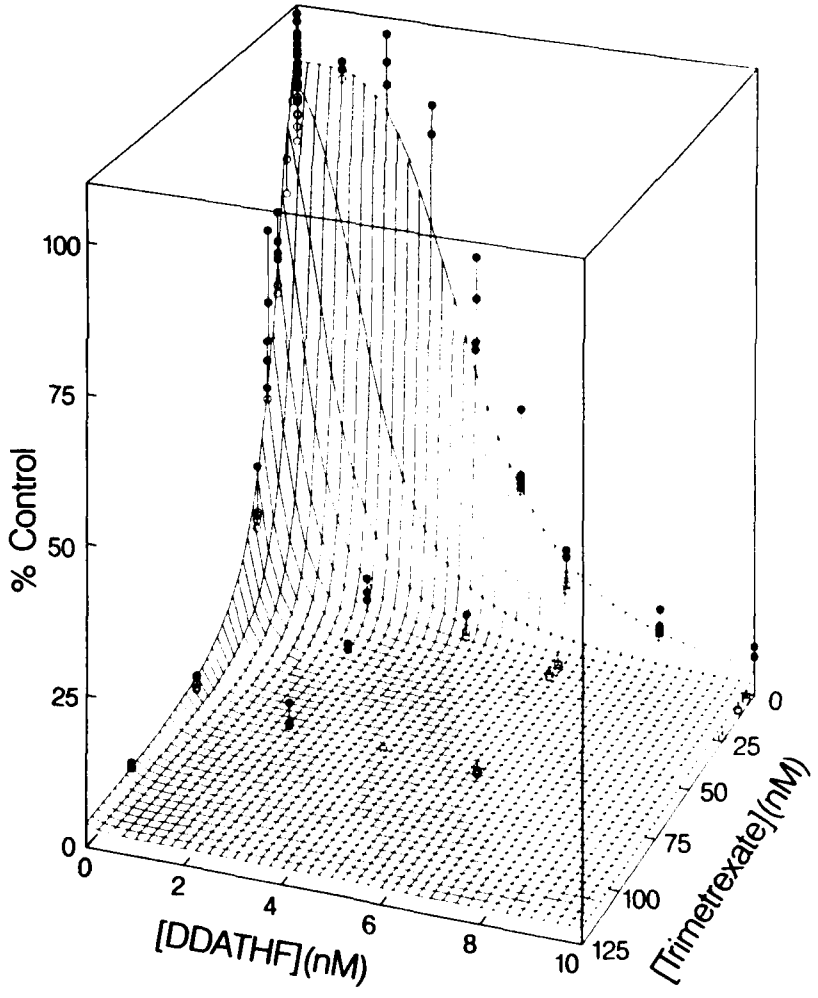


Figure 2. A 3-D surface of Equation 1 with parameter values listed in the text, constructed with SAS/GRAPH.

Figure 3. Comparison of 3-D LEGO model with the same surface generated on a computer screen.

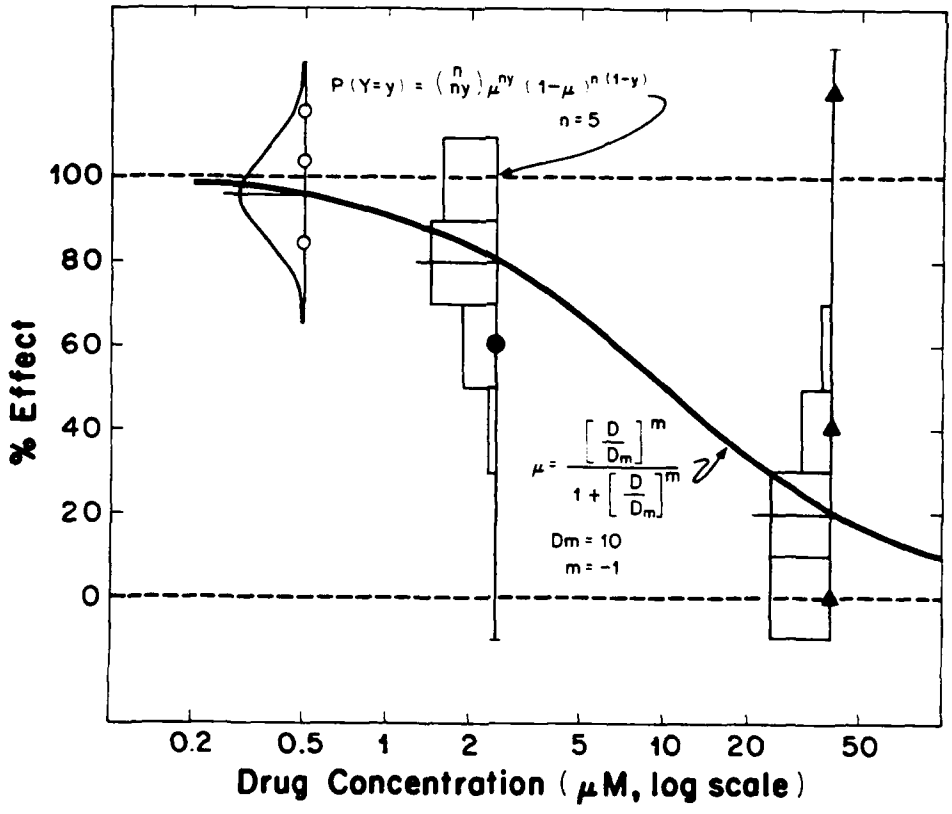


Figure 4. General scheme for the dissection of a generalized nonlinear model into random and structural components for a concentration-effect curve for a single drug.

B. Anatagonism. The LEGO model on the lower right of Figure 1 is for drug antagonism. The surface was simulated using the generic form of Equation 1, for which drug 1 and drug 2 are designated as D_1 and D_2, and with $Econ$ = 100; $IC_{50,1}$ = 1 μM; $IC_{50,2}$ = 1 μM; m_1 = -2; m_2 = -1; α = -1; and B = 0. Note the saddle shape of the surface.

C. Generalized Nonlinear Modeling. Figure 4 is a pictoral dissection of a generalized nonlinear model (McCullagh and Nelder, 1983) into structural and random components. The heavy black sigmoid curve is the structural component, and was simulated from the lower equation in Figure 4. In this equation, μ is the expected (mean) response; D is the concentration of drug; Dm is the median effective dose (same as IC_{50}); and m is the same slope parameter as in Equation 1. For the simulated curve, Dm = 10 and m = -1. The three distributions shown in Figure 4, the normal, a modified binomial and a modified Poisson distribution, represent possible random components of a complete generalized nonlinear model. For the modified binomial distribution, the equation is shown in the upper portion of Figure 4, where $Y = k/n$, and k is the number of successes, n is the number of tries; mean = μ; variance = μ(1-μ). The graph of the binomial distribution in Figure 4 was simulated at D = 2.5 μM, with μ = 0.80 and n = 5. For a complete description of the application of these generalized nonlinear models to concentration-effect data, see Greco and Lawrence, 1988 and Greco, 1989.

The LEGO model being held in Figure 1 is a 3-D representation of Figure 4, with the same sigmoid-logistic structural model, the same modified binomial model, and with the same parameters. However, in the LEGO model, the binomial distribution is shown all along the length of the sigmoid curve.

D. Likelihood Surfaces. Figure 5 shows a 3-D negative log likelihood surface for the following problem:

Equation 2 is a standard monoexponential 1-compartment pharmacokinetic structural model with bolus intravenous injection of drug.

$$Cp = [D/V] \exp(-[Cl/V]\, t) \qquad (2)$$

Where: Cp is the concentration of a drug in plasma; D is the administered dose of drug; t is the time that a plasma sample is drawn for a plasma drug level measurement; V is the volume of distribution (a parameter); and Cl is the plasma clearance of the drug (a parameter).

A set of hypothetical data is as follows (D = 1 μmole):

Data:	time (min)	Cp (μM)
	1	0.904
	5	0.607
	10	0.500
	20	0.135

The data were fit with the nonlinear regression software package, PCNONLIN (Statistical Consultants Inc., 1989), in which the weight, w_i, for each point was equal to the reciprocal of the square of the predicted plasma concentration, with the sum of the weights forced to equal N, the number of data points.

$$\text{Weight: } \frac{1}{\hat{Cp}^2} \qquad \sum_{i=1}^{N} w_i = N$$

The parameter estimates at the optimum (minimum of the objective function) were $\hat{V}$ = 0.896 ± 0.15 (S.E.) L, and $\hat{Cl}$ = 0.0909 ± 0.0093 (S.E.) L/min.

The objective function, O, shown in the LEGO model of Figure 5 is defined in Equation 3.

$$O = \sum_{j=1}^{N} \frac{w_j(Cp_j - \hat{Cp}_j)^2}{s^2} \qquad (3)$$

$$\text{where: } s^2 = \frac{\sum_{j=1}^{N} w_j(Cp_j - \hat{Cp}_j)^2}{N - P} \qquad (4)$$

at the optimum.

Thus, the 3-D surface in Figure 5 has $\hat{V}$ as the X-axis, $\hat{Cl}$ as the Y-axis, and O as the Z-axis. Note the irregular shape of the negative log likelihood bowl. Asymptotic 95% confidence intervals are usually calculated based upon the assumption that the negative log likelihood

surface is a regularly-shaped ellipsoid (football shape) near the minimum. The optimal values for the parameter estimates are at the lowest point in the negative log likelihood bowl. At this point,
$O_{opt} = N - P = 2$ (from combining Equations 3 and 4). The critical value, $F_{(0.95,1,2)} = 18.513$; and thus at $O = 20.513$ (2 + 18.513), the black top layer of the LEGO model, one can calculate 95% confidence intervals for the parameters via profile likelihood. The usual asymptotic 95% confidence intervals for V and CL were: 0.263 to 1.53 and 0.0508 to 0.131 respectively. The profile likelihood 95% confidence intervals for V and Cl were 0.015 to 1.68 and 0.005 to 0.198 respectively.

Figure 5. LEGO model of a negative log likelihood surface.

Summary

Lego bricks provide an inexpensive, easy-to-use medium for constructing useful 3-D models of mathematical and statistical functions. Many high level concepts in the field of Statistics can be communicated to non-mathematically trained scientists (as well as to more-mathematically trained scientists) with the use of these tangible models. This statistical ideas presented in this short article, could have been much more clearly and succinctly presented if the reader could have seen and touched the LEGO models. In a paradoxical sense, this article, filled with symbols, numbers and equations, is the antithesis of the point which I would like to emphasize: the great potential for excellent graphical models to improve both teaching and research involving statistical applications and theory.

Acknowledgements

I would like to thank Dr. John Nash for his encouragement and the for awarding the Nash Information Services Inc. Prize for Visual Presentation for this work. I would also like to thank Mr. Peter Eio, President of LEGO Systems Inc., for a generous gift of LEGO bricks.

References

Greco WR. Importance of the structural component of generalized nonlinear models for joint drug action. Proc. Biopharm. Sect. Am. Stat. Assoc. 183-188, 1989.

Greco WR and Lawrence DL. Assessment of the degree of drug interaction where the response variable is discrete. Proc. Biopharm. Sect. Am. Stat. Assoc. 226-231, 1988.

Greco WR, Park HS, Rustum YM. Application of a new approach for the quantitation of drug synergism to the combination of cis-diamminedichloroplatinum and 1-β-D-arabinofuranosylcytosine. Cancer Res. **50**:5318-5327,1990.

Greco WR, Parsons JC, Gaumont Y, and Kisliuk RL. Quantitation of the folic acid enhancement of antifolate synergism. Suppl. J. Cancer Res. Clin. Oncol. **116**:452, 1990.

McCullagh P and Nelder JA. **Generalized Linear Models.** London: Chapman and Hall, Ltd, 1983.

Statistical Consultants Inc. **PCNONLIN User's Guide.** Version 3., 1989.

Syracuse KC and Greco WR. Comparison between the method of Chou and Talalay and a new method for the assessment of the combined effects of drugs: a Monte-Carlo simulation study. Proc. Biopharm. Sect. Am. Stat. Assoc. 127-132, 1986

Optimal Airliner Parking Configurations

Michael J. Healy
PO Box 24346
MS 7L-22
Seattle, WA 98124-0346
mjhealy@atc.boeing.com

The year-to-year growth in airline passenger traffic, accompanied by the recent introduction of wide-bodied airliners to handle the demand, has caused increasingly severe parking problems for airliners at terminal gates. An algorithm has been developed to optimize airplane parking configurations. The algorithm is based upon dynamic programming, and determines a parking configuration which maximizes the utilization of airliners in a given fleet mix. It solves for a string of tokens which represent airplane parking maneuver envelopes. The envelopes are charac- terized by a discrete collection of possible combinations of airplane type, airline ground footprint, parking angle, and maneuver in and out of terminal loading configuration. /The maneuvers are predetermined to allow independent access to each terminal loading zone. A solution is constrained to obey width and parking obstacle constraints in multiple, overlapping loading zones, including corners. The complexity is a low-order polynomial in the linear extent of a contiguous string of loading zones.

1 Introduction

The year-to-year growth in airline passenger traffic, accompanied by the recent introduction of wide-bodied airliners to handle the demand, has caused increasingly severe parking problems for airliners in terminal loading zones. The competition for limited space within loading zones, makes it imperative to find solutions that conserve limited parking space while obeying FAA airplane clearance requirements. Space-saving solutions that allow more and larger airliners to occupy a given loading zone simultaneously can be of such importance that designers will modify the terminal building structure to acommodate them.

The algorithm presented here addresses this problem by identifying space- saving solutions for airplane parking within loading zones (Fig. 1). Constraints on geometry can be accounted for by applying simple rules to the two-dimensional geometry of airplanes, their clearance requirements, loading zones, and obstructions to airplane taxiing and independent access to a parking space. Once the constraints are analyzed and converted into simple geometric quantities, a dynamic programming algorithm [Gar72] can be applied to find solutions. The solutions are expressed in terms of the number of airplanes of each of one or more specified types (Boeing 767-200, 757-100, McDonnel-Douglas DC-10, etc.) that can be parked in a given loading zone simultaneously. For example, an airline with a fleet mix of DC-10s and 757s might wish to park as many DC-10s as possible, and then to park as many 757s as possible within any remaining space. Smaller airliners, such as 737s or DC-9s (depending on the fleet mix of the airline involved) might then occupy any remaining parcels of loading zone space if there is sufficient room.

In addressing this problem, several factors must be taken into account. Normally, loading zones are constricted on at least one side by the terminal wall, and on another side by a taxi lane (Fig. 2), from which airliners enter and leave the loading zone. Entrance and egress occurs along a prescribed, well-marked path in both the taxi lane and the loading zone. FAA clearance requirements, such as wingtip clearances, must be obeyed at all times. Thus, as in the case shown for La Guardia Airport in Fig. 1, an airplane can taxi along a progressively narrower taxi lane only until the wingtip clearance points of its *clearance envelope* touch the limit lines on either side of the taxi lane.

This limits the range of parking solutions available to an airplane of a given type in a given loading zone, quite apart from any restrictions imposed by the geometry of the loading zone itself. Normally, airliners pivot sharply and enter the loading zone from the taxi lane at close to a 90 degree angle to the limit line, thus approaching the terminal wall head-on (Fig. 3). For various reasons, however, they may come to rest in a loading configuration which forms an acute angle with the terminal wall. An example of this occurrence is shown in Fig. 2.

Added to the considerations already presented is the variation in airline requirements. These include airliner ground footprint requirements, which are specific to each airline for each airplane type in its fleet. These requirements consist of the space occupied by baggage handling and other equipment surrounding an airliner in loading configuration.

2 Analysis

The airplane parking optimization algorithm applies a form of dynamic programming to maximize the value of airplanes parked while obeying limits on total loading zone length. An example was stated in Section 1, in which the number of DC-10s parked simultaneously is to be maximized, followed by 757s. In dynamic programming, a solution is divided into a sequence of stages, and an optimality principle is applied at each stage according to a monotonicity assumption: If the cumulative value of a partial solution at any stage is a monotonically increasing function of value increments at each stage, and the cumulative cost of resources to obtain that value obeys a similar relationship to the partial cost at each stage, then the optimality principle allows one to avoid explicitly examining every possible combination of alternatives for all stages: the alternatives are narrowed down at each stage so that in succeeding search stages, a relatively small set of cumulative, partial solutions are maintained. These consist of only those whose values (costs) would contribute toward the total value (cost) in an optimal solution.

Before the parking algorithm can find solutions, however, the airplane clearances and loading zone geometry must be analyzed to derive quantities which can be manipulated by the dynamic programming algorithm. Hence, the solution method consists of two parts. The first part is a geometric analysis phase, in which a library of simple *interaction rectangles* and loading zone limits is created from the complexities of airport geometry, clearance requirements, independent access, airplane type, and airline ground equipment. In creating the library, it is often prudent to assume several different loading configuration angles relative to the terminal wall, to allow flexibility in finding solutions that allow an optimum mix of airplanes to be parked while accounting for obstructions and space limitations within a given loading zone. An intermediate result of the geometric analysis phase is a library of *airplane maneuver envelopes*, each envelope being defined by a particular combination of the geometry, clearances, and loading configurations analyzed (see Figs. 2 and 3).

In the second part of the solution method, a dynamic programming algorithm derives solutions by finding linear arrangements of the interaction rectangles within the extent of a loading zone. The loading zone can cansist of multiple bands, with corner slots at the corners of a terminal wall (Fig. 4). Each band and corner slot consists of overlapping bands which define the allowed regions for parking airplanes of different types in different loading configurations. These allowed regions depend upon taxi lane obstructions and parking strategy used, as previously discussed. Alternatively, the loading zone can be curved (Fig. 5).

User-specified fleet mix assumptions can be applied in solving the dynamic program: for example, the number of DC-10s available at a given time may be limited by scheduling constraints at the airport for which a solution is being sought.

The individual rules for the geometric analysis

of airplane parking maneuvers in loading zones are simple, but applying them is complicated by their interaction and the need to provide flexibility in solving the optimization scenarios. As mentioned before, alternative loading configurations must be considered within a given loading zone for each airplane that could conceivably park in the zone. A loading zone may have multiple limits because of, among other things, taxiing constraints as illustrated in Fig. 2: larger airplanes, or loading configurations of even medium-sized airplanes that require perpendicular entry into the loading zone, do not apply beyond the point at which the taxi lane is too narrow to allow them.

The geometric analysis phase is further complicated by the following consideration: The cost associated with a trial parking solution is the amount of space that it occupies in the loading zone. Suppose that a trial solution at stage N of the dynamic programming algorithm were modeled as a string of N maneuver envelopes, beginning at a specified end of the loading zone under consideration (see Fig. 6, in which the string is actually the final optimum solution derived for a particular loading zone at La Guardia airport). The envelopes are packed as closely as possible within the string, which obeys the limit constraints of all overlapping bands within the loading zone. The complication is that if maneuver envelopes define the dynamic programming stages, the space-utilization cost of a string cannot be measured by summing the partial costs of the N solution stages (the maneuver envelopes forming the string). This is not a well-defined quantity, since the space an envelope occupies really depends upon its interaction with its neighbors in the string. This problem is yet further complicated by the subdivision of the loading zone into overlapping bands, each with its own limits on airplane maneuver envelopes (some are excluded altogether from a given band, as mentioned).

The quantities that are actually measured are the lengths of *interaction rectangles* (Figs. 7 and 8). These rectangles are defined by dividing each maneuver envelope along a selected "midline", and then deriving the length of each interacting pair of envelopes, measured from midline to midline, as shown in Fig. 7. A solution string then consists of interaction rectangles, joined so that the maneuver envelopes from which they were derived are reconstituted. Thus, the interaction rectangles have labelled ends corresponding to the envelopes from which they were derived. In Figs. 7 and 8, the end-labels are shown as simply A, B, and C. In actuality, they are indexed to indicate the combinations of airline, airplane, loading configuration, and parking strategy by which their constituent maneuver envelopes were derived. In any trial solution string, the constraint that end-labels must match is applied, thereby reconstituting the maneuver envelopes (special rectangles with labels matching those at the ends complete each end of the string).

3 Solution Method

The dynamic programming algorithm solves the following general problem. Let the interaction rectangles derived in the geometric analysis phase be $R_{11}, R_{12}, \ldots, R_{MN}$, where i and j in R_{ij} are indices denoting the maneuver envelopes. Recall that each maneuver envelope is defined in terms of airline ground clearance requirements, FAA-specified airplane clearances, airplane type, and the geometry of different loading zones, taxi lanes, obstructions, and the possible loading configurations and parking maneuvers under consideration. Also, some of the indices represent the geometry of one end of a loading zone, serving as a starting point for solution strings. Thus, M and N differ only in that M provides for N maneuver envelopes plus the required end-geometry configurations, assuming solution strings are formed from left to right. Let $\ell(R_{ij})$ denote the length of the interaction rectangle R_{ij}. Let $A_1, A_2, \ldots, A_T$ denote the sets of indices of maneuver envelopes which correspond to each of the T airplane types, and let $I_t (t = 1, 2, \ldots, T)$ be the

corresponding number of copies of airplanes of type t (Boeing 767-200, for example) available. Finally, let $L_1, L_2, \ldots, L_K$ be sets of ordered pairs (i, j) of indices corresponding to the interaction rectangles that are allowed in each of the K overlapping bands and corner slots in the loading zone. These sets are determined from the original maneuver envelope limits during the geometric analysis phase. Let ℓ_k and u_k be the lower and upper bounds on interaction rectangles with index pairs in L_k.

Let $X^S = R_{i_0 i_1} R_{i_1 i_2} \ldots R_{i_{S-1} i_S}$, where each $X_{i_{k-1} i_k} = R_{pq}$ for some indices $1 \leq p \leq M$ and $1 \leq q \leq N (1 \leq i_{k-1}, i_k \leq S)$, denote a trial solution string at stage S of the algorithm. Let $|X_t^S|$ denote the number of airplanes of type t in the string X^S. The *cost* of the string X^S is its total length,

$$c(X^S) = \sum_{j=1}^{S} \ell(R_{i_{j-1} i_j}). \qquad (1)$$

Then at stage S, each partial trial solution string X^S must obey the following constraints:

$$|X_t^S| \leq I_t \quad (t = 1, 2, \ldots, T) \qquad (2)$$

and

$$(i_{S-1}, i_S) \in L_k$$

whenever

$$\ell_k \leq c(X^S) \leq u_k \quad (k = 1, 2, \ldots, K). \quad (3)$$

A final solution string must maximize the desired quantities (number of DC-10s and 757s, for example) subject to (2)–(3).

The dynamic programming algorithm avoids enumerating all possible partial strings. At stage S, let two trial solution strings be $X^S = R_{i_0 i_1} R_{i_1 i_2} \ldots R_{i_{S-1} i_S}$ and $Y^S = R_{i'_0 i'_1} R_{i'_1 i'_2} \ldots R_{i'_{S-1} i'_S}$. If the two strings X^S and Y^S have the same value (contain the same number of airplanes of each type to be maximized, for example) and if the end maneuver envelope indices are the same in both strings, $p = i_S = i'_S$, then the only advantage one string can possess over the other in determining an optimum is lower cost. This follows because they possess the same value, and because at stage $S + 1$, both strings are extended by adding interaction rectangles of the form $R_{p,q}$ for all q such that (2)–(3) are satisfied with S replaced by $S + 1$. That is, both strings offer the same possibilities for adding interaction rectangles at the next stage. The algorithm exploits this fact by eliminating either X^S or Y^S, depending on which possesses the higher cost. Elimination on pairs is performed repeatedly until the stage S solution strings contain no such pairs. Thus, only the lowest-cost strings with unique properties to contribute to a final solution are kept.

If there are U_S strings remaining following the elimination at stage S, the work involved in comparing and eliminating pairs at stage $S + 1$ is at most $U_S N$ (recall that there are N maneuver envelopes worthy of consideration: the other $M - N$ account for string start geometry). Normally, the number of different string values to consider depends upon the number of at most two airplane types in a string. Thus, $U_1 \leq N$ for a one-rectangle string $X^1 = R_{i_0 i_1}$, and $U_2 \leq N \cdot [(2 + 1)(2 + 2)/2]$ since there are at most $[(2 + 1)(2 + 2)/2]$ sums of two values having a maximum of 2 occurrences between them (either airplane type may occur 0, 1, or 2 times) in a string of length two. Similarly, there are $[(S + 1)(S + 2)/2]$ possible sums of two values having a maximum of S occurrences between them in a string of length S. Thus, allowing for N end-labels (maneuver envelope indices at the end rectangle) and $[(S + 1)(S + 2)/2]$ possible string values for each end-label, the number of strings that need be maintained at stage S is of order $N \cdot S^2/2)$. If the number of stages necessary to eliminate all but the highest-payoff solution strings is P, therefore (at which point no more interaction rectangles will fit within the loading zone), then the total number of comparisons is upper-bounded by $N^2 \cdot \sum_{k=2}^{P} ((S + 1)^2/2)$, a polynomial in N and P. Assuming that N is fixed for

a wide range of optimization scenarios studied following a geometric analysis, and note that P varies with loading zone length. This implies that the complexity of the optimization algorithm is a low-order polynomial in the linear extent of the loading zones involved in the scenarios.

References

[Gar72] R. S. Garfinkel and G. L. Nemhauser (1972), *Integer Programming.* New York: John Wiley & Sons.

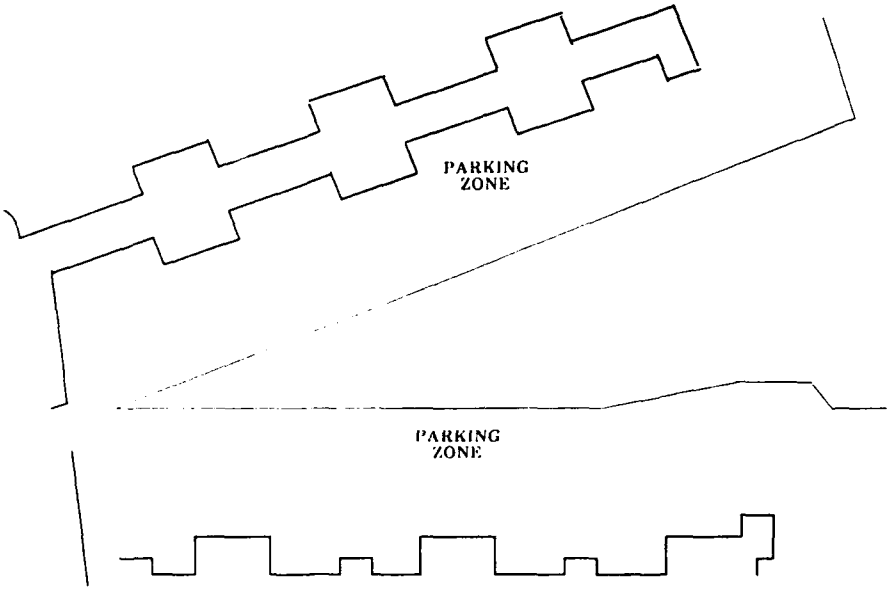


Figure 1. Airport terminal loading zones.

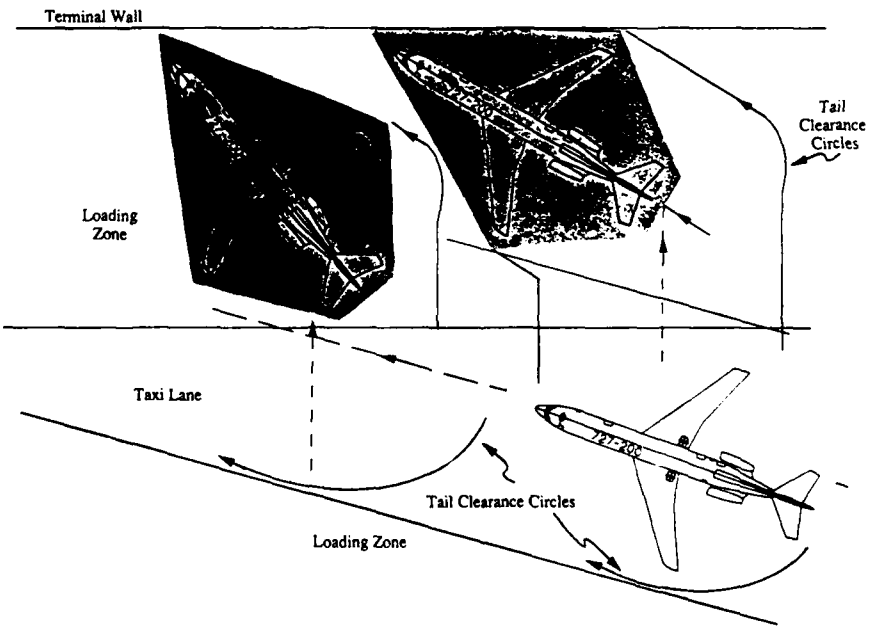


Figure 2. Geometry of taxi lane and loading zone in parking.

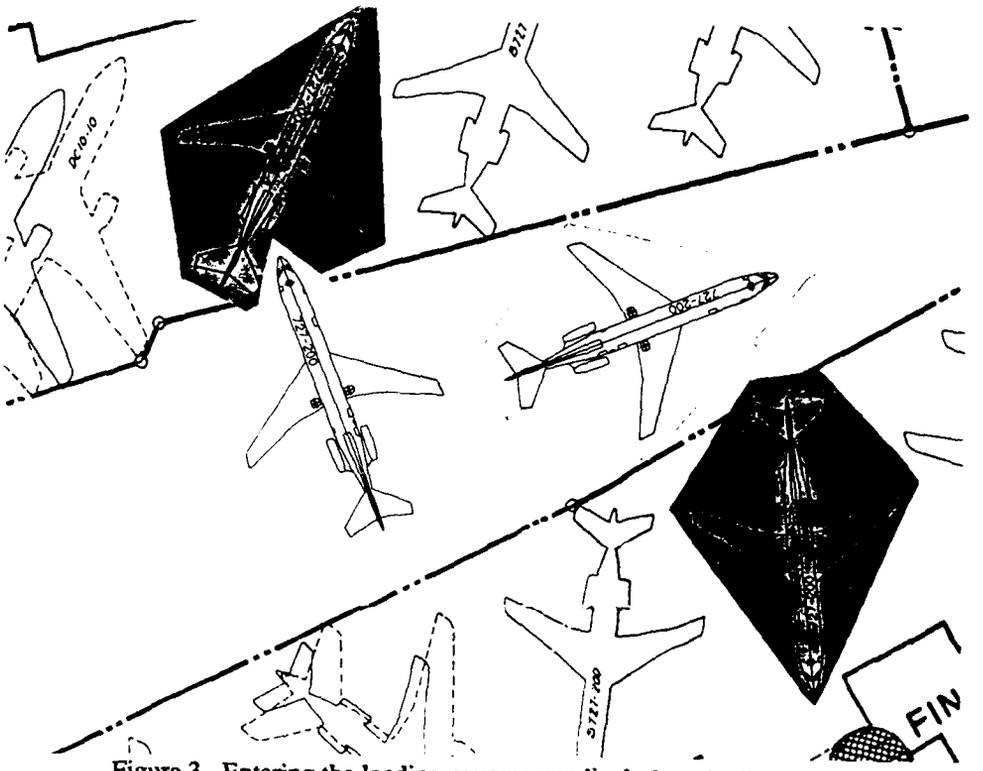


Figure 3. Entering the loading zone perpendicularly to its limit line.

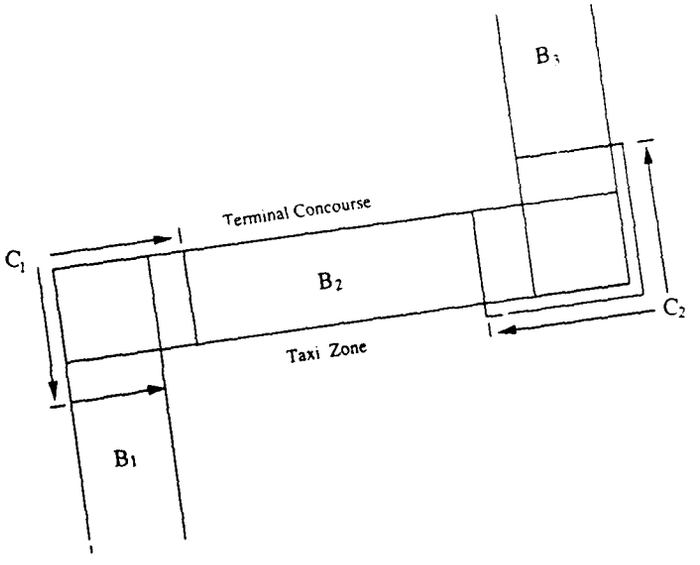


Figure 4. A loading zone formed from overlapping bands and corner slots.

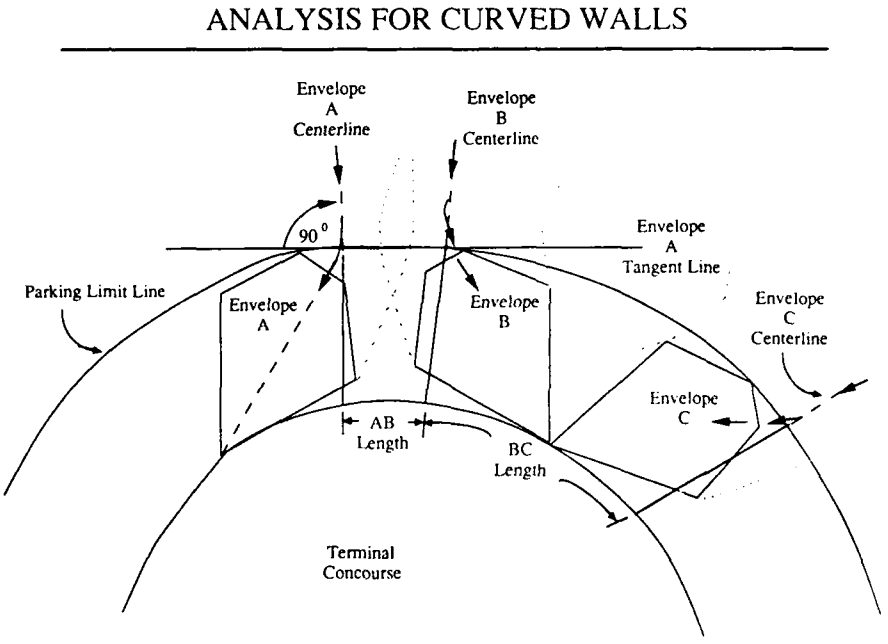


Figure 5. A loading zone following a curving terminal wall.

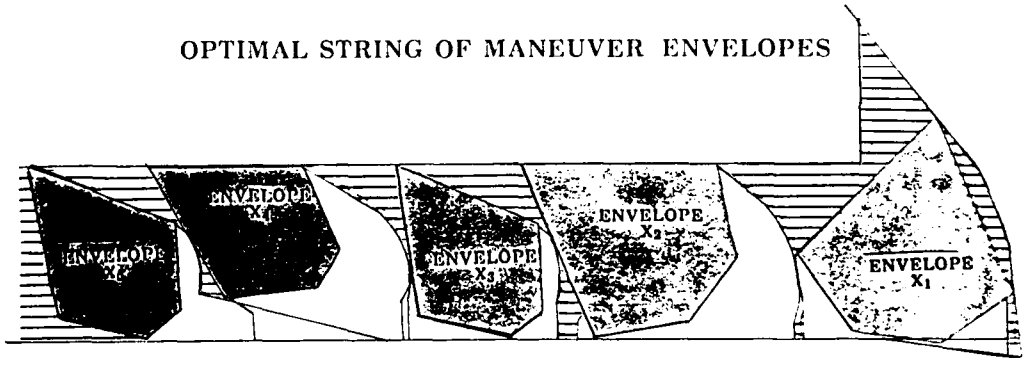


Figure 6. An optimum solution for two airplane types (large and small pentagons) in a loading zone at La Guardia Airport.

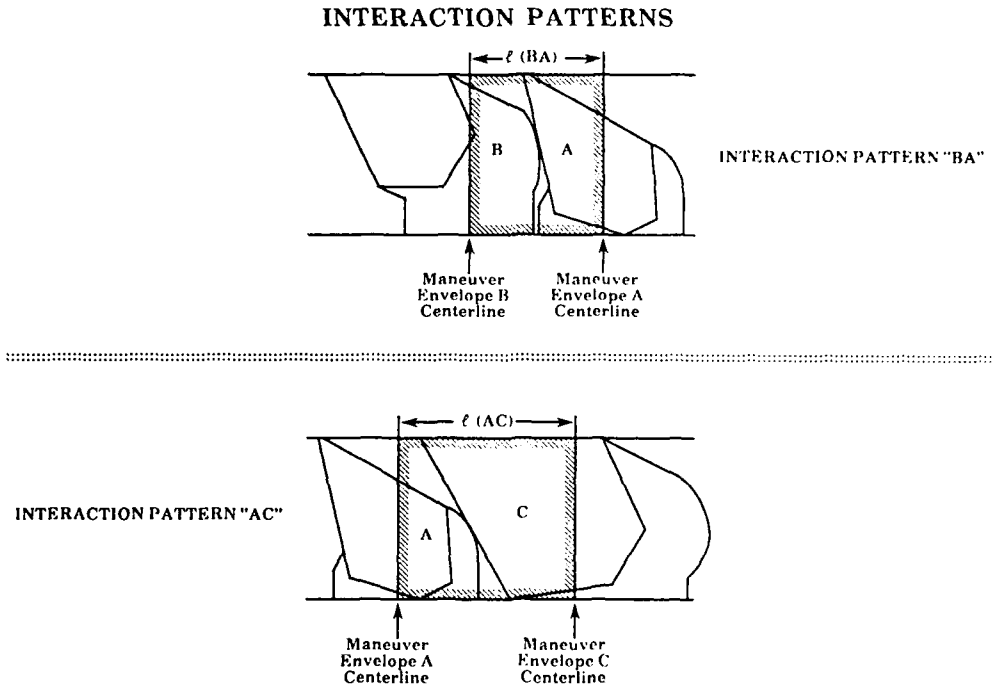


Figure 7. Two interaction rectangles created from maneuver envelopes A, B, C, and D.

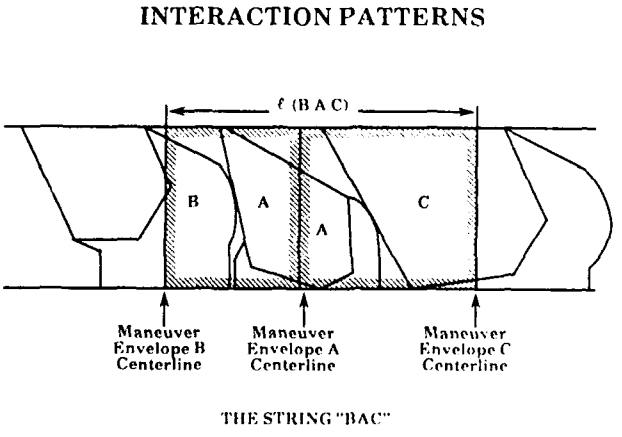


Figure 8. Two interaction rectangles joined to form part of a string. Note the matching of labels for A.

Dynamic Visualization of Late Quaternary Pollen Data

ALAN P. KNOERR,* Division of Applied Mathematics
THOMPSON WEBB, III, Department of Geological Sciences
THOMAS W. COLTHURST, Mathematics Department
Brown University, Providence, RI 02912

Abstract

Pollen from trees and other plants is preserved annually in the sediments of bogs, ponds, and lakes, forming a record of contemporary vegetation. Geologists sampling pollen preserved over the past 20,000 years have created large databases whose variables include latitude, longitude, time, abundance, and pollen type. Traditional methods of visualizing these data include pollen diagrams (time series of plant taxa abundances), classification diagrams, and contour maps of pollen abundance. While these are valuable tools, modern computer graphics technologies offer new ways of visualizing continuous multidimensional structures. In this paper, we describe the development of an interactive computer graphics package, supported by IRIS-4D workstations, for visualization of higher-dimensional aspects of the pollen database covering eastern North America.

Introduction

New methods of viewing and analyzing data in a given field often lead to theoretical insights. Such have arisen in late Quaternary palynology, the study of pollen preserved in sediments dating to 18,000 years ago or earlier. An excellent introduction to existing methods in this area is Birks and Gordon (1985).

Pollen is deposited annually, accummulating with sediments at sites such as bogs, ponds, and lakes which are favorable to its preservation. The raw data of palynology come from vertical cores or sections of such sediment, sampled at different levels. Under microscopic examination, individual pollen grains in a sediment sample are counted and classified taxonomically.

Qualitative descriptions of pollen preserved in peat bogs date from the latter part of the nineteenth century (Manten, 1967). Quantitative work begins with Lennart von Post (1918, in Swedish; reprinted in English in 1967). Von Post recorded the percent relative frequency, rather than the absolute frequency, of each taxon of interest in the sample. Plotting these percentages versus sample depth, he collected the graphs for all the taxa to form a *pollen diagram* for the site.

Pollen diagrams of percentage data continue to be the fundamental form of data representation in palynology. Through radioactive isotope dating, stratigraphic comparisons, and other methods, depth can be mapped to time with a resolution of from 100 to 250 years in most cases (Webb, 1982). Pollen collecting at a site originates with plants in the surrounding *sampling basin*, which varies with taxon, site, and time (Birks and Birks, 1980). Relative rather than absolute frequency is generally the preferred measure of pollen abundance due to robustness (Mather 1972, 1980; Davis *et al.*, 1973), despite the negative correlation it introduces between pollen types in a sample. Thus, as Von Post realized, the pollen diagram of a site approximates the vegetation history of the surrounding region. Pollen diagrams have become important sources of data for paleoecological and paleoclimatological studies of the late Quaternary (Birks and Birks, 1980; Webb *et al.*, 1987).

Pollen data may also be viewed as points in an abstract pollen space whose coordinate variables are latitude, longitude, time, abundance, and pollen type. Since the 1970's, computer-based techniques for exploratory multivariate analysis have provided new ways of visualizing pollen data. Classification methods (scaling methods, cluster analysis) based on 'dissimilarity' measures have been the most popular (Gordon, 1981). These methods generate a variety of diagrams revealing specific geometric relationships between point data. An important application has been finding modern analogues to fossil pollen assemblages (Overpeck *et al.*, 1985).

The spatial resolution of pollen data depends on both the density of core sites and the size of the sampling basins for each pollen type and site. Different processes affecting vegetation are also evident at different scales. By interpolating the data to a grid of the appropriate scales in space and time, vegetation patterns generated by processes acting at those scales emerge (Prentice, 1988). For given times and taxa, maps of *isopolls* (contours of pollen percentages) can be used to visualize these patterns. Introduced by Szafer (1935), isopoll maps have been used extensively since the 1970's. They have been especially useful at the subcontinental scale, revealing the influence of climate on vegetation (Webb, 1988).

*Present address: Department of Mathematics, Occidental College, Los Angeles, CA 90041.

Mapping methods regard data for a given pollen type as discrete samples of structures which are essentially continuous. This perspective can be extended to higher dimensions. For example, isopoll maps are temporal cross sections through a 3D *space-time box* in which isopolls are 2D surfaces (Webb, 1988). Selected plots of isopoll surfaces were first generated in 1984 by Stead and Webb (see Banchoff, 1990, pp. 82-83). Advances in computer and exploratory graphics since that time now make possible a much richer range of methods along these lines. In this paper, we report our work on interactive graphics programs for visualizing continuous structures in pollen space. Our focus is on understanding both the 4D character of a given pollen type and interactions between pollen types.

Data

We worked with twelve taxa from a late Quaternary pollen database for eastern North America maintained at Brown University. The data had been smoothed and interpolated to a space-time grid with increments of 1000 years in time and about 100 km in space. The number of original sites increases from less than 100 between 18 and 12,000 years ago to roughly 300 from 10,000 years ago to the present.

Some auxilliary data were also provided. Points of the grid covered by the receeding continental ice sheet of the last ice age were indicated. Continuous coastlines of the present and 18,000 years ago were included for geographical orientation. The paleocoastline was digitized using GSMAP (Selner and Taylor, 1989). Its northern part was traced from Dyke and Prest (1987). Since sea level rose as the ice sheet retreated, the southern part could be approximated by tracing modern bathymetric maps. The modern coastline came from a database supplied with GSMAP.

All geographical data had been projected onto the plane using an Albers equal area projection (see Snyder, 1987, p. 383). Further details concerning this database may be found in Webb (1988).

Design

Software design is a compromise between functional goals and technological constraints. While broad goals were clear initially, details were revised continuously during development. Close collaboration between users and developers contributed greatly to the success of this project.

The central problem was visualizing continuous objects in 4-space. We chose to solve this by linking 3D slices of these objects. This determined the basic visualization structure, a 3D box containing contour surfaces.

The three coordinate axes for each of the boxes are selected from the four continous variables in the dataset: latitude, longitude, time, and abundance. Four distinct boxes of this sort are possible. The variable not chosen as a coordinate determines the contour surface. For example, if a value of 40% is chosen for abundance in the space-time box, the isopoll surface enclosing grid points with an abundance value $\geq$40% is displayed. Since abundance is a function of the other variables, contour surfaces in boxes with abundance as a coordinate are also function graphs.

Each type of box is customized to enhance interpretation. Axes are labeled, and maps are drawn on the appropriate faces if the box has both space coordinates. In the space-time box, the ice sheet may be displayed as either a surface or point cloud.

The exploratory graphical principles of *linking* and *focusing* guided our work (see Stuetzle and Buja, 1990). In visualizing large multivariate data sets, focusing refers to the selection of views or subsets of data. Linking involves visually relating different views or subsets.

Focusing is accomplished here through choosing the boxes, pollen types, and surfaces to be displayed. Multiple selections may be displayed simultaneously. Surfaces in 3-space cannot be visually comprehended from a single 2D view, making the dynamic capabilities of interactive graphics essential. A box may be rotated in 3D automatically or under direct control. One can also zoom in on specific portions of a box.

Linking is used for data comparisons. Within a given box, multiple surfaces can be selected and displayed simultaneously. Wireframe or Gouraud shaded solid representations in a variety of colors and simulated materials can be selected for each surface, facilitating discrimination between multiple surfaces. Wireframes can also be overlaid on solid surfaces to create a textured appearance. For a given pollen type and 3D box, the implicit variable can be stepped through a range of values, generating an animated sequence of contour surfaces.

Different boxes can also be linked. For example, a specific time can be highlighted in the space-time box, generating the corresponding surface in the space-abundance box. If an isopoll sequence is then animated in the space-time box, a sequence of highlighted cross sections at corresponding abundance values will be animated on the surface in the space-abundance box.

Users control the programs interactively via pop-up menus and appropriate mouse or keyboard input. There are several recording options. A given image may be printed or stored as a snapshot. Entire sessions can also be recorded for later playback.

Implementation

This software was developed for the Silicon Graphics IRIS-4D series of workstations with version 3.3 of the operating system. Of the platforms available to us, this one offered the powerful interactive real-time graphics we required. The code was written in C using the graphics library and window manager documented in Silicon Graphics, Inc. (1990a, 1990b, 1990c). It is highly modular to facilitate continued development and adaptation to similar data sets.

In this environment, it easiest to run each 3D box as a separate program. Because of customization, several different programs are required. *Pollen* displays the space-time box, while *Slice* runs the space-abundance box. The boxes with time, abundance, and latitude or longitude, respectively, as coordinates are separate options in *Pathview*.

Most implementation is standard, using the rich function libraries provided by Silicon Graphics. It was necessary, however, to find a way of polygonalizing contour surfaces. A script language was also developed to simultaneously solve the two problems of recording snapshots or sessions and linking boxes.

For contour surface polygonalization, we used a variant of the *marching cubes* algorithm (Lorensen and Cline, 1987). A contour value partitions vertices in a 3D data grid into two sets. Contour surfaces separate these sets. Since membership for each vertex can be determined independently, suitably constrained contour surfaces can be constructed locally. Lorensen and Cline construct triangulated surfaces constrained to intersect edge midpoints. Favoring reduced computation at the expense of a somewhat rougher surface, we opted instead to construct triangulated surfaces contrained to intersect vertices. Each data cube has 8 vertices, so there are $2^8 = 256$ distinct ways such surfaces can intersect a cube. For each cube, a hash look-up table is used to determine what surfaces to draw.

A script language proved to be an flexible solution to recording and communication problems. The language contains all the possible user commands along with higher level programming constructs. In snapshot mode, a file of script commands necessary to reconstruct the current view is generated. While in record mode, all commands executed by the user are written to a script file which can be replayed later. The same language is used to exchange commands linking two programs.

Discussion

More experience is needed to determine how best to use features of the existing software. We have found, for example, that the choice of surface attributes is critical when viewing multiple objects in the same 3D box. Interpreting the results of linking boxes will also take practice. We need systematic ways of using these tools to develop higher-dimensional intuition.

Refinements and extensions are planned. These include improvements in contour surface polygonalization, interbox communication, and the user interface. The most important extension is to *Pathview*. The user will be able to specify a curvilinear geographical transect which will take the place of latitude or longitude. In this way, it will be possible to visualize the effects of specific geographical features.

While development continues, the current software is beginning to be used for both research and teaching. It has been found that certain contour values for each taxon yield distinctive surface forms. In some cases, these neatly summarize facts which had already been gleaned from map sequences, but new features are also emerging. The interesting geometry observed thus far is also motivating the development of mathematical models which capture this structure.

Although some of the features of this software are specific to this particular pollen dataset, only slight modifications would be required to visualize other pollen data. If more extensive changes were made, especially to the user interface, other multivariate data could be used. Future applications include visualization of relationships between pollen and climatalogical data.

Acknowledgements

This research was supported by grants from the National Science Foundation (ATM-8713981) and the Department of Energy (DE-FG02-85ER60304). We wish to thank the Center for Fluid Mechanics at Brown University for the use of their computing facilities. The pollen database we used is maintained by K. Anderson of the Department of Geological Sciences, who also supplied the digitized coastlines. As users of our programs in development, N. Rivera and G. Katz provided invaluable feedback. We thank F. Bisshop and S. Fulcomer for technical assistance, T. Banchoff and D. Cervone for useful discussions, and B. Morimoto of Silicon Graphics in Seattle for the use of a Personal IRIS workstation at the conference.

References

Banchoff, T. F. (1990). *Beyond the Third Dimension: Geometry, Computer Graphics, and Higher Dimensions.* Scientific American Library, New York.

Birks, H. J. B. and Birks, H. H. (1980). *Quaternary Paleoecology.* Edward Arnold, London.

Birks, H. J. B. and Gordon, A. D. (1985). *Numerical Methods in Quaternary Pollen Analysis.* Academic Press, New York.

Davis, M. B., Brubaker, L. B., and Webb, T., III, (1973). Callibration of absolute pollen influx. In *Quaternary Plant Ecology.* Edited by H. J. B. Birks and R. G. West. Blackwell Scientific Publications, Oxford.

Dyke, A. S. and Prest, V. K. (1987). *Paleogeography of Northern North America, 18,000-9,000 Years Ago.* Map 1703A. Geological Survey of Canada.

Gordon, A. D. (1981). *Classification: Methods for the Exploratory Analysis of Multivariate Data.* Chapman and Hall, New York.

Lorensen, W. E., and Cline, H. E. (1987). Marching Cubes: a high resolution 3D surface construction algorithm. Proceedings of SIGGRAPH'87. In *Computer Graphics*, **21**:4, 163-169.

Maher, L. J. (1972). Absolute pollen diagram of Redrock Lake, Boulder County, Colorado. *Quaternary Research*, **2**, 531-533.

Maher, L. J. (1980). The confidence limit is a necessary statistic for relative and absolute pollen data. *Proceedings of the IVth International Palynological Conference, Lucknow* (1976-1977), **3**, 152-162.

Manten, A. A. (1967). Lennart von Post and the foundation of modern palynology. *Review of Palaeobotany and Palynology*, **1**, 11-22.

Overpeck, J. T., Webb, T., III, and Prentice, I. C. (1985). Quantitative interpretation of fossil pollen spectra: dissimilarity coefficients and the method of modern analogs. *Quaternary Research*, **23**, 87-108.

Prentice, I. C. (1988). Records of vegetation in time and space: the principles of pollen analysis. In *Vegetation History*, pp. 17-42. Edited by B. Huntley and T. Webb, III. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Selner, G. I., and Taylor, R. B. (1989). *GSDRAW and GSMAP, system version 6.0: Graphics programs and utility programs for the IBM PC and compatible microcomputers to assist compilation and publication of geologic maps and illustrations.* USGS Open-File Report 89-373A (documentation) and 89-373B (disk). U. S. Geological Survey.

Silicon Graphics, Inc. (1990a). *IRIS-4D Series, Graphics Library Programming Guide, Document Version 2.0.* Silicon Graphics, Inc., Mountain View, Calif.

Silicon Graphics, Inc. (1990b). *IRIS-4D Series, Graphics Library Reference Manual, C Edition, Document Version 4.0.* Silicon Graphics, Inc., Mountain View, Calif.

Silicon Graphics, Inc. (1990c). *IRIS-4D Series, 4Sight Programmer's Guide, Document Version 3.1.* Silicon Graphics, Inc., Mountain View, Calif.

Snyder, J. P. (1987). *Map Projections - A Working Manual.* USGS Professional Paper 1395. U. S. Geological Survey.

Stuetzle, W., and Buja, A. (1990). *Visualization of Quantitative Data.* Video tape, Master #C66-91-87 4445-#1964. University of Washington Instructional Media Services, Seattle, Wash.

Szafer, W. (1935). The significance of isopollen lines for investigations of the geographical distribution of trees in the post-glacial period. *Bulletin de l'Académie Polonaise des Sciences et des Lettres B*, 1935, 235-239.

Von Post, L. (1918). Skogasträd pollen i sydsvenska torvmosselagerföldjer. *Förhandlingar Skandinavika Naturforskeres*, **16**, möte 1916, 432-465.

Von Post, L. (1967). Forest tree pollen in south Swedish peat bog deposits (Translation by M. B. Davis and K. Faegri). *Pollen et Spores*, **9**, 375-401.

Webb, T., III (1982). Temporal resolution in Holocene data. *Proceedings Third North American Paleontological Convention*, **2**, 569-572.

Webb, T., III (1988). Eastern North America. In *Vegetation History*, pp. 385-414. Edited by B. Huntley and T. Webb, III. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Webb, T., III, Bartlein, P. J., and Kutzbach, J. E. (1987). Climatic change in eastern North America during the past 18,000 years: comparisons of pollen data with model results. In *The Geology of North America*, v. K-3: *North America and adjacent oceans during the last deglaciation*, pp. 447-462. Edited by W. F. Ruddiman and H. E. Wright, Jr. Geological Society of America, Boulder, Colorado.

Didactic and Production Software for Computing Sample Variances

John C. Nash
Faculty of Administration, University of Ottawa
Ottawa, Ontario, K1N 6N5, Canada
Email: JXNHG@ACADVM1.UOTTAWA.CA

Abstract

The standard deviation (or equivalently the variance) of a sample of numbers is one of the most elementary concepts in statistics. Yet this computation harbours a number of serious difficulties, especially when the sample is large and the standard deviation is small relative to the mean.

This contribution will describe prototype software for both didactic and production use to allow reliable calculation of sample variances (or equivalently standard deviations), for a wide variety of sample sizes and data characteristics. Several illustrations of the software and its evaluation will be presented, if appropriate accompanied by a live demonstration.

1 Introduction

Computation of the sample variance, or equivalently the sample standard deviation, is one of the most common and fundamental tasks in statistical computation. Indeed it is so common that the difficulties it may present are often overlooked. There is a fairly rich literature on these difficulties and ways to overcome them. (Almost all the citations at the end of this paper concern this topic, and specific references will be placed in the body of the paper.) Nevertheless, the issue continues to give concern (see, for example, Smith, 1991, for a description of multiple complaints with the standard deviation function @STD in different versions of the popular spreadsheet program Lotus 1-2-3).

The defining formula for the variance (the adjective "sample" will be dropped where the meaning is clear) also provides a computational algorithm. Using symbols which are suitable for incorporation into a computer program, we first calculate the (sample) mean as

(1) X_bar = {SUM i:=1..n : X(i)} / n

then use this information in a second pass through the data to calculate the variance

(2) V(X) = {SUM i:=1..n : [X(i) - X_bar]^2} / (n - 1)

and hence the standard deviation is computed as

(3) SD(X) = sqrt(V(X))

The issues upon which this contribution is based are:
1) **Accuracy** -- How large is the deviation, either absolute or relative, between the computed variance and the "true" value which would be obtained if we made no error in computation? This is especially important when SD(X) << X_bar (Chan and Lewis, 1978).
2) **Efficiency** -- How fast is our calculation? How many basic arithmetic operations are we required to perform, and is this in some way optimal? In particular, we would like to avoid two passes through the data if the data set is large.
3) **Complexity** -- Are the program code and data structures simple and straightforward, or do we need complicated programs which require very careful attention to many details? Can we exploit parallel computational facilities, or partition the calculation so that regional offices can partly carry out the calculations?
4) **Education** -- How can the concerns and mechanisms for responding to them be made available to others? How can a greater awareness of the difficulties be achieved?

The work reported here is part of a long-standing and ongoing project to address these issues. The present contribution is directed primarily toward providing a prototype computer program which illustrates most of the algorithms which have been proposed to compute the sample mean and sample variance. Some ideas are presented on the preparation of "production" codes and design elements of a program to prepare specially formatted test data sets are discussed briefly.

2 Foundations

For convenience we will define the quantities

(4) T = {SUM i:=1..n : X(i)} and

(5) $S = \{SUM\ i:=1..n : [X(i) - X\_bar]^2\}$

This mirrors Chan, Golub and LeVeque (1983), who give a decision table for selecting an appropriate algorithm. The present work could be viewed in part as providing an illustration, within a single program, of the ideas contained within this decision table. They also include a survey of various error analyses which have been carried out for the different algorithms used to compute S or V(X); readers may wish to note that there are some minor differences in detail between the formulas which have been published in different reports. For consistency, we have computed error measures in our program(s) based on the formulas of Chan et al. (1983).

From a didactic point of view, error analyses can be dry and tedious enough that the important messages they carry may be overlooked. In the present application, some of these messages are:

- that the desire to overcome two passes through the data may tempt users to employ calculation methods which throw away information which is present in the data, such as the popular but dangerous "Textbook" algorithm. This is based on the algebraically equivalent forms for S

(6) $S = \{SUM\ i:=1..n : X(i)^2\} - n * X\_bar^2$
$= \{SUM\ i:=1..n : X(i)^2\} - (T^2) / n$

- that loss of information frequently occurs because the difference of two (often large) nearly equal numbers is calculated, causing digit cancellation. The Textbook algorithm can be seen to encourage such a subtraction, especially when the data elements X(i) are of a comparable magnitude.

- that a large n (large data sets) may cause inaccuracies on the computation of T, or equivalently the mean X_bar, if the accumulation is performed by adding the data elements one at a time into an accumulator. Similar difficulties may occur when updating methods are used to calculate S. This motivates the use of pairwise summation.

3 The computer program VARIANCE

The working prototype program VARIANCE has been developed and was demonstrated at Interface '91. The goals of its development are:

1) to show how the various approaches to variance calculation work, and

2) to allow different methods to be applied to different test data sets.

Furthermore, VARIANCE is designed with a unified program structure so that all variations are included within a single program, with no extra code to include or remove.

VARIANCE has the following features:

a) It has been programmed in Borland's Turbo Pascal, version 5.0, under the MS-DOS operating system. The commented, and hopefully readable, source code occupies 47K bytes and the executable form 39K. The author intends to distribute it as share-ware or at nominal cost as a self-teaching or classroom demonstration, which should interest a wide audience. Extended precision accumulation has been avoided to maintain some semblance of conformity to other variants of Pascal.

b) VARIANCE currently includes 5 main algorithms
- the Textbook algorithm
- the standard Two-Pass defining algorithm, with automatic calculation of the Björck (Chan et al. 1979) correction terms
- West's (1979) updating method
- the Pairwise algorithm of Chan et al. (1979)
- Cotton's (1975) updating method

c) Where appropriate, summations may optionally be performed in a pairwise manner.

d) The program and the data structure described below allow a set of data to be partitioned into blocks. This reflects the possibility of computation in parallel by multi-processor computing systems. Alternatively, we may think of data collected and partially processed by separate agents. The results for separate blocks may be combined by direct summation or by an extension of the pairwise updating formula of Chan et al. (1979) discussed below.

e) All data may be shifted (or coded), that is, a constant may be subtracted from each data element within a block of data. The strategies allowed for shifting are:
- No shift
- Fixed shift (for all blocks) entered by the user
- Sample the first data block, with a user-supplied sample size, and use the mean of the sample as a (fixed) shift for all blocks
- Sample each block of data, with a user-supplied sample size, and use the mean of the sample for each block as the shift for that block.

f) Operation counts of real and integer arithmetic, assignment (storage) operations, and control decisions (IF or WHILE or CASE or UNTIL) are recorded and displayed at various points in the program.

g) A number of options for control and information display are included such as:
- interactive or batch operation
- treat a data set as a single block
- pause for user response after each data element
- display intermediate results

h) error bounds from the error analyses reported by Chan et al. (1983) and others are reported with the computed mean and variance

i) control information may be obtained from a file for "hands off" operation

j) results are optionally saved in a file which is in a form suitable for use as an input file above

k) execution is timed.

4 Trial data sets

The program VARIANCE takes as input data a simple text file having the following structure:

a) Comment lines may appear anywhere in the data set and begin with a special character in the first position of the line. Currently the exclamation mark (!) is used as the comment character. We feel it is important that the number and position of comments not be restricted so that full documentation of data sets may be provided. For example, comment lines may contain "exact" results, or give timing or control information for specific computing platforms.

b) Numerical data is provided in text form, currently 1 number per line for simplicity.

c) Blocks (there must always be at least 1) are ended with a line consisting of the word ENDBLOCK

d) The data file is ended with a line consisting of the word ENDDATA. Clearly this is not needed, but documents the end of the data clearly.

An outline of a program to generate test data, VARDATA, has been prepared. The principles behind this program are that it

a) build data files in the format described above

b) be easily extended as new requirements are stated

c) use both fixed and pseudo-random series, and different distributions for pseudo-random series

d) allow different scalings and shifts to be applied, in particular to force roundings so that machine internal representations of data are inexact

e) provide exact results information in the form of comments in the data sets.

We believe that there is a need for several classes of data sets: 1) didactic sets to illustrate the difficulties and peculiarities of variance calculation methods; 2) test data sets which permit relatively rapid validation of codes and checking of details of implementations; 3) very large test data sets so that production codes can be exercised and timed. The last class may best be generated when needed so long as the generation process is reliable.

5 A generalized combining formula

Suppose we have two sets of data to which the following information applies:

	Set a	Set b
Number of Elements	n_a	n_a
Shift used	k_a	k_b
Sum of data from shift	sT_a	sT_b
Sum of deviations^2	S_a	S_b

(Note that the shift is irrelevant to these sums.)

$$sTe = \{SUM\ i:= 1\ ..\ n_e : X_e(i) - k_e\}$$

We now wish to compute the combined, unshifted values for n, T and S. Clearly

$$(7)\quad sT_e = T_e - n_e * k_e$$

$$(8)\quad T = T_a + T_b = sT_a + sT_b + n_a k_a + n_b k_b$$

The combining formula of Chan et al. (1979) may be cast in various forms

$$(9a)\quad S = S_a + S_b + Q_{ab} \qquad \text{where}$$

$$(9b)\quad Q_{ab} = (n_b T_a - n_a T_b)^2 / (n_a n_b (n_a+n_b))$$
$$= n_a n_b (T_a/n_a - T_b/n_b)^2 / (n_a+n_b)$$
$$= (n_a n_b/(n_a+n_b))\ (X\_bar(a)-X\_bar(b))^2$$
$$= [(sT_a+n_a k_a)/n_a-(sT_b+n_b k_b)/n_b]^2\ (n_a n_b/(n_a+n_b))$$
$$= n_a n_b\ [sT_a/n_a - sT_b/n_b + (k_a-k_b)]^2 / (n_a+n_b)$$

The last two forms are the generalizations for shifting. Since Chan et al. developed the formulae mainly for use in the pairwise algorithm, they did not need the extension to shifted data. Note that using a common shift for both blocks allows us to ignore the shifts. However, we may wish not to do this. As far as the author is aware, a full error analysis has not been completed for the pairwise algorithm. In the program VARIANCE, the bounds conjectured by Chan et al. (1983) have been used. No extra analysis for the extended formula involving shifts has been incorporated to date.

6 Production codes

The didactic program VARIANCE allows a user to examine the properties of different algorithms acting on various data sets. This is useful in selecting a method appropriate to a particular class of data sets so that the accuracy of the results may be controlled. The operation counts (and timing where user interaction is not required) suggest the relative efficiencies of algorithms, but do not offer a very precise measure of the timing which may be obtained under real-world operating conditions, where it is likely that data retrieval will dominate the timing.

Any production version of a mean and variance calculation program requires attention to the details of
- data transfer from storage to the calculation program and intermediate storage of sample data;
- efficient coding of algorithms in the chosen programming language -- we do not believe that the didactic code is necessarily appropriate without modification;
- extended length arithmetic for accumulation of sums;
- placement of control and timing functions so that they interfere as little as possible with the computations;
- handling of missing data;
- handling of multiple variables at one time;
- computing covariances (and hence correlations).

7 Ongoing work

Despite the fundamental nature of variance computation, a number of tasks remain to be completed. First, VARIANCE needs more thorough validation, improved commentary and documentation, and careful adaptation to different computing platforms. At Interface '91, Rich Heiberger made a number of useful suggestions, one of which is that restricted length arithmetic would be helpful in demonstrating the failure of the Textbook algorithm. Second, the data generation program VARDATA needs flesh on the skeleton. Third, some example production codes and applications to real-world data should be prepared. Collaboration in such development would be most welcome; indeed it is critical for the third task for the provision of the applications. Interested parties should contact the author.

Acknowledgements

This work and the author's participation in Interface '91 was supported by the Natural Sciences and Engineering Research Council of Canada. Earlier work (such as that reported in Nash, 1981) was carried out on machinery provided by Nash Information Services Inc.

Bibliography

Barlow, J. L. (1990) Error Analysis of a Pairwise Summation Algorithm to Compute the Sample Variance, Department of Computer Science, Pennsylvania State University, Technical Report CS-90-10.

Booth, A.D. and Booth, I.J.M. (1988) A note on the progressive calculation of the mean and variance, Journal of Computational Physics, 77, pp. 537-531.

Chan, Tony F., Golub, Gene H. and LeVeque, Randall J. (1979) Updating formulae and a pairwise algorithm for computing sample variances. Technical Report STAN-CS-79-773, November 1979, Stanford University, Dept. of Computer Science, see also *Compstat 1982* (H Caussinus et al., eds.), Proceedings of the 5th Symposium held at Toulouse, pp. 30-41.

Chan, Tony F., Golub, Gene H. and LeVeque, Randall J. (1982) Algorithms for computing the sample variance: analysis and recommendations, Tech. Report #222, Yale University, Department of Computer Science, 1982.

Chan, Tony F., Golub, Gene H. and LeVeque, Randall J. (1983) Algorithms for computing the sample variance: analysis and recommendations, American Statistician, 37(3), pp. 242-247.

Chan, Tony F. and Lewis, John Gregg, (1978) Computing standard deviations: accuracy, Tech. Rep 288, Dept. of Mathematical Sciences, The Johns Hopkins University.

Chan, Tony F. and Lewis, John Gregg (1979) Computing standard deviations:accuracy, Comm ACM, 22(9), pp. 526-531.

Cotton, Ira W., (1975) Remark on stably updating the mean and standard deviation of data, Comm ACM, 18(8), p.458.

Hanson, Richard J. (1975) Stably updating mean and standard deviation of data, Comm ACM, 18(1), pp. 57-58.

Ling R. F., (1974) Comparison of Several Algorithms for Computing Means and Variances, J Amer Stat Assoc, 69, pp. 859-866.

Miller A.J., 1989, Updating Means and Variances, J Comp Physics, 85, pp. 500-501.

Nash, John C. (1981) Fundamental statistical calculations, Interface Age, v. 6, n. 9, pp. 40-42, September 1981.

Neely P.M., (1966) Comparison of Several Algorithms for Computation of Means, Standard Deviations, and Correlation Coefficients, Comm ACM, 9, pp. 496-499.

Smith, Jennifer (1991) Nonstandard @STD in 1-2-3 Release 2.x, Lotus (magazine), April 1991, p. 19.

Welford, B.P. (1962) Note of a method for calculating corrected sums of squares and products, Technometrics, 4(3), pp. 419-420.

West, D.H.D. (1979) Updating mean and variance estimates: an improved method, Comm ACM, 22(9), September 1979, pp. 532-535.

Youngs, Edward A., and Cramer, Elliot M. (1971) Some results relevant to choice of sum and sum-of-product algorithms, Technometrics, 13(3), pp. 657-665.

KEYFINDER - A Prolog program for generating experimental designs

PETER J. ZEMROCH
Shell Research Ltd
Thornton Research Centre, P.O. Box 1, Chester CH1 3SH
England

Abstract

KEYFINDER is a menu-driven Prolog program that assists statisticians in the difficult task of generating blocked and/or fractional-replicate experimental designs in highly-constrained situations. Designs are constructed from sets of generators called "design keys". A depth-first search algorithm builds keys which yield designs matching detailed user specifications. Design parameters include the number of experimental units and the numbers of levels of the various block and treatment factors. Block factors may be combined into row-and-column, crossed or nested (split-plot) arrangements. The user can also specify the orders of treatment interactions that must remain (a) unaliased with treatment main effects and (b) unconfounded with blocks; further options are available to ensure that specific higher-order interactions of interest also remain estimable. Keys are used to generate balanced designs in which all the block and treatment factors have numbers of levels which are powers of the same prime number. Direct-product facilities allow the user to combine keys in different primes and thus produce totally asymmetrical plans. Procedures are provided for the correct randomization of all experimental plans generated, in accordance with the block structure. KEYFINDER is implemented on IBM PCs and PS/2s and, more effectively, on SUN 3 and SUN 4 workstations. Executable copies of Version 1 of the program are available on request from the author, free of charge.

1. Introduction

There are a vast number of computer programs currently on the market for the statistical analysis of experimental data, but relatively few for the equally important area of experimental design. Those design systems that have appeared in recent years have, in the main, been "expert systems" targeted at non-statisticians. Examples include CADEMO (Rasch *et al.*, 1987), DESIGN-EASE™ and DESIGN-EXPERT™ (STAT-EASE Inc., Minneapolis, MN), DESIGN EXPERT (Williams, 1991), EXPERIMENTAL DESIGN™ (Statistical Programs, Houston, TX) and SELINA (Baines *et al.*, 1986, 1988). Very few programs are available to help the professional statistician with those parts of experimental design construction that are difficult or laborious. Only PROC FACTEX and PROC OPTEX in SAS® (SAS Institute Inc., Cary, NC) readily spring to mind.

KEYFINDER (Zemroch, Lunn, Baines and Clithero, 1989; Zemroch, 1990, 1991) is a menu-driven Prolog program for generating blocked and/or fractional-replicate experimental designs. The program uses general algorithms, not stored catalogues, to produce designs so that plans can be constructed in arbitrary and quite complex situations. KEYFINDER's major strength is its ability to generate designs with user-defined confounding and aliasing patterns. This makes the program an invaluable aid to statisticians needing to produce designs in the real world of highly-constrained experimentation. Details of KEYFINDER's implementation are given in Section 2.

Designs are constructed from sets of generators called "design keys" (Patterson, 1965; Patterson and Bailey, 1978); these are described in Section 3. The aliasing and confounding properties of a design are readily deduced from its key, but the methodology has yet to win widespread acceptance because of the difficulty of reversing the deductive process. Writing down sets of generators to yield designs with predefined properties is a non-trivial task in the general case. Indeed this is a search process which is ideally suited to computerization. KEYFINDER finds keys matching detailed user specifications using a depth-first search algorithm; an outline of this is given in Section 4. The algorithm is published in Zemroch *et al.* (1989) and is more general than its predecessor, the earlier KEYGEN procedure of Zemroch (1986, 1988), and the pioneering algorithm of Franklin (1985).

KEYFINDER uses keys to generate a wide range of designs with a variety of block structures. Direct product facilities allow sets of keys to be combined together giving greater flexibility in design dimensions. Nevertheless not every useful design can be obtained from design keys and the direct product method and so the system is currently being expanded to incorporate other design classes. Full details of the designs covered are given in Section 5.

Sir R.A. Fisher (1935) first realized the necessity of

randomizing experimental designs. This must be done in due accordance with block structure. Section 6 describes the randomization and sorting facilities provided in KEYFINDER. The paper concludes with a discussion of KEYFINDER's performance in practice (Section 7).

2. The KEYFINDER program

The KEYFINDER program has facilities for (i) the construction, storage and retrieval of design keys, (ii) the subsequent generation and storage of the associated experimental plans, (iii) the randomization and sorting of experimental plans*, (iv) the construction of direct-product designs*, and (v) the execution of DOS or UNIX system commands.

KEYFINDER is written in Prolog, Prolog being chosen because of its pattern matching facilities (via unification), automatic backtracking, and richness of representation (lists, structures and a built-in database), combined with its ability to generate and execute code (see Clocksin and Mellish, 1984, or Bratko, 1986). A menu system has recently been provided to spare the user the tedium of providing sequences of complex multi-parameter Prolog queries in order to produce designs. The user simply has to select one of a number of options at each stage, or provide a single piece of information. Sensible defaults are provided as appropriate. The Main Menu is illustrated below:

```
*** KEYFINDER - Version 3.03 - Main Menu ***

1. Current design                7. Design generation, randomization
2. Declare number of units          and sorting
3. List generators               8. Form direct product design
4. Construct design key          9. Execute system command
5. Display design key            P  Exit to Prolog
6. Save/retrieve design key      Q  Quit KEYFINDER
```

Executable implementations of KEYFINDER have been developed for IBM PCs and PS/2s, using SD Prolog (Quintec Systems Ltd., Oxford), and for SUN 3 and SUN 4 workstations, using Quintus Prolog (Quintus Computer Systems Inc., Mountain View, CA). Version 1 of KEYFINDER was first released in 1989 and it, and its new User Manual (Zemroch, 1991), are available from the author, free of charge. The demonstration at Interface '91™ will include many of the new facilities to appear in the next release and these enhancements are discussed where appropriate in the present paper.

* Not available in Version 1.

3. Design keys

The most important design construction method in KEYFINDER is the "design key", invented by Patterson (1965) and discussed further by Patterson and Bailey (1978), Zemroch (1988) and Zemroch *et al.* (1989). A design key is a set of equations, e.g.

$$P = U,\ Q = V,\ \ldots;$$
$$A = UV,\ B = W,\ C = UW,\ D = VW,\ \ldots, \qquad \ldots\ldots (1)$$

relating the block factors P, Q, ... and treatment factors A, B, ... to a set of q p-level "plot factors" U, V, ... indexing the p^q experimental units (p prime). The levels of the block and treatment factors for each unit i are generated from the known levels of the plot factors by the equivalent equations

$$p_i = u_i,\ q_i = v_i,\ \ldots;\ a_i = u_i + v_i,\ b_i = w_i,\ \ldots \pmod p. \ldots\ldots (2)$$

The aliasing and confounding properties of a design are readily deduced from its key: if p = 2 in key (1), for example, then

$$CD = UW.VW = U^{1+0}V^{0+1}W^{1+1 \pmod 2} = UV = A$$
$$BC = W.UW = U = P. \qquad \ldots\ldots (3)$$

The CD interaction is thus "aliased" with the treatment factor A. This means that it will be impossible to disentangle the effect of treatment A from the CD interaction in the subsequent analysis of the experimental data. Therefore the generated design should only be used when the scientist is confident *a priori* that the CD interaction will not manifest itself in his experiment. The BC interaction is similarly "confounded" with the block factor P.

Each generator U, V, UV, ... in the design key (1) represents p-1 of the available p^q-1 available d.f. (degrees of freedom). A 4-level factor needs 3 d.f. and thus requires 3 generators, e.g.

$$A = (U\ V\ UV),$$

these forming a subgroup under multiplication (minus the identity I). The 4 combinations of levels of the plot factors U and V give the 4 levels of A.

The key for a $^1/_3$-replicate 3^3 design in 9 units might be

$$A = U,\ B = V,\ C = UV^2. \qquad \ldots\ldots (4)$$

Here the plot factors U and V each have 3 levels and so the terms U, V, UV (unused) and UV^2 each represent 2 d.f. The design points are computed as

$$a_i = u_i,\ b_i = v_i,\ c_i = u_i + 2v_i \pmod 3. \qquad \ldots\ldots (5)$$

4. Search algorithm

In KEYFINDER, design keys are tailor-made to the user's specification. The user first inputs the dimensions of his design, i.e. the number of points, the block structure (see Section 5) and the numbers of levels of the various block and treatment factors. If the design is to be a fractional-replicate, the "resolution" must then be specified. This determines the degree of aliasing which is to be permitted. In a resolution r design ($r \geq 3$), treatment main effects are mutually orthogonal but may be aliased with interactions of order r-1 and above (e.g. A = BCD if r = 4). Higher resolution designs are thus more robust against unexpected interactions than lower resolution ones, but generally need more design points to test the same number of factors. If the design is to be blocked, the user must also specify the "confounding limit" c ($c \geq 1$). This determines the permitted degree of confounding: treatment main effects and interactions of order c and below must remain unconfounded with blocks.

The KEYFINDER program searches for a key matching the user's specification using a depth-first search procedure. First the p-level plot factors U, V, W, ... are given values so that each combination of levels of U, V, W, ... uniquely identifies one of the p^q experimental units (p prime). Then a list of candidate generators, U, V, UV, W, ... , is set up, and these are allocated, without replacement, to each block factor P, Q, ... and treatment factor A, B, ... in turn; the actual order of allocation depends on the block structure. Prolog rules ensure that the design specification is adhered to at each stage, the status of the key being monitored by means of a number of internal lists. Backtracking occurs if the list of candidates is exhausted before the key is complete; sub-optimal choices of generators may thus be discarded and alternatives substituted. Combinatorial explosion is controlled using symmetry concepts. A full exposition of the search algorithm may be found in Zemroch *et al.* (1989).

5. Types of design

Version 1 of KEYFINDER has general procedures for constructing balanced multiple-, single- and fractional-replicate factorial designs using design keys. The experimental units in these designs may be arranged, as necessary, into four basic types of block structure, namely,

P		("simple")
P + Q + R + ...		("row and column")
P * Q * R * ...	(= P + Q + P.Q + R + ...)	("crossed")
P / Q / R / ...	(= P + P.Q + P.Q.R + ...)	("nested")

The terms, P, Q, P.Q, R, ... , in the above "block formulae" correspond to random terms δ_i, η_j, ξ_{ij}, ψ_k, ... (say) in the mixed analysis-of-variance model (see, for example, Scheffé, 1959), with the subscripts, i, j, k, ... , indexing the levels of P, Q, R, ... , respectively. Split-plot designs may be constructed with the block factors in a nested structure and the allocation of treatments to "error strata" (Nelder, 1965) completely under the user's control. Keys generate the general class of design in which the number of experimental units and the numbers of levels of the various block and treatment factors are all differing powers of the same prime p.

The next release, previewed at the present Interface '91™ conference, will offer a much wider range of designs. One of the most significant enhancements will be facilities for generating "compromise plans" of intermediate resolution. These designs are particularly useful in situations where cost constraints force the experimenter to use a resolution-3 design in which main effects are aliased with two-factor interactions. Whilst the user may feel confident *a priori* of the nonexistence of most of the possible two-factor interactions, there may be some pairs of treatments which he has nagging doubts about. KEYFINDER's new compromise-design procedures allow the user to request (say) a resolution-3 design in which a named subset of important two-factor interactions must remain unaliased with main effects and/or unconfounded with blocks. Design key (1) in Section 3, for example, protects the AB interaction by not allocating its associated generator UVW to any block or treatment factor. An outline of how the compromise algorithm works is given in Section 8 of Zemroch *et al.* (1989). A compromise resolution-3 design can, in many instances, allow a set of factors to be examined in substantially fewer design points than a blanket resolution-4 design.

"Asymmetrical" designs in which the numbers of levels of the block and treatment factors are free to vary can be constructed in KEYFINDER by the "direct-product" method. A resolution-3 36-unit $2 \times 3^2 \times 6^2$ design, for example, can easily be formed by combining resolution-3 2^3 and 3^4 sub-designs, generated from design keys, with 4 and 9 units respectively. The 36 rows of the main design matrix are obtained by juxtaposing the rows of the 4- and 9-unit sub-designs in all possible ways. Each 6-level factor is formed from the 6 combinations of levels of a 2-level and a 3-level sub-factor

The development of KEYFINDER is ongoing, the aim being to produce a comprehensive toolkit capable of generating almost all existing blocked and/or fractional-replicate designs of sizes likely to be used in real-world experimentation (under certain constraints of balance). In order to make the program as complete as possible, the main effort is presently being devoted to developing methods for generating balanced designs that cannot be obtained from design keys and/or the direct-product method. Typically these designs will have numbers of units that are not prime powers and they will

include both symmetrical and asymmetrical arrangements.

6. Randomization

Randomization of an experimental design reduces the risk of certain treatment levels being unfairly favoured or disfavoured in the experiment by extraneous sources of variation, for example equipment wear, fertility trends, or changes in the weather. Most of the design keys generated by KEYFINDER yield designs with the levels of the treatment factors in some sort of systematic order. This can increase the chances of systematic external factors introducing bias and heightens the importance of correctly randomizing the design in accordance with the block structure.

Randomization in KEYFINDER is a three-stage process, all stages being optional and under the user's complete control. First, the levels of each block and treatment factor are randomized by performing a random permutation on the integers 0, 1, ..., s-1 labelling its s levels; different permutations are used for each factor. If a block factor, Q say, is nested within another block factor P (e.g. fields within farms), then the levels of Q are randomized separately for each level of P. If the design is blocked, the next stage in the process is to sort the experimental units according to the (randomized) levels of the block factor(s). The final step is to randomize the order of the experimental units. If the design is blocked, then the units are randomized separately within each block (or block factor combination).

7. Performance

On a SUN 3 or SUN 4 workstation, KEYFINDER can generate keys for designs with, say, 250 or fewer points, without time or storage becoming a problem. Thus the program caters comfortably for most of the design sizes likely to be used in real-world experimentation. However, storage problems are, at present, a constraint on the scope of the PC version and this cannot generate keys for designs with more than about 100 points. Expected software and hardware developments should ameliorate these problems in the near future.

References

Baines, A. and Clithero, D.T. (1986). Interactive user-friendly package for design and analysis of experiments. In "COMPSTAT 1986 Proceedings in Computational Statistics" (eds. F. De Antoni, N. Lauro and A. Rizzi), Physica-Verlag, Heidelberg, 320 - 325.

Baines, A., Clithero, D.T. and Zemroch, P.J. (1988). SELINA - A conversational package for the statistical design and analysis of experiments. In "COMPSTAT 1988 Software Catalogue", UNI•C, Copenhagen, 13 - 14.

Bratko, I. (1986). Prolog Programming for Artificial Intelligence. Addison-Wesley, Wokingham.

Clocksin, W.F. and Mellish, C.S. (1984). Programming in Prolog, Second Edition. Springer-Verlag, Berlin.

Fisher, R.A. (1935). The Design of Experiments. Oliver & Boyd, Edinburgh.

Franklin, M.F. (1985). Selecting defining contrasts and confounded effects in p^{n-m} factorial experiments. Technometrics, **27**, 165 - 172.

Nelder, J.A. (1965). The analysis of randomized experiments with orthogonal block structure. I. Block structure and the null analysis of variance. Proceedings of the Royal Society **A, 283**, 147 - 162.

Patterson, H.D. (1965). The factorial combination of treatments in rotation experiments. Journal of Agricultural Science, **65**, 171 - 182.

Patterson, H.D. and Bailey, R.A. (1978). Design keys for factorial experiments. Applied Statistics, **27**, 335 - 343.

Rasch, D., Guiard, V., Nürnberg, G., Rudolph, E. and Teuscher, F. (1987). The expert system CADEMO, computer-aided design of experiments and modelling. Statistical Software Newsletter, **13** (No. 3), 107 - 114.

Scheffé, H. (1959). The Analysis of Variance. Wiley, New York.

Williams, C.L. (1991). A clinical application of expert system methodology. Journal of Applied Statistics, **18**, 185 - 201.

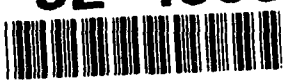
Zemroch, P.J. (1986). The computerized generation of blocked incomplete factorial designs in the conversational experimental design and analysis package SELINA. In "COMPSTAT 86 Short communications and posters", Dipartimento di Statistica Probabilità e Statistiche Applicate, Università "La Sapienza", Rome, 227 - 228.

Zemroch, P.J. (1988). Strategies for generating blocked fractional replicate designs by computer. Computational Statistics Quarterly, **4**, 43 - 57.

Zemroch, P.J., Lunn, K., Baines, A. and Clithero, D.T. (1989). Finding design keys using Prolog. Computational Statistics Quarterly, **4**, 311 - 332.

Zemroch, P.J. (1990). KEYFINDER: A Prolog system for finding design keys - Version 2. In "COMPSTAT 1990 Software Catalogue", Atlas - Congress Department, Zagreb, 5 - 6.

Zemroch, P.J. (1991). KEYFINDER User Guide - Version 1.3. Shell Research Thornton, Chester.



Exploratory CART For Semi-Markov Models*

Orna Intrator
Division of Applied Mathematics, and
Center for Gerontology and Health Care Research
Brown University, Providence, RI 02912
Email: orna@brownvm.brown.edu

Abstract

This paper describes a nonparametric application of CART (Breiman et al., 1984) to semi-Markov models, to provide a nonparametric regression analysis of transition data. Modeling data without any assumptions about the nature of the underlying distributions is needed for initially investigating predictor effects in an exploratory analysis. The semi-Markov assumption specifies a structure for the transition process, which is characterized by the one-step transition distributions. The nonparametric regression is done on these distributions. For each one-step transition distribution, the recursive partitioning of the variable space allows greater interpretability of the data by splitting the data into homogeneous subpopulations, and by providing insight into the relative importance of the different predictors, and the way in which they interact. This method is then applied to modeling payment source changes of nursing home residents.

1 Introduction

Transition processes occur naturally in many settings, classical examples are progression between disease states, changes in employment, etc. Probabilistic modeling of a transition process requires assumptions about the dependence of the process on its past history. Many times a first-order Markov assumption is made, identifying this dependence only on the current occupied state. A Markov assumption for continuous time processes can be extended to a semi-Markov assumption which allows for non-exponential waiting times in states. Lagakos et al. (1978) proposed nonparametric estimates for a homogeneous semi-Markov process. However, in applying this method to some of our application data, we found that the estimated process predicts very poorly (Intrator, 1991a). One reason for this is that the process varies for different subpopulations. This observation is the motivation for this work: We would like to be able to find the processes of different subpopulations within a nonparametric framework. The different subpopulations can be identified by answering some prognostic type questions such as: Which are the important variables for prediction? Can these variables be ranked in order of importance? Which questions most *likely lead to* others so as to determinate a sample more homogeneous in term of its waiting time distribution? These questions naturally lead to decision trees.

Classification and regression trees (Breiman et al., 1984) is a method that recursively partitions the space of explanatory variables, building a binary decision tree. Since a full grown tree may be biased towards the training data, they suggest to prune the full grown tree by penalizing the relative improvement of a split compared to the addition of an extra node. In this way a sequence of nested trees can be defined, and one can chose the "best" nested tree in an exploratory fashion, or by a good estimate of a prediction error. Among the contributions of CART are the ease of interpreting its results, by providing insight and understanding into the predictive structure of the data. It is a variable selection method which helps in reducing sensitivity to many variables in a model. It is relatively unbiased to training data, due to the pruning mechanism. It is totally nonparametric and does not require underlying model assumptions. Most importantly, it identifies effects within subgroups in contrast to standard regression methods which identify effects across the entire sample.

*Research partially supported by a grant from the Agency for Health Care Policy and Research #062332

2 Methodology

Gordon and Olshen (1985) presented perhaps the first extension of regression trees (CART) to survival data. Regression trees for survival data are nonparametric methods for estimating the distribution of a censored failure time r.v. T, given regressors x, $Pr(T > t|x)$. Under a semi-Markov assumption it is possible to reduce transition data to a set of conditional one-step survival distributions and apply an extensions of CART to survival data to each one-step waiting time distribution. We will first discuss the reduction, and then give the highlights of an extension of CART to survival data: Survival trees (Intrator, 1991b).

2.1 Reduction to one-step transitions

A finite state space continuous time semi-Markov process can be defined as: (a) a continuous time process with a Markov embedded chain of state occupancies; (b) distribution of waiting times that depend only on current state and destination state, which are independent between epochs.

The general likelihood under this model is

$$\mathcal{L} = \prod_{i=1}^{N} \theta(z_0^i) \prod_{m=1}^{M_i} f(z_m^i, t_m^i | z_{m-1}^i)$$

for N individual histories, with M_i transitions for each individual history i, $(z_0^i, t_1^i, z_1^i, \ldots, t_{M_i}^i, z_{M_i}^i)$. z_j denote sattes, and t_j denote waiting time in state t_{j-1}. $\theta(z_0^i)$ are the initial state probabilities, and $f(j, t|i)$ the densities of transition from state i to state j at time t. If censoring is considered an absorbing state we can rewrite the likelihood by:

$$\mathcal{L} = \prod_{z \in T} \prod_{z^* \in T \bigcup A} \left\{ \prod_{i=1}^{N} \theta(z_0^i) \prod_{m=1}^{M_i} \{ f(t_m^i | z, z^*) \Theta(z^* | z) \}^{\delta(z_m^i, z^*, z_{m-1}^i, z)} \right\}$$

where A is the set of absorbing states, and T is the set of transient states. $\delta(1, b; c, d) = 1$ if $a = b$ and $c = d$, and 0 otherwise. $\Theta(z^* | z)$ is the transition probability of the embedded Markov chain.

Under this framework every particular transition is a separate failure event, with processes at the same current state with other destinations considered as "censored". Applying survival trees to $f(t | z, z^*)$ should indicate the structure of the variables affecting this conditional one-step transition distribution.

2.2 Survival trees

Any extension of CART includes the following ingredients that should all be nonparametric: (1) Prediction rule; (2) Dplitting rule for growing the tree; (3) Pruning mechanism; (4) Method for tree selection.

Intrator (1991b) reviews the different extensions of CART to survival data in liu of these points and proposes an extension in which all above ingredients are addressed to achieve CART's advantages. The following is a summary of that extension.

(1) Prediction rules are the nonparametrically estimated conditional Kaplan-Meier survival distributions (which are equivalent to the estimates of Lagakos et al., 1978 , and Dinse and Larson, 1986).

(2) The splitting rule defined is based on between node separation measures such as extensions to censored data of rank type tests for the conditional survival distributions.

(3) The pruning method is based on the significance level values $v(t, d) = \Pr(S(t_L^d) \neq S(t_R^d))$, the p-value of the test d for the split at that node to the left and right branches t_L^d and t_R^d. The chosen split is $v(t) = \min_d v(t, d)$. The risk for every terminating node is defined to be zero. For any decision node t we define the risk of its branch t_T by:

$$R(t_T) = p(t) \cdot \max_{\tau \in \tilde{t}_T - \tilde{t}_T} \left[(1 - v(\tau)) \right]$$

where $p(t)$ is the proportion of observations at node t. Notice that the maximization is done over all the nodes of the branch t_T and not only over the terminating nodes $\tilde{t}_T$, so $R(t_T)$ is monotonically non-increasing when going from top down in the tree. This definition allows us to use the CART method for cost complexity pruning in the usual way. For more information about the pruning see Intrator (1991b).

(4) We can choose a tree from the nested sequence of pruned subtrees in an exploratory fashion by choosing the tree at a prespecified level of α, or by choosing a tree with a certain number of terminating nodes. Exploratory trees serve as a basis for comparison with other trees, for the effects of covariates.

An alternative to exploratory tree selection is based on computing an *honest estimate* $R^*(T_\alpha)$ of the prediction error of the trees $\{T_\alpha\}$ in the sequence of pruned subtrees. We propose to use:

$$R^*(T_\alpha) = \sum_{t \in \tilde{T}_\alpha} p(t) \Pr(S^{t_\alpha}(t) \neq S^{\mathcal{L}}(t)),$$

where $S^{t_\alpha}(t)$ is the estimated survival curve at node t based on the testing data, and $S^{\mathcal{L}}(t)$ is that based on

the learning sample. We estimate the prediction error of a node, $\Pr(S^{ts}(t) \neq S^{\mathcal{L}}(t))$, by running a test sample down the tree and comparing survival distributions between the training sample and the testing sample, at every terminating node. We can choose the right-sized tree as that subtree with the lowest estimate of the risk.

3 Application

In the following application we look at changes in payment sources for patients newly admitted to nursing homes. More details about this study can be found in Mor et al. (1991). The data come from regular assessments of patients in the National Health Corporation (NHC), a chain of 48 for-profit nursing homes operating in 11 states, mainly in Missouri, Kentucky, South Carolina, and Tennessee. Assessment records are collected at admission, periodically thereafter, and at discharge on a standard form. Data on payment source changes is recorded retroactively with the correct date of change, thus payment sources are monitored continuously. The specific cohort used for this analysis consists of all newly admitted patients to an NHC home in 1982 or in 1984. This analysis is part of a research on characterizing residents who spend down to Medicaid. In applying the above methods our aim is to find important predictors for modelling the process of payment source changes till patients become Medicaid recipients, go back to the community, or die.

3.1 Results

We defined the following state space for this application: Transient states: Private Payment; Medicare; "Other" Insurance payment; Home; Hospital. Absorbing states: Medicaid; Death; Censoring (usually due to return into the community). Our attention in this paper focuses on the transitions from Medicare to Medicaid and to death. Discussion of other transitions can be found in Intrator (1991a) . The variables investigated include activity of daily living (ADL), a heirarchical scale from 1 (best) to 5 (worst), sex, education, marital status, living arrangement prior to entrance into nursing home, diagnoses of acute or chronic illness, state (KY, MO, SC, TN), age, and initial payment source.

Figure 1 presents the full grown tree (without pruning) for the transition from Medicare to Medicaid

A first level pruning would eliminate terminating nodes E and F, and a second level pruning would eliminate all subnodes of node 2. State participation seems to be the most important predictor for this transition.

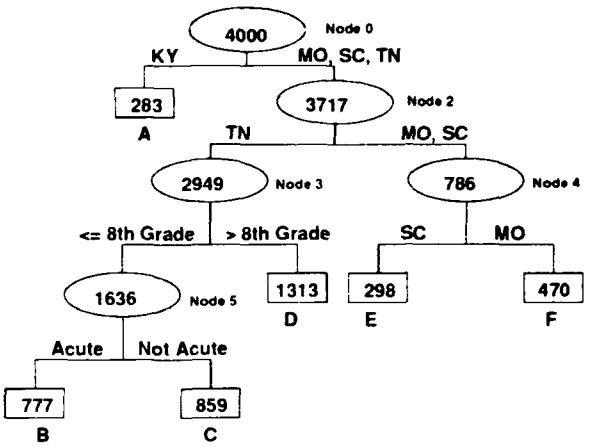


Figure 1: Full Tree for Transition from Medicare to Medicaid. Squares and capital letters indicate terminating nodes, number in nodes indicates number of observations in node.

Specifically, being in Kentucky increases the hazard of conversion the most. The competition at this node between the split on Tennessee and Kentucky is very close, (at chi-square of 143 vs. 145), therefore it is not surprising that the next split is on Tennessee. Education is important only in Tennessee.

Figure 2 presents a pruned tree for the transition from Medicare to death. The pruning was done in an exploratory manner, at a level $\alpha = .07$.

For this tree, the root split is by most severe functional impairment ADL=5. Patients who are severely impaired (ADL=5) are then split according to whether or not their impairment is acute (hip fracture or stroke). Patients who are less severely functionally impaired (node 2) are split by sex, with competition of the split from both the next ADL level 4, and from acute. The next split for both males and females of ADL levels 1-4 is on acute, and thereafter on ADL levels.

3.2 Cox type regression

In Intrator (1991a) a Cox type regression method for transition data was developed, and applied to this data as well. Here we would like to compare the results of that method (Table 1), after variable selection at a 95% level, with the tree results.

For the transition to death both methods reveal that ADL is a most prominent predictor. Sex is also predictive in both analyses. Kentucky, or any other state participation is not present in the trees at all, although it is present in the regression analysis.

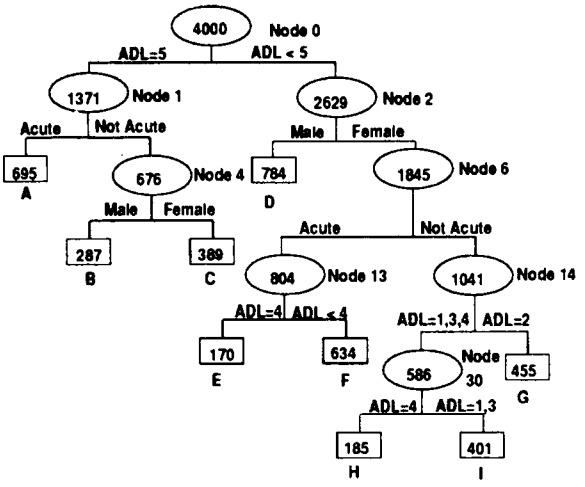


Figure 2: Pruned Tree for Transition from Medicare to Death. Squares and capital letters indicate terminating nodes, number in nodes indicates number of observations in node.

For the transitions to Medicaid, the trees emphasize the effect of state participation, and education in Tennessee. In the regression model we have Tennessee and more education variables, but also ADL levels which do not appear in the trees. This may reflect either correlated covariates, or effects across samples, which are eliminated with interactions.

Further analysis of this data, concentrating on those residents initially admitted as private paying individuals, under a Cox model, with comparison to results of the tree analysis is forthcoming in a joint work with Anthony Lancaster, and Vincent Mor.

From Medicare	Variable	$\hat{\beta}$	$\hat{\sigma}(\hat{\beta})$
To Medicaid	ADL = 2	-0.472	0.228
N = 637	1-8 Grades	-0.449	0.186
	9-12 Grades	-0.920	0.201
	> 12 Grades	-1.465	0.249
	Tennessee	-0.900	0.201
	Home Days	-.994e-2	.140e-2
To Death	ADL = 3	0.903	0.230
N = 1420	ADL = 4	1.231	0.230
	ADL = 5	1.825	0.225
	South Carolina	-0.330	0.132
	Kentucky	-0.366	0.134
	Male	0.462	0.060

Table 1: Coefficients for one-step transition from Medicare

4 Conclusions

The importance of the tree analysis in this context was to highlight the structure of the interactions between the variables affecting the one-step transition distributions. The Cox-type regression model could only identify effects across the sample, thus leading to identification of meaningful predictors that were perhaps only correlated with other important predictors. The interpretability of the tree results is self evident. It is easy to point out the prognostic variables affecting the process. The regression model, on the other hand, can provide easier predictions of summary statistics, as total probability of transition at different times, and expected number of transitions.

References

Breiman, L., Freidman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees.* The Wadsworth Statistics/Probability Series, California.

Dinse, G. and Larson, M. L. (1986). A note on semi-markov models for aprtially censored data. *Biometrika*, 73(2):379–386.

Gordon, L. and Olshen, R. A. (1985). Tree-structured survival analysis. *Cancer Treatment Reports*, 69:1065–1069.

Intrator, O. (1991a). *Methods for Exploring Survival Data.* PhD thesis, Brown University.

Intrator, O. (1991b). Survival trees: Exploratory cart for survival analysis. Technical report, Center for Gerontology and Health Care Research, Brown University. Submitted to Biometrics.

Lagakos, S., Sommer, C. J., and Zellen, M. (1978). Semi-markov models for partially censored data. *Biometrics*, 65:311-317.

Variance-reducing Kernels for Mixture Decomposition

Michael D. Lock
Michael E. Tarter
Department of Biomedical and Environmental Health Sciences
University of California, Berkeley, California 94720

Christina C. Mellin
Precision Data Group
2717 Benvenue, Berkeley, CA. 94705

Abstract

Methodology is described for constructing kernels for the purpose of identifying and separating the components of a mixture of densities. One such kernel has the property of reducing the variance of the individual subcomponents of a mixture thereby making them more visible. A second method based on a weighted version of the Mean Integrated Square Error metric takes advantage of the properties of mixtures comprised of densities with differing location parameters. The resulting kernel focuses alternatively on either the right or the left side of the variate support region. Combined with the variance-reducing kernel, this procedures enhances the estimation of either the leftmost or rightmost mixture subcomponent.

1. Introduction

As detailed by Titterington, Smith and Makov (1985), mixture distributions have had a long and rich history. The methodology outlined below is a kernel-based curve estimation approach to mixture decomposition which incorporates two novel elements. The first is that a kernel is constructed which has the property of reducing the variance of the individual subcomponents of the mixture. The second relies on modifying the kernel to enhance the estimation of a particular subregion of the density.

This second procedure takes advantage of the fundamental asymmetry inherent in mixtures which have components with unequal location parameters. Since by definition there exists some region of the density where one subpopulation is more prevalent than another, the estimation of individual subcomponents can be improved by using different kernels for different subregions. In this way the method outlined below is similar to the variable kernel method described by Breiman, Meisel and Purcell (1977). The combination of a variable kernel with a variance-reducing kernel has the potential to greatly enhance mixture decomposition methodology.

Given an independent sample $X_1,\ldots,X_n$ from the density f, the kernel estimator of f is defined by Silverman (1986) as

$$\hat{f}(x)=\frac{1}{h}\sum_{j=1}^{n}\mathrm{K}\left(\frac{x-X_j}{h}\right)$$

where K is the kernel function and h is called the smoothing parameter or bandwidth. K is often chosen a priori from a class of nonnegative, symmetric functions, for example the family of Gaussian densities with scale parameter h (Rudemo, 1982).

Another popular class of density estimators is based on orthogonal series expansions. In particular, the Fourier series estimator is defined

$$\hat{f}(x)=\sum_{k=-\infty}^{\infty}b_k\hat{B}_k\exp\{2\pi ikx\}$$

where

$$\hat{B}_k=n^{-1}\sum_{j=1}^{n}\exp\{-2\pi ikX_j\},$$

$i=\sqrt{-1}$ and $\{b_k\}$, the multiplier sequence, is a sequence of real numbers chosen to optimize the estimator in some respect. For example, $\{b_k\}$ may be used to truncate above expansion at some optimal point. The main focus of this paper is the choice of particular multipliers for the purpose of identifying and separating the components of a mixture distribution.

One of the advantages of Fourier series estimators is their near identity with kernel methods; that is, with few exceptions, a particular estimate may be expressed as either a Fourier series estimator or a kernel estimator, whichever interpretation is more convenient. This can be seen through a simple rearrangement of the above expression for the Fourier series estimator:

$$\hat{f}(x)=\sum_{k=-\infty}^{\infty}b_k\left[n^{-1}\sum_{j=1}^{n}\exp\{-2\pi ikX_j\}\right]\exp\{2\pi ikx\}$$

$$=n^{-1}\sum_{j=1}^{n}\left[\sum_{k=-\infty}^{\infty}b_k\exp\{2\pi ik(x-X_j)\}\right].$$

If $b_k=b_{-k}$ and $\sum_{k=-\infty}^{\infty}b_k<\infty$, then $\{b_k\}$ is the sequence of Fourier coefficients of the kernel defined by the Fourier series density estimator. Alternatively, many kernel

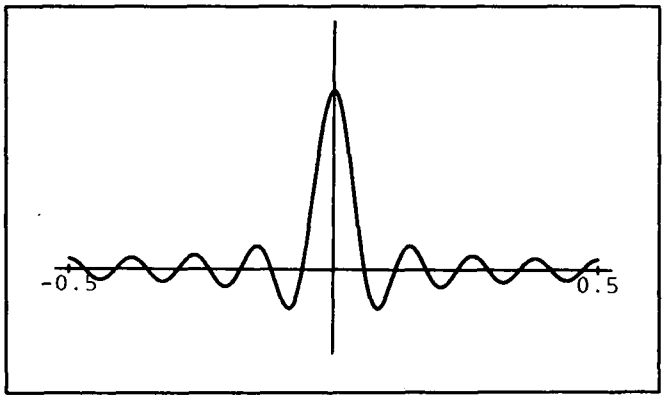


Figure 1. Dirichlet kernel with the truncation point, m, equal to 6.

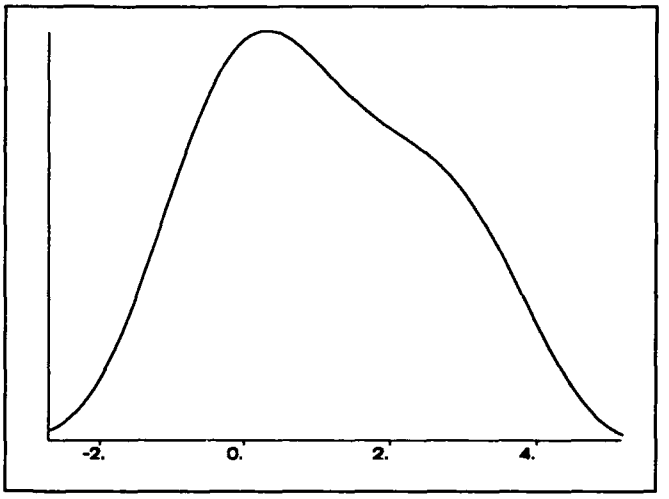


Figure 2. Estimate of a mixture of two Gaussian densities: $f(x) = .6N(0,1) + .4N(2.5,1)$.

estimates can be expressed as Fourier series by using the expressions for the characteristic functions of truncated densities given by Kronmal and Tarter (1968).

As an illustration, consider the estimator determined by the multiplier sequence $b_k = 1, |k| \le m;\ b_k = 0, |k| > m$, where m, the truncation point, is some positive integer. This leads to what Wahba (1981) calls the *raw* Fourier series estimator:

$$\hat{f}(x) = \sum_{k=-m}^{m} \hat{B}_k \exp\{2\pi i k x\}.$$

The kernel defined by this multiplier sequence is the Dirichlet kernel, shown in Figure 1.

The close correspondence between kernel and Fourier estimators means that theoretical results derived for one method are often applicable to the other method as well. In addition, the kernel form the estimator is often quite helpful in conceptualizing the series estimation process. Thus, in the following exposition we will interchange kernel and series terminology and concepts where one is more appropriate than the other. In particular, a mixture decomposition process will be developed from the series point of view but will also be presented in terms of the kernel defined by the procedure.

2. Variance-reducing Kernels

As noted above, the multiplier sequence, and thus the kernel, is usually chosen to optimize the estimator with respect to some global measure of accuracy, that is, with respect to some metric. For example, an extensively studied metric is the Mean Integrated Square Error, MISE:

$$J(\hat{f}, f) = \mathrm{E} \int \left\{ \hat{f}(x) - f(x) \right\}^2 dx$$

where the integral is taken over the entire range of the density's support. Methods for choosing an MISE-optimal multiplier sequence based on selecting a truncation point, m, for the raw Fourier series estimator have been proposed by Tarter and Kronmal (1976), Hart (1985) and Diggle and Hall (1986). More general multiplier sequences have been suggested by Watson (1969), Fellner (1974), Brunk (1978) and Wahba (1981).

By minimizing the MISE, these methods all strive to produce density estimates which are optimal in an overall sense, that is, accurate throughout the entire range of the estimate. Thus, these multiplier sequences are designed to provide an overall view of the density function. Other choices of multipliers, however, may be more suitable for more specialized purposes. Two such alternatives described below are designed to help identify and separate the individual components of a mixture of densities.

Consider the density estimate shown in Figure 2. Here $\{b_k\}$ was chosen to minimize the MISE according to a procedure outlined in Tarter and Kronmal (1976). The true density is a mixture of two Gaussian curves, although the substantial overlap between the components has made this structure difficult to see. To enhance the distinction between the subcomponents, a density estimate could be constructed which reduced the overlap between them. Such an estimate is shown in Figure 3; here the two subpopulations are clearly visible. Although certainly not optimal in an overall sense, the estimate shown in Figure 3 is certainly more useful than the estimate in Figure 2 for searching for hidden subcomponents.

The procedure used to create the estimate shown in Figure 3 is described in Chapter 4 of Titterington, Smith and Makov (1985) and relies on a particular choice of multiplier sequence. Specifically, let $\hat{b}_k$ be the multiplier sequence chosen for an initial estimate of the density, like

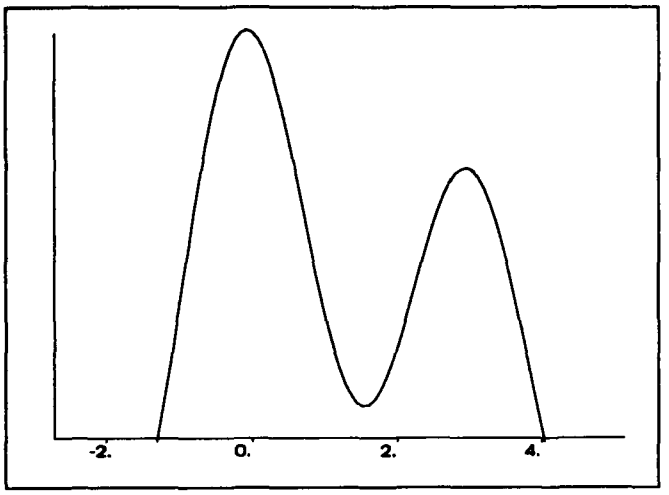


Figure 3. Estimate of the density described in Figure 2 after application of the variance reduction process.

that shown in Figure 2. Usually $\hat{b}_k$ will be chosen to be optimal in some global sense such as minimal MISE. A reduced-overlap estimate of the density can then be obtained by selecting the sequence

$$b_k^\lambda = \hat{b}_k \exp\left\{2(\pi k\lambda)^2\right\}, \quad k = \pm 1, \pm 2, \ldots$$

where λ is a user-selected, positive number. Although used on a mixture of Gaussian components here, this process has been shown by Tarter (1979) to be applicable to a broad class of mixtures.

The application of $\{b_k^\lambda\}$ reduces the variance of the estimated subcomponents while leaving the other moments of the distribution unaffected. The degree of variance reduction is determined by the magnitude of the constant λ. The effect of λ can be seen by graphing the kernel determined by the multiplier sequence $\{b_k^\lambda\}$. The kernels depicted in Figure 4 both incorporated the initial sequence $\hat{b}_k = 1, |k| \le 6$; $\hat{b}_k = 0, |k| > 6$ but used different values of . Note that for a very small value of λ the kernel looks very much like the Dirichlet kernel shown in Figure 1 and thus the process has little effect. The larger value of λ results in a more extreme kernel; in particular, the amount of negativity increases as λ increases. Since as noted in Jones (1991), a nonnegative kernel inflates the variance of an estimate, it is not surprising that variance is reduced by increasing λ and hence increasing kernel negativity.

Once the overlap between components has been reduced, a procedure outlined in Tarter (1979) can be used to eliminate the contribution of one of the now distinct subcomponents. The variance-reduction process can then be reversed resulting in an estimate of the remaining component. (The process also extends easily to any number of distributional subcomponents.) Once isolated the remaining component can be analyzed by the model identification techniques described in Tarter and Lock (1988).

3. Locally-enhanced kernels

In the previous section it was suggested that a globally optimal estimate of the density should be constructed prior to decomposing a mixture distribution. However, the ultimate goal of the above example was to produce and analyze an estimate of only the left component of the mixture. With this in mind it is clearly advantageous to estimate the left side of the distribution as accurately as possible, even at the expense of losing some resolution in the right side of the estimate. This can be accomplished by selecting a multiplier sequence which minimizes the weighted MISE:

$$J(\hat{f}, f, w) = \mathrm{E}\int\left\{\hat{f}(x) - f(x)\right\}^2 w(x)dx.$$

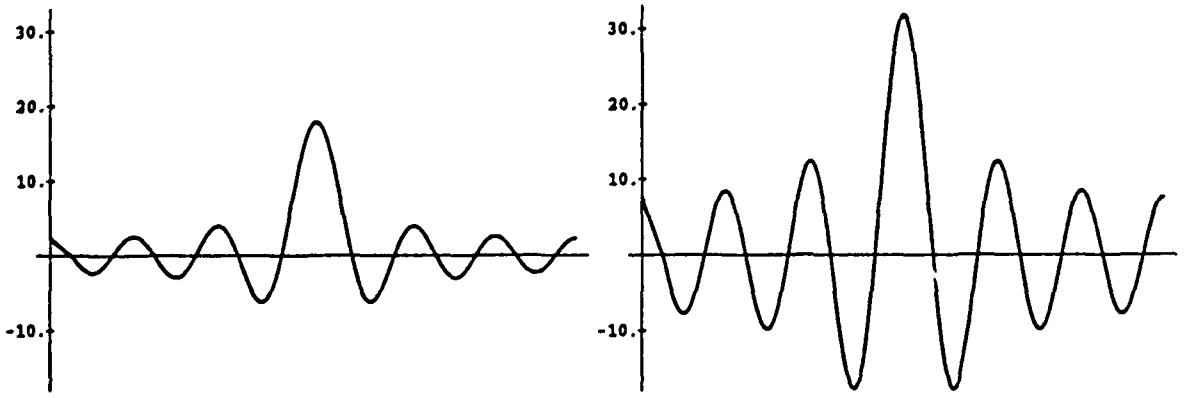


Figure 4. Kernels determined by the $\{b_k^\lambda\}$ multiplier sequence. On the left, $\lambda = .001$; at right $\lambda = .001$.

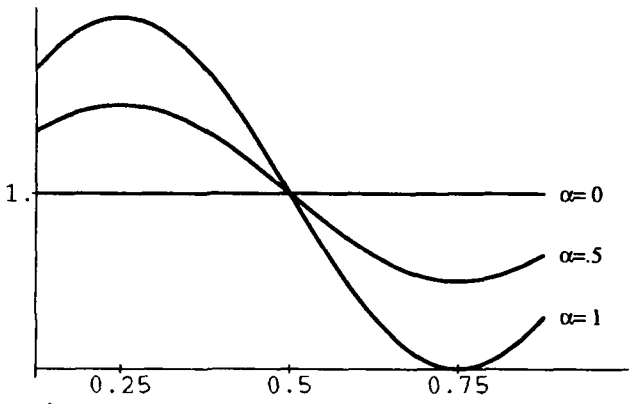


Figure 5. Graph of $w_L(x)$ for three different values of α.

The weight function $w(x)$ is selected by the researcher to emphasize the estimation of a particular region of the density. The function

$$w_L(x) = 1 + \alpha \sin(2\pi x),\ x \in [0,1],\ \alpha \in [0,1],$$

where α is a user-selected constant, can be used to increase the accuracy of the left side of the density estimate. A graph of $w_L(x)$ for three different values of α is shown in Figure 5.

Methods for choosing $\{b_k\}$ to minimize the weighted MISE are explored in Lock (1990) and Tarter, Freeman and Polissar (1990). In particular, both authors consider the case where $w(x)$ can be represented by a three-term Fourier expansion:

$$w(x) = \sum_{k=-1}^{1} w_k \exp\{2\pi i k x\}.$$

This is the case for the weight function $w_L(x)$. Utilizing this class of weight functions, methods are detailed in Lock (1990) for choosing an optimal truncation point for the raw estimator and for choosing more general multipliers as well. Simulation results show the efficacy of the methods in increasing the local accuracy of the estimator. Combined with the techniques described in the previous section, these methods offer a promising new approach to mixture decomposition.

4. References

Breiman, L., Meisel, W. and Purcell, E. (1977), Variable kernel estimates of multivariate densities, *Technometrics*, 19: 135-44.

Brunk, H.D. (1978), Univariate density estimation by orthogonal series, *Biometrika*, 65: 521-528.

Diggle, P.J. and Hall, P. (1986), The selection of terms in an orthogonal series density estimator, *Journal of the American Statistical Association*, 81: 230-233.

Fellner, W.H. (1974), Heuristic estimation of probability densities, *Biometrika*, 61: 485-492.

Hart, J.D. (1985), On the choice of a truncation point in Fourier series density estimation, *Journal of Statistical Computation and Simulation*, 21: 95-116.

Jones, M.C. (1991), On correcting for variance inflation in kernel density estimation, *Computational Statistics and Data Analysis*, 11: 3-15.

Kronmal, R.A. and Tarter, M.E. (1968), The estimation of probability densities and cumulatives by Fourier series methods, *Journal of the American Statistical Association*, 63: 925-952.

Lock, M.D. (1990), *Optimizing Density Estimates Based on Unweighted and Weighted Mean Integrated Square Error*, unpublished doctoral thesis, University of California, Berkeley.

Rudemo, M. (1982), Empirical choice of histograms and kernel density estimators, *Scandinavian Journal of Statistics*, 9: 65-78.

Silverman, B.W. (1986), *Density Estimation for Statistics and Data Analysis*, London and New York: Chapman and Hall.

Tarter, M.E. (1979), Biocomputational methodology - an adjunct to theory and applications, *Biometrics*, 35: 9-24.

Tarter, M.E. and Kronmal, R.A. (1976), An introduction to the implementation and theory of nonparametric density estimation, *American Statistician*, 30: 105-112.

Tarter, M.E. and Lock, M.D. (1988), A modular nonparametric approach to model selection, *Computing Science and Statistics-Proceedings of the 20th Symposium on the Interface*, (Editors, E. J. Wegman, D. T. Gantz and J. J. Miller) American Statistical Association, Alexandria, Virginia.

Titterington, D. M., Smith, A.F.M., and Makov, U.E. (1985), *Statistical Analysis of Finite Mixture Distributions*, Chichester, England: Wiley.

Wahba, G. (1981), Data-based optimal smoothing of orthogonal series density estimates, *The Annals of Statistics*, 9: 146-156.

Watson, G.S. (1969), Density estimation by orthogonal series, *The Annals of Mathematical Statistics*, 40: 1496-1498.

Neural Network Learning Systems: An Overview

John E. Moody
Department of Computer Science, Yale University
P.O. Box 2158 Yale Station, New Haven, CT 06520-2158
Internet: moody@cs.yale.edu, Phone: (203)432-1200

Abstract

Neural Network Learning Systems are models which are loosely inspired by notions of how self-organization and learning in biological systems might occur. These models are closely related to many established pattern recognition, classification, and regression techniques. Many exciting applications of these methods are being pursued, including nervous system modeling, robotics, signal processing, zipcode and speech recognition, speech production, computer backgammon, and financial analysis. This short paper is intended as a pointer to some of the vast literature covering this field.

Introduction

Statistics and neural network learning systems have much in common. Since neural network learning systems are being developed in a wide variety of contexts, statisticians are likely to find that the field offers many relevant and exciting avenues to explore. Furthermore, statisticians are well equipped to make significant contributions to this field.

As many excellent sources on neural networks are available, I will not attempt to provide a complete and detailed introduction to and overview of this field in the short amount of space available here. Rather, I shall make a few brief comments and provide some pointers to the literature.

Types of Learning

Neural network learning systems can be grouped into three categories: supervised learning, unsupervised learning, and reinforcement learning. Supervised and unsupervised learning systems include a number of standard statistical methods, while reinforcement learning systems are more similar to the way animal learning systems probably work.

Supervised learning systems are analogous to statistical classification and regression techniques. Unsupervised learning systems are analogous to such established statistical methods as density estimation, cluster analysis, principal components, multidimensional scaling, and so on. Neural network models often differ from their statistical analogs, however, in that they are usually nonlinear and are often real-time or adaptive.

Reinforcement learning differs from classification and regression in two ways. First, the system is not provided with explicit target values during training, but is simply given reward or penalty signals based upon performance. These reward/penalty signals may be delayed. Second, the behavior of the system has a random component which allows it to explore via trail and error.

Of these three types of learning, supervised learning algorithms have received the greatest amount of theoretical analysis and have enjoyed the widest ranging practical application. Unsupervised learning systems are also widely used. Reinforcement learning algorithms are the least widely applied and the most poorly understood. However, they are perhaps the most interesting.

Some Active Areas of Research

The range of active research topics in neural networks covers many disciplines. The following list is adapted from the list of oral sessions of the 1990 Neural Information Processing Systems conference program (see Lippmann, Moody, and Touretzky 1991):

Learning and Memory: Associative Memory, Classical Conditioning, Memory Organization and Indexing, Biophysics of Synaptic Change, etc.

Navigation and Planning: Animal Behavior, Robotics

Temporal and Real Time Processing: Timeseries Prediction, Music, Architectures for Real Time Adaptive Signal Processing and Control.

Learning and Generalization: Learning Algorithms & Architectures, Data Representations, Theory.

Visual Processing: Motion Processing, Color Constancy, Perceptual Grouping, Psychophysics, Organization of Visual Cortex, etc.

Speech Processing: Speech Recognition, Language Understanding

Signal Processing: Nonlinear Adaptive SP; Animal Perception, eg. Bat Echo Location; Signal Pattern Classification, eg. Dolphins Speech

Control: Animal Motor Control, eg. VOR; Robot, Vehicle, and Engine Control; Chemical Process Control, etc.

Unsupervised Learning: Competitive Learning, Hebb Rules, Clustering, Exploratory Projection Pursuit.

Self Organization: Development of Cortical and Dendritic Organization.

A Short Bibliography

Textbooks and Reprint Collections

James A. Anderson and E. Rosenfeld (1988). *Neurocomputing: Foundations of Research.* MIT Press. A tasteful and authoritative collection of classic papers.

John Hertz, A. Krogh, and R. Palmer (1991). *Introduction to the Theory of Neural Computation.* Addison Wesley. Excellent and comprehensive textbook written by physicists.

S.Y. Kung (1991). *Digital Neurocomputing.* Prentice-Hall. Excellent textbook with an engineering orientation, forthcoming.

Carver Mead (1989). Analog VLSI and Neural Systems. Addison Wesley. Beautifully written and accessible introduction to electronic neural circuit design.

David E. Rumelhart and J. McClelland (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vols 1 and 2.* MIT Press. A classic, psychologically-oriented.

David E. Rumelhart and J. McClelland (1988). *Explorations in Parallel Distributed Processing.* MIT Press. Includes software for Unix and MS/DOS.

Some Journals

Biological Cybernetics, Springer-Verlag, New York, Berlin. *IEEE Transactions on Neural Networks,* IEEE Press, Piscataway NJ. *International Journal of Neural Systems,* World Scientific Press, Teaneck NJ, Singapore. *Journal of Neural Network Computing,* Auerbach Publishers, Boston MA. *Network: Computation in Neural Systems,* Institute of Physics Publishing, Bristol England. *Neural Computation,* MIT Press, Cambridge MA. *Neural Networks,* Pergamon Press, Elmsford NY, Oxford.

NIPS Conference Proceedings

Advances in Neural Information Processing Systems, Vol. 3. R. Lippmann, J. Moody, and D. Touretzky, eds., Morgan Kaufmann, San Mateo CA, 1991.

Advances in Neural Information Processing Systems, Vols. 1 and 2. David Touretzky, ed., Morgan Kaufmann, San Mateo CA, 1989-90.

Generalization through Minimal Networks with Application to Forecasting

ANDREAS S. WEIGEND* and DAVID E. RUMELHART
Stanford University

Inspired by the information theoretic idea of minimum description length, we add a term to the usual back-propagation cost function that penalizes network complexity. From a Bayesian perspective, the complexity term can be usefully interpreted as an assumption about prior distribution of the weights. This method, called *weight-elimination*, is contrasted to *ridge regression* and to *cross-validation*. We apply weight-elimination to time series prediction. On the sunspot series, the network outperforms traditional statistical approaches and shows the same predictive power as multivariate adaptive regression splines.

We show how the effective number of parameters changes during training by analyzing the eigenvalue spectra of the covariance matrix of hidden unit *activations* and of the matrix of *weights* between inputs and hidden units. We find that the effective ranks of these matrices are equal to each other when a solution is reached, and interestingly also equal to the number of hidden units of the minimal network obtained with weight-elimination.

1 INTRODUCTION

Connectionist networks, also called brain-style computation or artificial neural networks, are ensembles of interconnected, usually nonlinear, units. The values of the connections between the units are estimated by a learning algorithm. This approach differs from traditional statistics both by the ubiquitous use of nonlinearities and by the sheer number of parameters.

Connectionist networks were first applied to time series prediction by Lapedes and Farber (1987). Whereas many researchers in the dynamical systems community only deal with noise free, computer generated time series, we focus on noisy, real world data of limited record length. In this case, the *problem of overfitting* can become serious.

*Present address of the first author:
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA

A priori, it is not clear what network size is required to solve a given problem. If the network is too small, it will not be flexible enough to emulate the dynamics of the system that produced the time series ("underfitting"). If it is too large, the excess freedom will allow the network to fit not only the signal but also the noise ("overfitting"). Both too small and too large networks thus give poor predictions in the presence of noise.

The key idea of *weight-elimination* is to add a penalty term accounting for network complexity to the usual cost function. The trade-off between performance and complexity is reflected in the sum of a performance and a complexity term. There is a u-shaped minimum between the extremes of having a too simple network that produces horrendous errors and a network with small errors on the training data that has enormous complexity. This sum is minimized through back-propagation (Rumelhart *et al.*, 1986).

1.1 ARCHITECTURE

Fig. 1 shows the architecture (the pattern of connectivity or topology) of a feed-forward network with one hidden layer. (For the time series we analyzed, one hidden layer sufficed.) The abbreviation d-n-1 denotes the following network:

- The *d input units* are given the past values $x_{t-1}, \ldots, x_{t-d}$ of the time series $\{x_t\}$.
- The input units are fully connected to *n nonlinear hidden units.*
- All hidden units are connected to a linear *output unit.*
- Output and hidden units have adjustable *biases* b.
- The *weights* can be positive, negative or zero.

The nonlinearities are located in the activation function (or transfer function) of the hidden units. The output (or response) of a hidden unit is called its **activation**. It is a composition of two operators: an affine mapping, followed by a nonlinear transformation. First, the inputs into a hidden unit h are

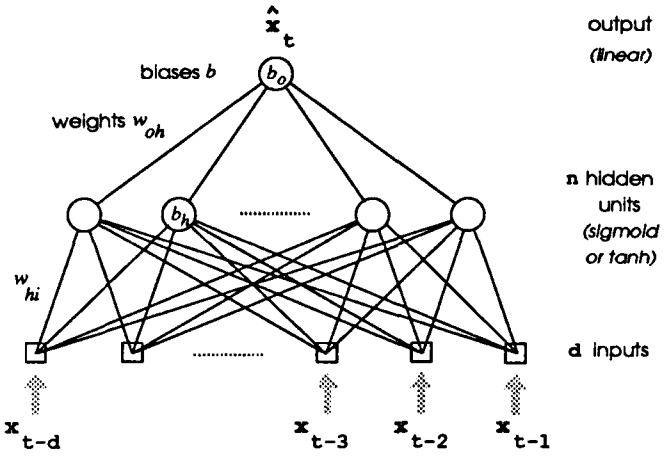


Figure 1: Architecture of a simple feed-forward network.

linearly combined, and a bias b_h (or offset) is added,

$$\xi_h = \sum_{i=1}^{d} w_{hi} x_i + b_h = \vec{w}_h \cdot \vec{x} + b_h \quad .$$

x_i stands for x_{t-i}, the value of input i, and w_{hi} is the weight between input unit i and hidden unit h.

Before turning to the second step, we give a geometric interpretation of ξ_h. A hidden unit only reponds to $\vec{w}_h \cdot \vec{x}$, the projection of the input vector $\vec{x} = (x_1, x_2, ..., x_d)$ onto the weight vector $\vec{w}_h = (w_{h1}, w_{h2}, ..., w_{hd})$. Changes in the input that are orthogonal to the direction of the weight vector have no effect on the activation of the hidden unit. The "equi-activation surfaces" (on which a hidden unit's activation is constant) are hyperplanes orthogonal to the direction of $\vec{w}_h$. The parameters of a hidden unit h can be characterized by

- a direction, $\vec{w}_h / ||\vec{w}_h||$,
- a scale parameter, $||\vec{w}_h||$, and
- a location parameter, $b_h / ||\vec{w}_h||$.

The symbol $|| \cdot ||$ denotes the (Euclidean) length of the vector.

The second step can be viewed as "piping" ξ_h through a nonlinear *activation function*. We here choose *sigmoid* (or logistic) units whose activations S_h are given by

$$S_h = S(\xi_h) = \frac{1}{1 + e^{-a\xi_h}} = \frac{1}{2}\left(1 + \tanh\frac{a}{2}\xi_h\right) \quad .$$

The gain a can be absorbed into weights and biases without loss of generality and is set to unity. The sigmoid performs a smooth mapping $(-\infty, +\infty) \rightarrow (0, 1)$.

The output of the network yields the prediction $\hat{x}_t$ as a weighted sum of the activations of the hidden units. To summarize, connectionist networks globally superimpose nonlinear functions to produce an output that can be viewed as a surface above the $(x_1, x_2, ..., x_d)$-plane of the inputs.

Viewed from the perspective of statistics, the network estimates the *conditional mean*,

$$\mathbf{E}\left[p(x_t | x_{t-1}, x_{t-2}, ..., x_{t-d}, \text{parameters})\right] \ ,$$

where p is the probability of an output value x_t for a given input vector. Note that this probability distribution p of the model, given inputs and parameters, is not to be confused with the probability distribution of the observed output given the predicted output, *i.e.*, the error model. Whereas the prior assumptions about measurement noise and model misspecification are reflected in a usually simple error model (we here assume the errors to be Gaussian distributed), the conditional mean depends on the data and can be fairly complicated.

1.2 EVALUATION

To evaluate and compare the predictive power of different algorithms, we use the *relative mean squared error* or **average relative variance**[1] of a set $\mathcal{S}$, $\texttt{arv}(\mathcal{S})$, defined as

$$\frac{\sum_{k\in\mathcal{S}} (\text{target}_k - \text{prediction}_k)^2}{\sum_{k\in\mathcal{S}} (\text{target}_k - \text{mean})^2} = \frac{1}{\widehat{\sigma}^2} \frac{1}{N} \sum_{k\in\mathcal{S}} (x_k - \widehat{x}_k)^2 \ . \tag{1}$$

The sum extends over the set $\mathcal{S}$ of pairs of the actual values (or targets, x_k) and predicted values ($\widehat{x}_k$). The averaging (division by N, the number of data points in a set $\mathcal{S}$) makes the measure independent of the size of the set. The normalization (division by $\widehat{\sigma}^2$, the estimated variance of the data) removes the dependence on the dynamic range of the data.

This quantity corresponds to the fraction of the squared error of the data that is not "explained" by the model. The symbol $\mathcal{S}$ in Eq. 1 indicates the data set used to compute the errors:

- **Training set.** This part of the data is used to estimate the parameters. The *fitting error* (or *approximation error* or *in-sample performance*) describes the fidelity to the data. If the model also needs to be determined, this set is further split into two sets. The first set, still called *training set*, is used for direct parameter estimation. The second set is referred to as *cross-validation set* and is used to determine the stopping point of the training process.[2]
- **Prediction set.** A certain part of the available data is strictly kept apart and only used to quote the expected performance in the future as *prediction error* or *out-of-sample performance*.

[1] In this paper, the term variance refers to sums of squared errors. In the statistics community, there also exists a narrower meaning, as in bias–variance tradeoff. We use the term variance to denote the sum of both the squared bias and the variance in the narrower sense. Incidentally, in the connectionist community, the term bias simply denotes an additive constant to the input of a unit.

[2] Our use of the term cross-validation differs from repeated leave-k-out procedures in that we often pick only one cross-validation set. Since our emphasis is on the *training* process, we use the validation set to monitor the progress during training.

Ultimately, we are interested in good performance for future predictions. Can we simply use the performance on the training set as an estimate of the predictive performance? Do we really need to set some data apart as prediction set?

It is well known that the in-sample performance can be a poor estimate of the out-of-sample performance, particularly in the presence of noise. For linear regression, it is sometimes possible to correct for the usually over-optimistic estimate. An example is to multiply the fitting error with $(N+k)/(N-k)$, where N is the number of data points and k is the number of parameters of the model (Akaike, 1970). It is not at all clear to what degree such approximations hold for nonlinear models, such as connectionist networks.

Now, even if we decided to ignore the issue of nonlinearities completely, what value should we use for k? Although the number of *available* parameters of the network is fixed, the number of *effective* parameters increases during training. Although all parameters are already present at the beginning of the training process, the number of parameters that are effective for solving the task is zero since they were just randomly initialized. We show in Section 3.1.2 how the number of effective parameters increases during training. This focus on learning is different from the typical assumption in statistics that the parameters are fully estimated at the time of model selection.

Up to now, we have ignored the question of how to determine the values of the weights and biases. In the next section, we turn to this question of parameter estimation and also to the problem of model selection in the presence of noise.

2 LEARNING

2.1 BACK-PROPAGATION

We use the error back-propagation algorithm by Rumelhart *et al.* (1986) to train the network: the parameters are changed by gradient descent on the cost surface over the weights and biases. On the whole, the problem of building a network that readily memorizes a set of training data has proven easier than expected. However, the problem of good generalization has proven more difficult.

2.2 GENERALIZATION

Connectionist networks are in essence statistical devices for inductive inference. There is a trade-off between two goals. On the one hand, we want such devices to be as general as possible so that they can learn a broad range of problems. This recommends large and flexible networks. On the other hand, the true measure of an inductive device is not how well it performs on the training examples, but how it performs on cases it has not yet seen, *i.e.*, its out-of-sample performance.

Too many weights of high precision make it easy for a network to fit the noise of the training data. In this case, when the network picks out the idiosyncrasies of the training sample, the generalization to new cases is poor. This *overfitting problem* is familiar in inductive inference, such as polynomial curve fitting. In the extreme, the polynomial fits the training points exactly and merely interpolates between them.

There are several potential solutions to this problem. We focus here on the so-called minimal network strategy. The underlying hypothesis is: if several networks fit the data almost equally well, the simplest one will on the average provide the best generalization. Evaluating this hypothesis requires *(1)* some way of measuring simplicity, and *(2)* a search procedure for finding the desired network.

The complexity of an algorithm can be measured by the length of its minimal description in some language. The old but vague intuition of Occam's razor—or dream—can be formalized as the *Minimum Description Length Criterion*: Given some data, the most probable model is the model that minimizes the sum

$$\textit{description length(data given model)} + \textit{description length(model)}.$$

This sum represents the trade-off between residual error and model complexity. The goal is to find a network that has the lowest complexity while fitting the data adequately. The complexity of a network is dominated by the number of bits needed to encode the weights. It is roughly proportional to the number of weights times the number of bits per weight. We focus here on the procedure of weight-elimination that tries to find a network with the smallest number of weights.

In Section 3.1.1, we compare weight-elimination to *cross-validation*: in that case, the cost function only consists of the error term. Overfitting is prevented by stopping the training early, *i.e.*, before the error reaches its asymptotic minimum. This leads to a network with fewer effective parameters than the total number of weights and biases (Section 3.1.2).

2.3 WEIGHT-ELIMINATION

In 1987, Rumelhart proposed several methods for finding minimal networks within the framework of back-propagation learning. A natural description of the complexity of a network uses quantities such as the size of the weights, the number of connections, the number of hidden units, the number of layers of hidden units, or the symmetries of the network. We focus on the method of weight-elimination that considers the size of the weights and the number of weights, and interpret the complexity term as a prior distribution of the weights.

2.3.1 Method

The idea is indeed simple in conception: add to the error a term which counts the number of parameters. We are looking for a differentiable function that is zero for zero weights and approaches a constant for large weights. We choose

$$\frac{w_i^2/w_0^2}{1+w_i^2/w_0^2}$$

w_0 is the scale for the weights. The subscript i in w_i simply enumerates the weights. The sum extends over all connections $\mathcal{C}$. Note that the biases do not enter the cost function: all offsets are a priori equally probable. (In the framework developed below, this corresponds to a non-informative prior: the probability density for the location parameter is flat.)

The performance term depends on the model for measurement errors. Since we assume that the errors are Gaussian distributed, the complete cost function is given by

$$\sum_{k\in\mathcal{T}}(\text{target}_k-\text{output}_k)^2+\lambda\sum_{i\in\mathcal{C}}\frac{w_i^2/w_0^2}{1+w_i^2/w_0^2} \qquad (2)$$

The first term, summed over the set of training examples $\mathcal{T}$, measures the performance of the network. The second term measures the size of the network. λ represents the relative importance of the complexity term with respect to the performance term.

The learning rule is to change the weights and biases according to the gradient of the *entire* cost function, continuously doing justice to the trade-off between error and complexity. This is different from the methods mentioned in Section 1.2 that consider a set of fixed models, estimate the parameters for each of them, and then compare between the models.

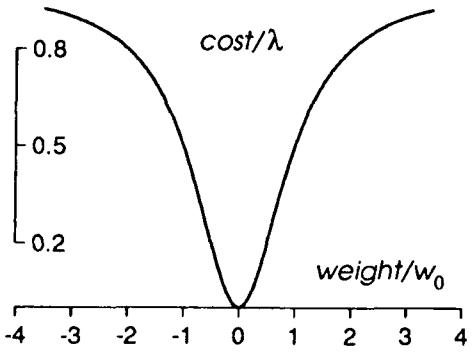


Figure 2: Complexity cost (in units of λ) of a weight as function of the size of the weight (in units of w_0).

The complexity cost is shown in Fig. 2 as function of w_i/w_0. For $|w_i| \gg w_0$, the cost of a weight approaches unity (times λ). This justifies the interpretation of the complexity term as a counter of significantly sized weights. For $|w_i| \ll w_0$, the cost is close to zero. "Large" and "small" are defined with respect to the scale w_0. It is a free parameter of the weight-elimination procedure. In our experience, choosing w_0 of order unity is good for activations of order unity. The effects of the choice for w_0 are discussed further in Section 2.3.3.

λ is dynamically adjusted in training. This dynamic increase, described in detail in Weigend *et al.* (1991), is related to the concept of iterated training as opposed to one-shot parameter estimation. At the beginning of the training, the weights are not useful yet, since they were just initialized randomly. Any significant cost for complexity would devour the whole network. Hence, λ starts at zero. The usual subsequent increase corresponds to attaching more importance to the complexity term or, from the perspective developed in the next section, to sharpening the peak around zero of the prior distribution of the probability density function of the weights.

2.3.2 Interpretation as Prior Probability

In a Bayesian framework, the complexity cost can be viewed as the negative logarithm of the prior probability of a weight.

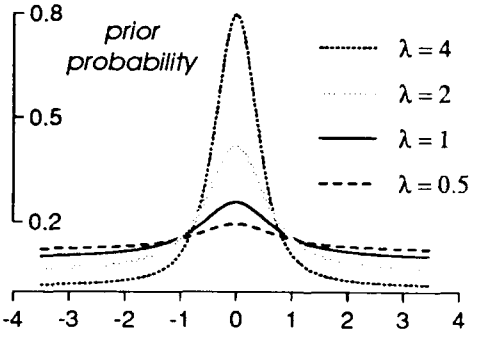


Figure 3: Prior probability of a weight as function of the size of the weight (in units of w_0), plotted for different values of λ.

In Fig. 3, we show the prior probability density function from which single weights of size w_i are drawn,

$$\text{prior} \propto \left[\exp\left(-\frac{w_i^2/w_0^2}{1+w_i^2/w_0^2}\right)\right]^{\lambda}$$

It is a mixture of a flat distribution and a bump around zero. Relevant weights are drawn from the flat distribution. Weights that are merely the result of noise are drawn from the bump centered on zero; they are expected to be small.

So far, we have only described our choice of the prior for a single weight. How do we get to the whole network? Assuming that the weights can be treated as independent, we simply sum over the connections in Equation 2.

2.3.3 Ridge Regresssion as Special Case

We here discuss the relationship of our method of weight-*elimination* to weight-*decay*, proposed by Hinton and by

Le Cun in 1987. In weight-decay, a small percentage of the weight is subtracted at each weight update,

$$\Delta w_i = (\text{weight change due to error back-prop.}) - \alpha w_i \quad .$$

This can be viewed as an exponential decay of the weight. It corresponds to a quadratic complexity cost ($\propto w_i^2$), known in the statistics community as *ridge regression*. It is contained in the weight-elimination scheme as the special case of large w_0. Weight-decay always prefers networks with many small weights. Weight-elimination prefers few large weights over many medium sized weights in the region where it acts as a counter. The scale parameter w_0 allows us to express a preference for many small weights (w_0 large) versus a few large weights (w_0 small). Depending on the dynamic range and the number of the units of the preceding layer, w_0 might be given different values for different layers of the network.

Expressing the cost of a weight as a prior can make it easier to interpret distributions that are not intuitive when viewed as penalty costs. Nowlan (1991) proposes a mixture of a few Gaussians as prior. This prior assumes that networks with weights around a few centers are more likely than networks with weights of many different values.

We now apply these methods to time series prediction.

3 SUNSPOTS

The sunspot series has served as a benchmark in the statistics literature. Within the paradigm of autoregression, different models differ in the specific choice of the primitives for the surface above the input space. In the simplest case, a single hyperplane approximates the data points. Such a *linear autoregressive model* is a linear superposition of past values.

The evaluation of the network model, however, is carried out by comparison to a *nonlinear* model, the *threshold autoregressive model* (TAR) by Tong and Lim (1980), see also Tong (1990). It has served as a benchmark for Subba Rao and Gabr (1984), for Priestley (1988), for Lewis and Stevens (1991), for Stokbro (1991), and for others.

The TAR model is globally nonlinear: it consists of two local linear autoregressive models. Tong and Lim found optimal performance for input dimension $d = 12$. They used yearly sunspot data from 1700 through 1920 for training, and the data from 1921 to 1979 to evaluation the predictions.

To make the comparison between network and TAR performance as close as possible, we use their exact data for training and evaluation, their choice for the input dimension, their error model and their evaluation criterion. The only remaining difference is the choice of the primitives for the surface.

3.1 LEARNING THE SERIES

In this section we analyze the in-sample learning behavior of the networks: first with a cross-validation set (needed to determine a stopping point when there is no complexity term in the cost function), then with weight-elimination. The out-of-sample performance will be analyzed in Section 3.2.

3.1.1 Internal Validation (Early Stopping)

The learning of the sunspot series of a 12-8-1 network is shown in Fig. 4 as a function of epochs. An epoch is one iteration of gradient descent in which the network sees each point from the training set once. Training with standard back-propagation (no weight-elimination) is displayed in the left panel. (The panel on the right hand side is discussed in Section 3.1.3.)

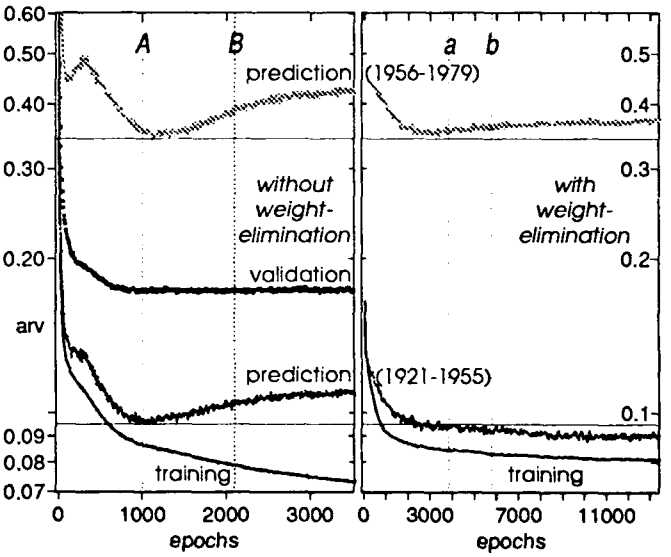


Figure 4: Learning curves of a 12-8-1 network. The average relative single-step prediction variances are given for the training sets, and early and late prediction sets (as well as for the cross-validation set for the network trained without weight-elimination on the left side). The vertical lines (A, B, a, b) indicate different stopping points. The average relative variance is normalized by the variance of the entire record, $\widehat{\sigma}^2 = 1535$.

The success in mastering the training set is indicated by the monotonic decrease of the lowest curve, indicating the *in-sample-performance* (or *fitting error*). To get a feeling for the non-stationarity of the time series, the prediction set was split in two parts, 1921-1955 and 1956-1979. On both prediction sets, the error first decreases, but then starts to increase: the network begins to use its resources to fit the noise of the training set. It starts to pick out properties that are specific to the training set, but not present in the prediction sets. This is an indication of overfitting.

When should the training should be stopped? Since prediction sets should not be used for this decision, a validation set is required to determine the end of the training process. To get a feeling for the effect of the sampling error by picking a specific training set–validation set combination, we investigated

several training set–validation set pairs.

The validation sets consisted of 22 years chosen at random from the time before 1920. Those points were removed in the corresponding training sets, reducing their size by 10%. The variations in performance due to different pairs of training and validation sets are larger than the variations due to different sets of random initial weights.[3] In the example given in Fig. 4, the validation set error approaches an asymptotic value. Since it does not increase, it is not entirely clear which set of weights should be taken. We thus compare in Section 3.2 the performance for two stopping points, *A* and *B*.

Some of the problems with early stopping through cross-validation are that *(1)* a part of the available training data cannot be used directly for parameter estimation, *(2)* the monitored validation set error often shows multiple minima as a function of training time (even in the simple linear case analyzed by Baldi and Chauvin, 1991), *(3)* the specific solution at the stopping points depends strongly on the specific pair of training set and validation set, and *(4)* the results are sensitive to the initial parameters.

Before comparing cross-validation with weight-elimination, we turn in the next section to the question how the effective number of parameters changes with training. We first focus on the activations of the hidden units, then on the weights between inputs and hidden units.

3.1.2 Effective Dimension of Hidden Units

Still within the framework of standard back-propagation, we analyze the change of the effective dimension of the hidden unit space during training by computing the spectrum of the eigenvalues of the **covariance matrix of the hidden unit activations**. The covariance C_{ij} corresponds to the two-point correlation between the activations of the two hidden units i and j, computed over the training set,

$$C_{ij} = \mathbf{E}\left[(S_i - \overline{S}_i)(S_j - \overline{S}_j)\right] \quad ,$$

where $\overline{S}_i = \mathbf{E}[S_i]$ is the mean activation of hidden unit i, taken over the set of training points. Since the covariance matrix is symmetric, $C_{ij} = C_{ji}$, its eigenvalues are real.

Linear correlation is appropriate, since the output linearly combines the hidden unit activations. The number of significantly sized eigenvalues is a measure of the effective dimension of the hidden unit space. It can be viewed as the effective rank of the covariance matrix. For linear networks, Baldi and Hornik (1992) use similar concepts.

[3]We chose the years for the validation sets randomly. An improvement might be to only consider random splits where the first and second moments (mean and variance) of the validation set match the training set. Another idea is to first train, stop and save several networks on different training-validation pairs, and then combine their individual predictions. The combination is done by freezing the weights and biases of the sub-nets, and only letting the few new combination weights adapt to the entire training set.

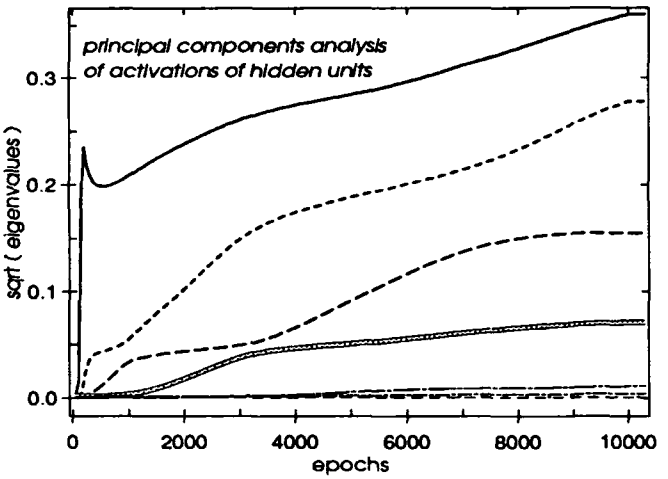


Figure 5: Eigenvalue spectrum of the covariance matrix of the hidden unit activations. The double line represents the fourth largest eigenvalue.

Fig. 5 shows the eigenvalue spectrum as a function of training time.[4] The eigenvalues correspond to the variances captured by the corresponding eigenvectors. In the figure, we plot the square root of the eigenvalues. They correspond to the standard deviations "explained" by the corresponding principal components. The figure shows that *gradient descent extracts one component after another*. This provides some justification for the whole strategy of oversized networks and early stopping: the dimension of the hidden unit space starts essentially at zero and then increases in training. The goal is to stop at just the right dimension.

So far, we have focused on eigenvalues derived from hidden unit *activations*. We now turn to eigenvalues derived from *weights*. We analyze the **singular value decomposition of the weight matrix** between inputs and hidden units. We decompose the 12 × 8 weight-matrix (inputs × hidden units) into two orthogonal matrices and one diagonal matrix and display the square root of the eigenvalues of that diagonal matrix in Fig. 6. At the beginning of the training, the eigenvalues just reflect the initialization of the weights.[5] As training proceeds, the dimension spanned by the weight space increases.

Both Fig. 5 and Fig. 6 only contain information from the *training set*. We now compare this information with the performance on the *prediction set*. In the run used for the eigenvalue calculations, the out-of-sample error reached its mini-

[4]The activations of the eight hidden units for each of the 209 points of the training set were recorded after every 50 epochs of training with learning rate 0.03. The overshooting of the largest principal component disappears if the hidden unit activations are multiplied with their corresponding output weights prior to computing the covariance matrix.

[5]We started the training with weights drawn from a uniform distribution over the interval [-0.03,0.03], corresponding to almost linear hidden units.

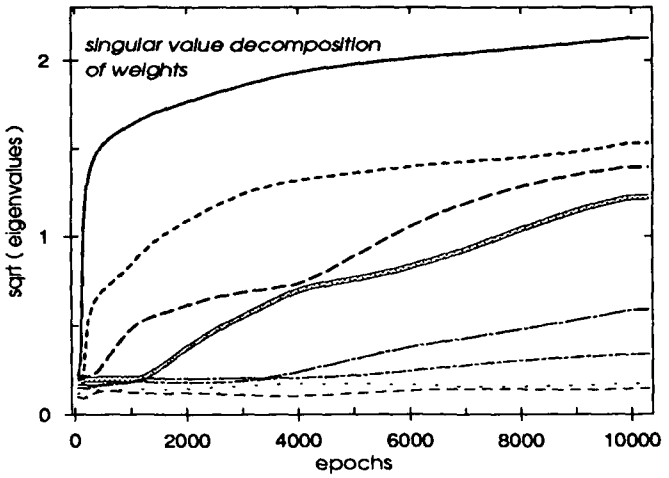


Figure 6: Eigenvalue spectrum of the singular value decomposed matrix of weights between input and hidden units.

mum around epoch 1000. At that point in training, the hidden units span an effectively three dimensional space. Extracting the fourth and subsequent eigenvalues hence corresponds to overfitting.

In the next section, we turn to training with weight-elimination. Interestingly, weight-elimination yields networks with three hidden units. This agreement between the effective dimension of hidden units at the onset of overfitting and the number of hidden units after weight-elimination is encouraging.

3.1.3 Weight-elimination

As in back-propagation without weight-elimination, we start with a network sufficiently large for the task. The training curve for back-propagation with weight-elimination is shown in the right panel of Fig. 4. Significant overfitting is avoided, even for training times four times as long. Since the entire training set is used, we are relieved from the uncertainty of a specific choice for a validation set. But we still have to decide when the asymptotic state is reached. The performance of two solutions (a and *b*) is compared in Section 3.2. It turns out that the exact stopping point is not important. In the first 5000 epochs, the procedure eliminated the weights between the output unit and five of the eight hidden units. Only three hidden units survived.

Weights from inputs to dead hidden units have no effect on the output. Since there is no reason for the network to pay a price for these weights, they subsequently get also eliminated. For time series prediction, weight-elimination acts as hidden unit elimination.

We analyzed the specific solution of the network that was stopped at point *b* and subsequently trained with a very small learning rate for a few epochs. The main contribution to the first hidden unit comes from x_{t-9}, to the second hidden unit from x_{t-2}, and to the third hidden unit from x_{t-1}. In contrast to the output weights, only very few of the weights from the input units to the active hidden units disappeared. (The parameters of the network are given in Weigend *et al.*, 1990.)

Predictions are obtained by adding the values of these three hidden units. The main encoding is performed by the nonlinear projection from the twelve dimensional input space onto the three dimensional hidden unit space.

3.2 PREDICTIONS AND COMPARISONS

So far, we have concentrated on the *learning* behavior of the network. Just obtaining a small network, however, is not an end in itself: the ultimate goal is to *predict* future values. In this section, we assess the predictive power of the network and compare it to other approaches. We first analyze single-step predictions and then turn to multi-step predictions.

3.2.1 Single-Step Prediction

The term *single-step prediction* (or *one-step-ahead prediction*) is used when all input units are given the actual values of the time series (as opposed to the predicted values). To assess the single-step prediction performance, we use the relative mean squared error (or average relative variance, `arv`), *defined in Equation* 1.

The weight-eliminated network gives

$$\texttt{arv}(\text{train}) = 0.082\ , \quad \texttt{arv}(\text{predict})_{1921-1955} = 0.086 \quad .$$

The corresponding values for the TAR model are

$$\texttt{arv}(\text{train}) = 0.097\ , \quad \texttt{arv}(\text{predict})_{1921-1955} = 0.097 \quad .$$

Comparing these numbers, we see that the single-step predictions of the network and the benchmark model are comparable. Despite this similarity, significant differences will appear for predictions further than one step into the future.

3.2.2 Multi-Step Prediction

There are two ways to predict further than one step into the future. We first present the results of iterated single-step predictions and subsequently turn to direct multi-step predictions. Most of the analysis so far applies to regression in general. Iterated predictions, however, are specific to time series.

In **iterated single-step** predictions, the predicted output is fed back as input for the next prediction and all other input units are shifted back one unit. Hence, the inputs consist of *predicted* values as opposed to observations of the original time series. The predicted value for time t, obtained after I iterations, is denoted by $\hat{x}_{t,I}$.

The prediction error will not only depend on I but also on the time $(t - I)$ when the iteration was started. We wish to obtain a performance measure as a function of the number

of iterations I that averages over the starting times. Since we want to fully exploit the standard prediction set range for the sunspot data from $t_{\text{BEGIN}} = 1921$ to $t_{\text{END}} = 1955$, we compute for each I the average

$$\frac{1}{\sigma^2}\,\frac{1}{t_{\text{END}} - (t_{\text{BEGIN}} - 1 + I)} \sum_{t_{\text{BEGIN}}-1+I}^{t_{\text{END}}} \left(x_t - \hat{x}_{t,I}\right)^2$$

This (average relative) prediction variance after I iterations is shown in Fig. 7. Only to indicate the spread of network performances, we give several network solution. The letters *A, B, a, b* refer to the different stopping points, shown in Fig. 4. The differences between the different network solutions are not significant.

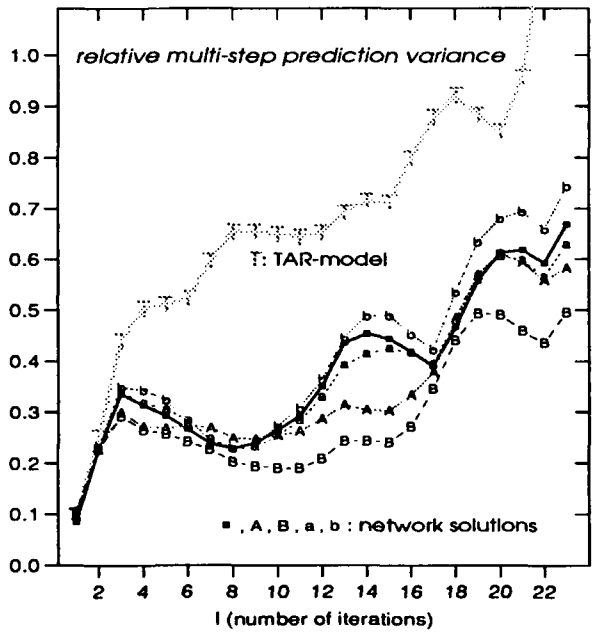


Figure 7: Relative prediction error after I iterations for the sunspot series. Gray T's give the performance of the TAR model. Black squares show the performance of the weight-eliminated network with three hidden units. The other curves indicate the performance of the network solutions from Fig. 4.

An alternative to this multi-step prediction by *iterated single-step* prediction is **direct multi-step** prediction: the network is trained to predict directly several steps ahead. On the sunspot data set, the prediction error for direct multi-step prediction was worse than the error for iterated single-step prediction.

In summary, although we took extreme care not to gain any unfair advantage over Tong and Lim (1980) (by taking the same input dimension, using identical data sets, minimizing the same sum of squared errors, *etc.*), the multi-step predictions were found to be significantly better: on average, the iterated prediction variances of the network were about half the iterated prediction variances of the TAR model. This concludes the comparison with the benchmark model.[6]

Subba Rao and Gabr (1984) apply a *bilinear model*[7] to the sunspot data and find an improvement of about 15% over the TAR model, both for single-step and iterated predictions. On predictions further than one step into the future, the networks outperform the bilinear model on average by 35% in mean squared error.

Stokbro (1991) uses a *weighted linear predictor* (WLP). In a WLP, each primitive is the product of a first order polynomial and a normalized Gaussian radial basis function. The predictor is the linear superposition of these primitives. Stokbro compares WLP with the network solution on the on the 1921 to 1946 prediction set given in Weigend *et al.* (1990). For one and two iterations, both methods have similar errors. When iterated more than twice, the network outperforms the WLP model.

Recently, Lewis and Stevens (1991) applied *multivariate adaptive regression splines* (MARS) by Friedman (1991) to the sunspot series. We find that the performance of MARS is very similar to the performance of the network. Given that the primitives of both schemes (sigmoids and splines) are smooth, and given that both approaches employ a regularization scheme that penalizes complexity, the similar performance is not astonishing but rather encouraging.

3.3 VARYING THE INPUT DIMENSION

Up to now, all predictions were based on information of the preceding twelve years. What happens if we vary the input dimension? When the number of input units is reduced, we expect the error to increase, at least at some stage. But when the number of input units is increased, two effects compete. On the one hand, more information becomes available, possibly allowing for better predictions. On the other hand, the higher the input dimension, the more sparsely distributed the training data. Will the networks be robust if more input units than necessary are present?

[6]The discrepancy between a negligible difference in single-step prediction accuracy and a factor of two for iterated predictions is interesting. A conjecture (from a discussion with Jerry Friedman) is the following:
Consider the single-step squared error decomposed into a squared bias and a variance. In this footnote—in contrast to the rest of the paper—the term variance refers to the spread of network solutions, see Geman, Bienenstock and Doursat (1991). Since a network is a more flexible model than a TAR models, the bias of the network is smaller than the bias of TAR. If iterating amplifies the squared bias more than the variance, the observed effect is explained.

[7]In addition to linear autoregression (terms proportional to x_{t-i}), Subba Rao and Gabr allow terms proportional to the forecasting errors ϵ_{t-j} as well as terms proportional to the product $x_{t-k}\epsilon_{t-l}$ (bilinear interactions). In the framework of connectionist networks, arbitrary interactions between lagged inputs x_{t-k} and past prediction errors ϵ_{t-l} are modeled by enhancing the usual input with a set of units representing ϵ_{t-l}. Such a network can learn to extract possibly nonlinear responses to outside shocks.

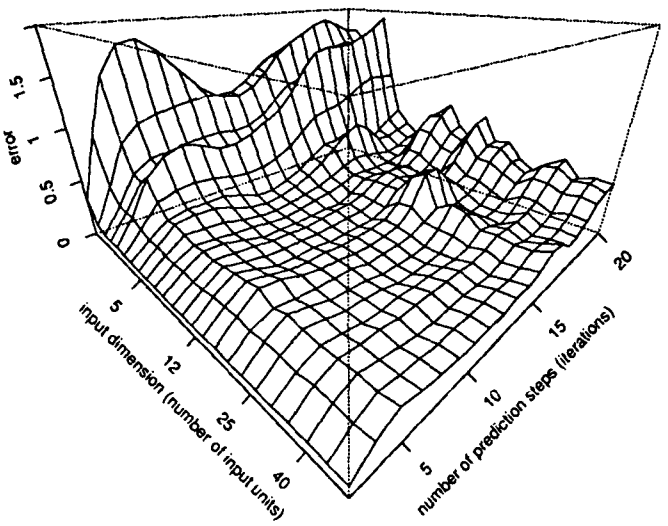


Figure 8: Prediction error for sunspots as function of the number of inputs and forecasting time.

The prediction error for iterated single-step predictions (for the 1921–1955 set) is shown in Fig. 8 as surface above the number of inputs and the prediction time into the future.

Networks with one input unit of lag one already manage to capture two thirds of the single-step variance, reducing it to 0.33. The solution is practically linear with an offset. Networks with two input units reduce the relative mean squared error to 0.17. They begin to use the available nonlinearities. With increasing number of input units, the error reaches a roughly constant value. The performance does not degrade with input dimension several times larger than necessary: the networks ignore irrelevant information.

To summarize, we use a procedure called weight-elimination that addresses the related problems of network size and overfitting by dynamically eliminating weights during training. In this paper, we focus on the time series of yearly sunspot averages from the year 1700 onward. On iterated predictions into the future, the network performance turns out to be very similar to MARS and significantly better than other models.

We close with two references to further examples of networks for time series prediction. In Weigend *et al.* (1990), we analyze a time series from a computational ecosystem and show that connectionist networks can predict the utilization of the resources of the ecosystem for hundreds of steps into the future. And in Weigend *et al.* (1991), we apply networks to the prediction of the notoriously noisy foreign exchange rates and show that the key to a solution there is selection of the relevant variables through weight-elimination.

We thank Jerry Friedman and Art Owen for discussions.

4 REFERENCES

Akaike, Hirotugo. **Statistical predictor identification.** *Ann. Institute of Statistical Mathematics*, 22:203–217, 1970.

Baldi, Pierre and Chauvin, Yves. **Temporal evolution of generalization during learning in linear networks.** Submitted to *Neural Computation*, 1991.

Baldi, Pierre and Hornik, Kurt. **Back-propagation and unsupervised learning in linear networks.** In Chauvin, Y. and Rumelhart, D. E., editors, *Backpropagation and Connectionist Theory*. Lawrence Erlbaum, 1992.

Friedman, Jerome H. **Multivariate adaptive regression splines.** *The Annals of Statistics*, 19:1–141 (with discussion), 1991.

Geman, Stuart, Bienenstock, Elie, and Doursat, René. **Neural networks and the bias/variance dilemma.** Submitted to *Neural Computation*, 1991.

Lapedes, Alan S. and Farber, Robert M. **Nonlinear signal processing using neural networks: prediction and system modelling.** Technical Report LA-UR-87-2662, Los Alamos National Laboratory, 1987.

Lewis, Peter A. W. and Stevens, J. G. **Nonlinear modeling of time series using multivariate adaptive regression splines (MARS).** Submitted to *Journal of the American Statistical Association*, 1991.

Nowlan, Steven J. *Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures.* PhD thesis, CMU (Computer Science), 1991.

Priestley, Maurice B. *Non-linear and Non-stationary Time Series Analysis.* Academic Press, 1988.

Rumelhart, David E., Hinton, Geoffrey E., and Williams, Ronald J. **Learning internal representations by error propagation.** In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing*, pages 318–362. MIT Press, 1986.

Subba Rao, T. and Gabr, M. M. *An Introduction to Bispectral Analysis and Bilinear Time Series Models*, volume 24 of *Lecture Notes in Statistics.* Springer, 1984.

Stokbro, Kurt. **Predicting chaos with weighted maps.** Technical Report 91/10 S, Nordita, Copenhagen, 1991.

Tong, Howell and Lim, K. S. **Threshold autoregression, limit cycles and cyclical data.** *Journal Royal Statistical Society B*, 42:245–292, 1980.

Tong, Howell. *Non-linear Time Series: a Dynamical System Approach.* Oxford University Press, 1990.

Weigend, Andreas S., Huberman, Bernardo A., and Rumelhart, David E. **Predicting the future: a connectionist approach.** *International Journal of Neural Systems*, 1:193–209, 1990.

Weigend, Andreas S., Huberman, Bernardo A., and Rumelhart, David E. **Predicting sunspots and exchange rates with connectionist networks.** In Casdagli, M. and Eubank, S. G., editors, *Nonlinear Modeling and Forecasting*. Addison-Wesley, 1991.

Probabilities on Complex Pedigrees; the Gibbs Sampler Approach

Elizabeth Thompson*

Department of Statistics, GN-22,
University of Washington
Seattle, WA 98195

Abstract

The analysis of complex familial traits requires the computation of likelihoods for complex genetic models on extended and/or complex pedigrees. This challenge has defeated conventional computational algorithms, but the pedigree Gibbs sampler provides an effective method of Monte Carlo evaluation of the required probabilities and likelihood ratio functions.

KEY WORDS: Genetic models; Complex pedigrees; Conditional independence structure. Monte Carlo summation; Importance sampling; Gibbs sampler;

1 Introduction

The objective is to compute the probability of trait data observed on some subset of the related members of a specified pedigree structure, or the probability of underlying genotypic configurations on the pedigree conditional upon trait data, in either case under some specified genetic model for the trait. Often in the analysis of complex familial traits the genetic models required will be complex, with several genetic and non-genetic factors contributing to the observed trait. On the other hand, the pedigrees on which such traits are analysed are not necessarily complex, even when extended pedigrees of several hundred individuals are used to help to ensure genetic homogeneity of the trait in question. However, the pedigrees of genetically isolated populations are complex, and in this paper we shall address the question of likelihood computations on complex pedigrees. By contrast, we shall restrict attention to simple genetic models, mainly for expository convenience.

For computational purposes a pedigree is most easily specified by giving, for every individual, the unique individual identifiers of his/her mother and father. Graphically, a complex pedigree is represented most easily as a

*Research supported in part by NSF grant BSR-8921839.

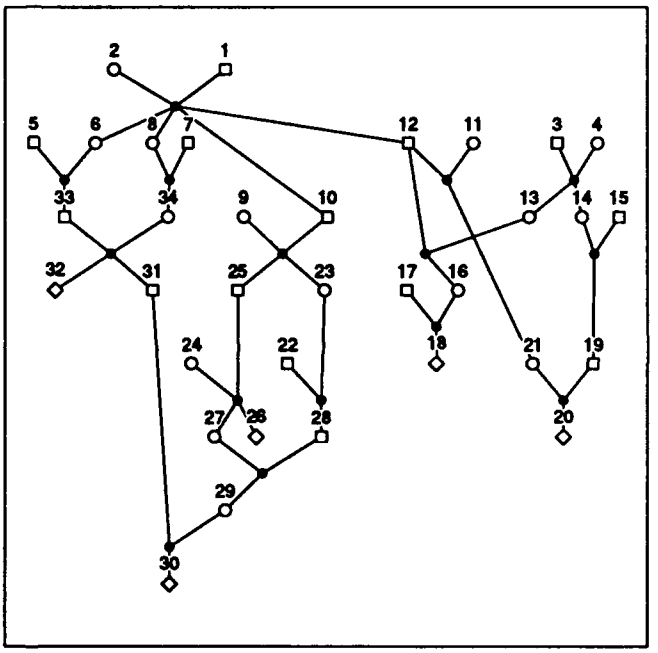


Figure 1: An example pedigree; from Thompson (1986).

marriage node graph: an arc joins each individual to his parents' marriage (or mating) and to each of his own marriages. Figure 1 shows a small example pedigree in marriage node graph form. The pedigrees on which we require probabilities and likelihoods are far larger and more complex than that of figure 1: one example is discussed in section 6. A general characteristic of the extended complex multi-generation pedigrees of genetic isolates is that data are available normally only for current individuals, the lower fringe of the pedigree, and often not even for all such individuals. Although problems of graph-theoretic type arise in the development of algorithms for computations on pedigrees, pedigrees are not arbitrary graphs. Chronology and marriage preferences often result in substantial ordering and regularity

(figure 2). Note that throughout this paper it is assumed that the pedigree itself is known correctly.

There are two main classes of purpose in computing the probability of trait data observed on a specified pedigree structure under a specified genetic model. The first is where the genetic model for the trait is known. In this case, of interest are genetic counselling probabilities: probabilities that specified individuals carry genes that cause them to be at risk of developing a certain disease, or, jointly with a specified partner, have offspring at risk of developing the disease. Related to these probabilities are ancestral inference probabilities: posterior probabilities, given the trait data, that the genes now exhibiting effects in current individuals entered the pedigree through certain founder individuals, and/or descended via certain ancestral paths. The second purpose of probability computation is where the genetic model is not known, and the probability of trait data under some model is required as the likelihood of that model, to aid in likelihood inference of the true genetic model underlying the trait. Examples of each of these two classes of problem will be given in section 6.

2 Genetic models

The elements of genetic models are straightforward: genes exist, genes segregate (are copied) from parents to offspring, and the types of genes carried by an individual influence observable trait characteristics.

a) Genes exist: For the simplest genetically determined traits, each individual carries two genes, which many be of any of a number of types. The simplest possibility is that there are two types, say A and B, and in this case an individual has three possible combinations of types of genes, or *genotypes*, AA, AB or BB. The frequencies, or prior probabilities, for the types of genes are parameters of the model. Genotypes for an individual i will be denoted G_i. Individuals whose parents are not pedigree members, *founders*, have genotype probabilities $P_\theta(G_i)$, under a genetic model indexed by parameters θ.

b) Genes segregate: in modern terminology, Mendel's first law (1866) states that one of the two genes that an individual carries for a trait is a copy of a random (equiprobably chosen) one of the two in his father, the other a copy of a random one of the two in his mother, and that a random one of the two he carries will be copied to each child, independently for each child. For any individual i with parents M_i and F_i, Mendel's law provides the segregation probabilities $P(G_i|G_{M_i}, G_{F_i})$. For example, $P(G_i = AA|G_{M_i} = AA, G_{F_i} = AB) = 1/2$. Thus, segregation of the genes for simple Mendelian traits do not provide any additional unknown parameters of the model. However, as described below, one of the most important parameters of modern genetic analyses is a segregation parameter.

c) Genes influence traits: For the purposes of this paper, we assume such influences can be summarised by probabilities $P_\theta(y_i|G_i)$ where y_i is the observed (qualitative or quantitative) trait value of individual i. Such probabilities are known as *penetrance probabilities*, and their specification may involve unknown parameters.

We shall subsume all the parameters of the genetic model into the parameter vector θ, and use $P_\theta(\cdot)$ to denote probabilities under the model. The total set of genotypes on a pedigree will be denoted $\mathbf{G}$, and of observed phenotypes $\mathbf{y}$.

Many genetic analyses are concerned with *linkage analysis*, the objective being inference of the location within the genome of genes controlling a trait of interest, by observing cosegregation with DNA markers whose position in the genome is known. The recombination frequency between trait genes and genes determining a marker trait is the frequency with which genes for the two traits passed on by an individual, i, to an offspring derive from the two different parents of i. If the genes determining the two traits are located close together in the genome, this frequency is close to 0, while if they are far apart, or on different chromosomes, the frequency is 1/2. Thus the recombination frequency, or linkage parameter, r, ranges from 0 to 1/2, and determines the segregation probability $P_\theta(G_i|G_{M_i}, G_{F_i})$, where now the genotypes refer to the combined genotypes for both trait and marker. Estimation of r, or testing of the hypothesis $r = \frac{1}{2}$, is a frequent objective of genetic analyses; an example will be given in section 6.

3 Exact likelihood computation

The probability of data observed on the pedigree, computed under a specified genetic model, or the likelihood of that model, can be written as

$$L(\theta) = P_\theta(\mathbf{y}) = \sum_{\mathbf{G}} P_\theta(\mathbf{y}|\mathbf{G}) P_\theta(\mathbf{G}) \qquad (3.1)$$

For simple genetic models

$$P_\theta(\mathbf{y}|\mathbf{G}) = \prod_i P_\theta(y_i|G_i) \qquad (3.2)$$

where the penetrance probability is interpreted as unity for individuals i for whom no trait data are observed, and

$$P_\theta(\mathbf{G}) = \prod_i P_\theta(G_i|G_{M_i}, G_{F_i}) \qquad (3.3)$$

where M_i and F_i are the parents of i, and for founders i the segregation probability is to be interpreted as the population probability $P_\theta(G_i)$. Thus neither term within the sum (3.1) is difficult to compute. The difficulty lies only in the summation over all genotypic configurations $\mathbf{G}$ on the pedigree.

The most successful algorithms previously developed for evaluating (3.1) were first described by Hilden (1970), Elston and Stewart (1971), and Heuch and Li (1972). The approach was generalised by Cannings, Thompson and Skolnick (1978). In quite other contexts, similar approaches have more recently been developed by Lauritzen and Spiegelhalter (1988). The method rests on the conditional independence structure of genetic models: given the types of the genes carried by the spouses, parents and offspring of an individual, the genotype and data observation on that individual are independent of all other data and genotypes in the pedigree. This conditional independence, guaranteed by the model specifications of section 2, is fundamental also to the Gibbs sampling approach of section 5.

Another expression of the same fact is that, conditional on the types of the genes carried by individuals who constitute a *cutset*, dividing the pedigree into two or more disjoint components, the trait data observed on the each component, and on the cutset, are jointly independent. In our small example pedigree (figure 1), $\{31, 25, 23\}$ is a cutset; conditional on the cutset genotypes, data observed on the three sets, $\{22, 24, 26, 27, 28, 29, 30\}$, $\{31, 25, 23\}$ and the remainder, are independent. The pair $\{13, 21\}$ is also a cutset, dividing $\{3, 4, 14, 15, 19, 20\}$ from the remainder of the pedigree. A single individual (e.g. 12) can be a cutset.

The conditional independence exhibits itself in (3.1) through the fact that terms involving members of one component of a pedigree, for example $\{3, 4, 14, 15, 19, 20\}$ involve additionally only the relevant cutset members—the spouses, parents or offspring of some member of the component (in this example $\{13, 21\}$). Thus the summation (3.1) can be accomplished sequentially through the pedigree, considering only a few individuals at each stage and producing at each stage a real-valued function defined on the possible genotypic configurations of a cutset. These functions were called R-functions by Cannings, Thompson and Skolnick (1978). Specifically, summation of all terms involving individuals 3 and 4 can be accomplished for each genotypic configuration of the cutset $\{13, 14\}$. Then this R-function can be incorporated into summation over the possibilities for individuals 14 and 15, producing a new R-function on $\{13, 19\}$, and thirdly summation over 19 and 20 produces an R-function on $\{13, 21\}$. In this way it is possible to work through an entire complex pedigree, accomplishing finally the summation (3.1).

This method has proved successful in analysing traits on a number of large and complex pedigrees, but it is severely bounded in a way that seems unlikely to be much relieved by increased computing capacity. For the simplest genetic models, an individual has three possible genotypes; for the simplest linkage model there are ten. Where there are k possible genotypes, for a cutset of n individuals, there are k^n possible genotypic configurations; the R-function has k^n discrete values. Nor can the problem be resolved by including more individuals in each sequential summation; if N cutset and non-cutset individuals are involved in a given step, there are (at least in principle) k^N terms to be considered in the summation. Since 1978 the increase in computing power has enabled us to extend from the initial programs with cutsets of size 8 ($3^8 = 6561$) to cutsets of size 14 (or 13 with double precision; $3^{13} = 1,594,323$). But large complex pedigrees cannot always be resolved with cutsets of size no more than 13, and for a genetic model with ten genotypes even cutsets of size 8 remain impossible.

For more details of the peeling approach, Cannings, Thompson and Skolnick (1978) give the theory, and examples are discussed by Thompson (1986). We have given here only a sufficient description to demonstrate the need for other approaches and to provide a basis for the discussion of section 7. One later requirement will be the probabilistic interpretation of an R-function. Where the component for which summation has been accomplished (the *peeled set*, $\mathbf{Q}$) contains no parents of cutset individuals, the interpretation is straightforward; each term is simply the probability of data observed on $\mathbf{Q}$ conditional on that particular genotypic configuration on the cutset $\mathbf{C}$. For example, in figure 1, with $\mathbf{Q} = \{22, 24, 26, 27, 28, 29, 30\}$ each term of $R(G_{31}, G_{25}, G_{23})$ is the probability of data observed on $\mathbf{Q}$ given the particular configuration of genotypes on $\mathbf{C} = \{31, 25, 23\}$. Where parents of $i \in \mathbf{C}$ are in $\mathbf{Q}$, the probability is joint with the relevant genotype of i, and it is further important to recognise that the R-function incorporates only probabilities resulting from genealogical relationships within $\mathbf{Q}$. Thus

$$R(G_{13}, G_{21}) = \text{prob}^*(\text{data on } \{3, 4, 14, 15, 19, 20\}, G_{13}|G_{21}) \tag{3.4}$$

where prob* denotes the fact that this probability is computed only on the subpedigree consisting of $\mathbf{Q}$ and $\mathbf{C}$—that is, it incorporates that 13 is the great aunt of 21's offspring 20, but not that she is also the spouse of 21's father, 12, through whom there is also dependence

by virtue of any data on their joint descendants 16 and 18. Note also that if one parent of a cutset individual is in **Q** the other must either be in **Q** or else in **C**.

4 A Monte Carlo approach

An alternative approach to the summation (3.1) is a Monte Carlo one. Simulation from $P_\theta(\mathbf{G})$ is straightforward, by assigning genes to the founders of the pedigree, with appropriate probabilities, and simulating the Mendelian segregation of those genes. A set of M realisations $\{ \mathbf{G}^{(i)} : i = 1, \ldots, M \}$ provides a simple Monte Carlo estimate

$$\frac{1}{M} \sum_{i=1}^{M} P_\theta(\mathbf{y}|\mathbf{G}^{(i)}). \tag{4.1}$$

However, this estimate is useless on a large or complex pedigree, particularly where data are confined to the lower part of the pedigree. There are huge numbers of genotypic configurations on a pedigree. Normally only a minute proportion are even compatible with the data (give a non-zero value of (3.2)), and even these are likely to give negligible contribution to the likelihood (3.1).

Another proposal was made by K. Lange in Ott (1979): namely, to rewrite (3.1) as

$$\mathrm{L}(\theta) = \sum_{\mathbf{G}} P_\theta(\mathbf{y}|\mathbf{G}) \frac{P_\theta(\mathbf{G})}{P_{\theta_0}(\mathbf{G})} P_{\theta_0}(\mathbf{G}) \tag{4.2}$$

and to simulate $\{ \mathbf{G}^{(i)} : i = 1, \ldots, M \}$ from $P_{\theta_0}(\mathbf{G})$, giving a Monte Carlo estimate

$$\frac{1}{M} \sum_{i=1}^{M} P_\theta(\mathbf{y}|\mathbf{G}^{(i)}) \frac{P_\theta(\mathbf{G}^{(i)})}{P_{\theta_0}(\mathbf{G}^{(i)})} \tag{4.3}$$

This is likewise not effective on a large pedigree for $P_{\theta_0}(\mathbf{G})$ will generate realisations even less related to $P_\theta(\mathbf{y}|\mathbf{G})$ than does $P_\theta(\mathbf{G})$, but it contains the seeds of two key ideas. The first is that of sampling according to some other distribution and reweighting the summand accordingly—that is, of importance sampling. The second is that by simulation at a single θ_0, Monte Carlo estimates of an entire function $L(\theta)$ are, in principle, obtainable.

Pursuing the idea of importance sampling, what would be the optimal simulation distribution? Since $P_\theta(\mathbf{y}|\mathbf{G})P_\theta(\mathbf{G})$ is, as a function of $\mathbf{G}$, proportional to $P_\theta(\mathbf{G}|\mathbf{y})$, we require something of similar form to $P_\theta(\mathbf{G}|\mathbf{y})$, for example $P_{\theta_0}(\mathbf{G}|\mathbf{y})$ for some θ_0 similar to θ. Unfortunately, $P_{\theta_0}(\mathbf{G}|\mathbf{y})$ cannot be written down explicitly; evaluation is equivalent to that of (3.1). And how do we sample from this distribution? Deferring the latter question, consider first the form that the estimate would take. The likelihood (3.1) or (4.2) is also

$$L(\theta) = \sum_{\mathbf{G}} \frac{P_\theta(\mathbf{y}|\mathbf{G})P_\theta(\mathbf{G})}{P_{\theta_0}(\mathbf{G}|\mathbf{y})} P_{\theta_0}(\mathbf{G}|\mathbf{y}) \tag{4.4}$$

and thence, by Bayes theorem,

$$L(\theta) = \sum_{\mathbf{G}} \frac{P_\theta(\mathbf{y}|\mathbf{G})}{P_{\theta_0}(\mathbf{y}|\mathbf{G})} \frac{P_\theta(\mathbf{G})}{P_{\theta_0}(\mathbf{G})} P_{\theta_0}(\mathbf{y}) P_{\theta_0}(\mathbf{G}|\mathbf{y}) \tag{4.5}$$

Now $P_{\theta_0}(\mathbf{y})$ is also unknown; again an evaluation equivalent to (3.1) is required. However, by definition, this probability is the likelihood $L(\theta_0)$. In likelihood inference, likelihood ratios are sufficient. Thus rewriting (4.5) as

$$\frac{L(\theta)}{L(\theta_0)} = \sum_{\mathbf{G}} \frac{P_\theta(\mathbf{y}|\mathbf{G})}{P_{\theta_0}(\mathbf{y}|\mathbf{G})} \frac{P_\theta(\mathbf{G})}{P_{\theta_0}(\mathbf{G})} P_{\theta_0}(\mathbf{G}|\mathbf{y}) \tag{4.6}$$

we can obtain a Monte Carlo estimate of the likelihood ratio $L(\theta)/L(\theta_0)$,

$$\frac{1}{M} \sum_{i=1}^{M} \frac{P_\theta(\mathbf{y}|\mathbf{G}^{(i)})}{P_{\theta_0}(\mathbf{y}|\mathbf{G}^{(i)})} \frac{P_\theta(\mathbf{G}^{(i)})}{P_{\theta_0}(\mathbf{G}^{(i)})} \tag{4.7}$$

where now the $\mathbf{G}^{(i)}$ are realisations from $P_{\theta_0}(\mathbf{G}|\mathbf{y})$. Moreover, from a single set of realisations at θ_0 we can obtain estimates of $L(\theta)/L(\theta_0)$ for many different values of θ.

Note also that if the $(\theta : \theta_0)$ difference lies only in the segregation probabilities, it is not even necessary to be able to compute $P_\theta(\mathbf{y}|\mathbf{G})$. If $P_\theta(\mathbf{y}|\mathbf{G}^{(i)}) = P_{\theta_0}(\mathbf{y}|\mathbf{G}^{(i)})$, (4.7) reduces to

$$\frac{1}{M} \sum_{i=1}^{M} \frac{P_\theta(\mathbf{G}^{(i)})}{P_{\theta_0}(\mathbf{G}^{(i)})} \tag{4.8}$$

where the data $\mathbf{y}$ now enter only through the sampling of the $\mathbf{G}^{(i)}$ from $P_{\theta_0}(\mathbf{G}|\mathbf{y})$. Thus, in particular, linkage analysis for complex traits is possible (Guo and Thompson, 1991b); an example is given in section 6.

The development of this section presupposes the availability of realisations from the global posterior conditional distribution of genotypic configurations on the pedigree, conditional on the observed data. The next section will complete the picture by describing how the pedigree Gibbs sampler provides such realisations. Obtaining such realisations we solve also the problem of estimation of risk probabilities on pedigrees, for such probabilities are precisely specified marginals of this conditional distribution. Estimates are thus provided by relative frequency counts in the realisations. An example is given in section 6.

5 The pedigree Gibbs sampler

Thus the remaining task is to obtain the realisations from $P_{\theta_0}(\mathbf{y}|\mathbf{G})$ required for (4.7) above. This can be achieved by a Gibbs sampler (Hastings, 1970) on the genotypes of the individuals of the pedigree. The Gibbs sampler has recently become widely used in image analysis (Geman and Geman, 1984), a situation in which the global conditional distribution of true image conditional on data observations cannot be computed nor directly simulated from, but in which the local conditional distributions are easily specified and easy to simulate from. This is the situation in pedigree analysis. Although $P_{\theta_0}(\mathbf{G}|\mathbf{y})$ cannot be evaluated,

$$P_{\theta_0}(G_i|\mathbf{y}, \mathbf{G}_{-i}) = P_{\theta_0}(G_i|y_i, \mathbf{G}_{\mathbf{N}_i}) \qquad (5.1)$$

where $\mathbf{G}_{-i}$ denotes the genotypes of all individuals other than i, and $\mathbf{G}_{\mathbf{N}_i}$ the genotypes on the neighbours $\mathbf{N}_i$ of i, which, from section 3, comprise his parents, spouses and offspring. Moreover, the probability (5.1) is proportional, as a function of G_i, to the product of penetrance probability $P_{\theta_0}(y_i|G_i)$ and the segregation (or founder) probabilities for triplets (j, M_j, F_j) for $i = j$, and for $i = M_j$ or F_j.

Thus one possible implementation of the Gibbs sampler is as follows:

Start from a genotypic configuration on the pedigree for which $P_{\theta_0}(\mathbf{G}|\mathbf{y}) > 0$.

Take a random permutation of the individuals in the pedigree. For each individual, according to this permutation, update G_i in the current configuration $\mathbf{G}$ by sampling from the local conditional distribution (5.1). We refer to this procedure as one random scan of the pedigree.

We now perform repeated random scans, taking a new permutation of the pedigree members each time. This process defines a Markov chain on the space of genotypic configurations for which $P_{\theta_0}(\mathbf{G}|\mathbf{y}) > 0$, and $P_{\theta_0}(\mathbf{G}|\mathbf{y})$ is an equilibrium distribution of this Markov chain. Provided the Markov chain is irreducible, the configuration after successive scans converges in distribution to the required global conditional distribution, and dependent realisations $\mathbf{G}^{(i)}$ can be obtained by sampling the chain after a sufficient initial period for the convergence in distribution to be approximately accomplished.

There are many details of the above procedure not fully detailed here. First a feasible initial configuration must be found; this is usually not hard. Second, the chain must be irreducible; in general this is a problem, but irreducibility does obtain for the examples of this paper (among many others). Third, decisions must be made as to the sampling of the chain; here our guidelines are very preliminary. In cases where the convergence is likely to be slow, such as the Hutterite example below, we have used an initial period of 4000 scans. Where the convergence in distribution of the chain realisations can be shown to be faster, as in the linkage example below, an initial period of 400 scans suffices.

Thereafter the chain should be sampled at a frequency that depends on the trade-off between the autocorrelation between successive scans, and the amount of computation to be performed with each sampled realisation (Geyer, this volume). For the Hutterite example below, we sample every scan, since we are merely counting aspects of each realisation. Nonetheless, long runs will still be needed, to ensure the space is well sampled; high autocorrelation on a large pedigree means that many scans are needed to traverse the space. For the linkage example, where more computation is needed from each sample, we sample only every 20 scans. For this particular example, more frequent sampling might well be justified, but for likelihood analysis of complex genetic models is has been found that such an interval between samples may be necessary (Guo and Thompson, 1991a).

6 Two examples

In this section, two examples are given. The first is of the performance of the Gibbs sampler itself, through its use in providing posterior probabilities on a 583-member section of the Hutterite genealogy. The second example shows the use of Gibbs sampler realisations in providing Monte Carlo estimates of likelihood curves for a genetic linkage model. Each example is only a preliminary solution to the problem it addresses.

The Hutterite population is a North American religious and genetic isolate, now numbering over 25,000 but descended from only about 77 founders some 10 generations ago. Cystic fibrosis, a simple recessive genetic disease, has a high frequency in this population, and the ancestry of the haplotypes carrying the cystic fibrosis (CF) gene is of some interest (Fujiwara et al., 1989). The pedigree of 11 current cystic fibrosis cases is shown in figure 2 (see also Fujiwara et al. 1989) these 11 individuals trace to 62 founders. This pedigree of 583 individuals is far too complex to peel, but is well suited to Gibbs sampling. The data consist of the 11 individuals known to carry two copies of the CF gene; additionally, since cystic fibrosis is lethal, no ancestor can carry two copies of the CF gene. Here we consider only estimation of the marginal probability that each of the 62 founders carries the gene. This is achieved by running the Gibbs sampler, starting from an initial configuration in which all ancestors carry one CF gene, and ennumerating, after every scan, the founders who

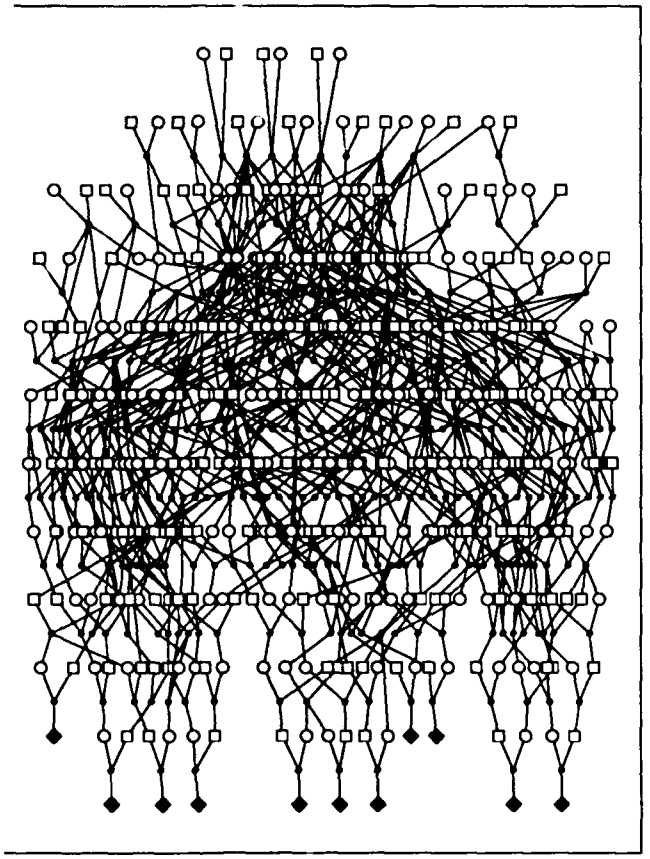


Figure 2: The Hutterite Cystic Fibrosis pedigree.

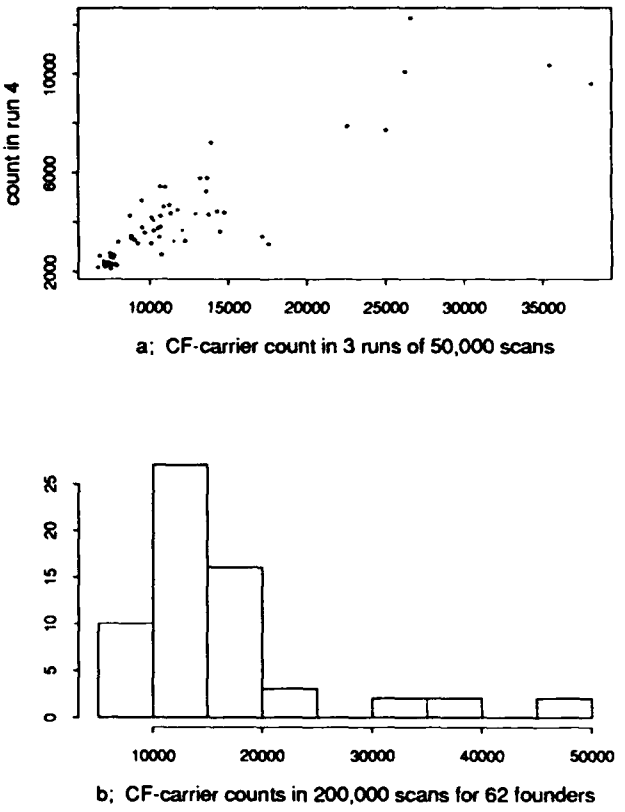


Figure 3: CF-carrier count Gibbs sampler results for 62 Hutterite founders.

carry the gene. The excess of carriers is very quickly eliminated from the configuration, particularly when a founder allele frequency of 0.025 (a typical European value) is assumed for the CF gene.

The results displayed here are preliminary, and illustrative of the method only. We have considered only the ancestors of CF cases, and not the information also provided by many unaffected lateral relatives. Second, these 11 cases are not the only identified CF cases in the Hutterite population. Third, we have not made use of information (Fujiwara et al., 1989) on closely linked DNA markers. Additionally, we have not made use of the symmetry between members of a founder couple, preferring to use this as a partial check on the results obtained, rather than as a constraint.

Four runs each of 50,000 sampled random scans were obtained, and the count of the number of times each of the 62 founders was exhibited as a CF carrier were tabulated. Figure 3a shows a plot of the counts for one of the runs against the total for the other three, showing broad agreement, but also considerable variation. In the 3-run totals couples were generally in good agreement. The 6 extreme points constitute 3 founder couples; only one of these shows substantial discrepancy between the estimates for the two members, even for the single run of only 50,000 scans. Figure 3b shows the histogram of counts for the 62 founders. The three couples with CF-carrier probability estimates greater than 0.15 (30,000/200,000) stand as outliers. Although these results are preliminary, this is a particularly encouraging result; there are good population genetic reasons (Fujiwara et al., 1989) for assuming that there must have been at least 3 original CF genes in this population.

Our second example shows the estimation of linkage likelihood curves from single runs of a Gibbs sampler, as described in section 4. The quantitative data $\mathbf{y}$ were simulated on an extended pedigree of 230 individuals, according to a complex genetic model (that is, $P_{\theta_0}(\mathbf{y}|\mathbf{G})$ is not as straightforward as described in section 2, and in fact is not easily evaluated). However, the feature primarily of interest is linkage with a marker locus, also simulated, and estimation of the LOD score $\log_{10}(L(r)/L(1/2))$ was achieved as in equation (4.8), estimating $L(r)/L(r_0)$ and $L(1/2)/L(r_0)$, where r_0 is the recombination frequency used in running the Gibbs

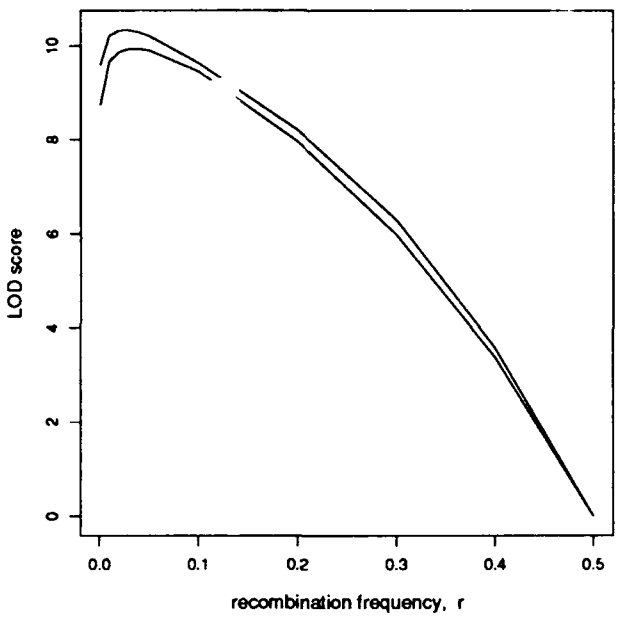


Figure 4: LOD score curves for genetic linkage example.

sampler. The simulation value of the recombination frequency, r, is 0.1. The maximum likelihood estimate, under the best fitting complex model within the class considered is 0.04. Two Gibbs sampler runs were therefore made; one at the maximum likelihood estimates, and the other with all other parameters at the maximum likelihood estimates, but with $r = 0.1$. The two curves are shown in figure 4.

Again, these are preliminary results, but demonstrate the feasibility of obtaining an entire likelihood curve from a single run of the Gibbs sampler. The similarity of shape of the two curves is encouraging. The small discrepancy in height is to be expected, since the curves are best estimated in the neighbourhood of r_0, and the normalisation relative to $r = 1/2$ will be more uncertain. In view of this, the small magnitude of the difference in the estimated maximum LOD score (the goal of so many applied analyses!), is very encouraging also.

7 Discussion

There is clearly considerable scope for Monte Carlo methods using the Gibbs sampler to address many questions of statistical genetics arising in the analysis of large and complex pedigrees. Although, where peeling is possible, there may be no good reason for resorting to Monte Carlo estimates, there are many cases where due to the complexity of the pedigree, of the model, or of both, an exact result is unobtainable. In such cases the Gibbs sampler can often be quite easily implemented and effectively employed.

There is considerable scope for the combination of exact computational algorithms with Monte Carlo approaches; specifically, to combine the Gibbs sampler and peeling. One such approach has been developed by Kong (this volume), and permits multilocus linkage analysis, which is another important area of modern genetic analysis in which exact computations have proved intractable or impossible. Peeling and Gibbs sampling can also be combined to provide likelihoods for other complex genetic models. Consider, for example, the mixed model of statistical genetics, in which there are both heritable random effects (say $\mathbf{z}$) and the effects of Mendelian genes ($\mathbf{G}$). While it is possible to generalise (4.7) and use a Gibbs sampler to obtain realisations from $P_{\theta_0}(\mathbf{z}, \mathbf{G}|\mathbf{y})$, this would not be an effective method of estimation (Thompson and Guo, 1991). In fact, for random effects models on pedigrees $P_\theta(\mathbf{y}|\mathbf{G})$ can be evaluated for any specified major-genotypic configuration $\mathbf{G}$ by a rather different form of the peeling algorithm. That is, for models with both heritable random effects and major genotypic effects, (4.7) can be used to estimate likelihood ratios (Thompson and Guo, 1991; Guo and Thompson, 1991a).

A third way of combining the Gibbs sampler and peeling is by dividing the pedigree rather than the model. On a large complex pedigree, it will often be the case that some portions can be peeled, providing R-functions on cutsets, each of several individuals who are members of a core pedigree too complex to be peeled. As a simple example, we might peel the right-hand segment of the pedigree of figure 1, providing an R-function on individuals 13 and 21, but wish to use the Gibbs sampler on the remainder

We wish to combine the R-functions into the Gibbs sampling to provide realisations on the core pedigree that are from the posterior distribution of core genotypes conditional on trait data on the entire pedigree. In fact the implementation is straightforward, if we recall the interpretations of the R-functions given by (3.4). The cutset, $\mathbf{C}$, consists of individuals whose parents are not in the peeled set, say $\mathbf{C}_u$ (e.g. 21), and others with at least one parent peeled, say $\mathbf{C}_d$ (e.g. 13). Let the peeled set be $\mathbf{Q}$ and the remainder of the pedigree $\mathbf{T}$. For $i \in \mathbf{C}$, let $\mathbf{K}_i$ denote the offspring of i in $\mathbf{T}$, and for j in $\mathbf{K}_i$ let S_j denote the other parent of j. Normally, S_j is

Γ: relationships through i's spouses in $\mathbf{Q}$ are already peeled. For individuals in $\mathbf{T}$ but not in $\mathbf{C}$, the Gibbs sampling is unaffected. For cutset individuals, the required Gibbs sampling of G_i should be conditional on data $\mathbf{y_Q}$ and y_i and genotypes $\mathbf{G_{T-i}}$, that is in the set

$\mathbf{T}$ less the individual i. For i in $\mathbf{C}_u$:

$$\begin{aligned}
&\text{prob}^*(y_{\mathbf{Q}}, \mathbf{G}_{\mathbf{C}_d}|G_{\mathbf{C}_u})P(y_i|G_i) \\
&\quad \times P(G_i|G_{M_i}, G_{F_i}) \textstyle\prod_{j\in\mathbf{K}_i, s_j\in\mathbf{T}} P(G_j|G_i, G_{S_j}) \\
&= P(y_{\mathbf{Q}}, \mathbf{G}_{\mathbf{C}_d}, G_i, y_i, G_{\mathbf{K}_i}|G_{\mathbf{C}_u - i}, G_{M_i}, G_{F_i}, \{G_{S_j}\}) \\
&\propto P(G_i|y_{\mathbf{Q}}, \mathbf{G}_{\mathbf{C}_d}, y_i, G_{\mathbf{K}_i}, G_{\mathbf{C}_u - i}, G_{M_i}, G_{F_i}, \{G_{S_j}\}) \\
&= P(G_i|y_{\mathbf{Q}}, y_i, G_{\mathbf{T}-i})
\end{aligned}$$

and for i in $\mathbf{C}_d$ we have similarly, without the segregation from i's parents,

$$\begin{aligned}
&\text{prob}^*(y_{\mathbf{Q}}, \mathbf{G}_{\mathbf{C}_d}|G_{\mathbf{C}_u})P(y_i|G_i) \\
&\quad \times \textstyle\prod_{j\in\mathbf{K}_i, s_j\in\mathbf{T}} P(G_j|G_i, G_{S_j}) \\
&= P(y_{\mathbf{Q}}, \mathbf{G}_{\mathbf{C}_d - i}, G_i, y_i, G_{\mathbf{K}_i}|G_{\mathbf{C}_u}, \{G_{S_j}\}) \\
&\propto P(G_i|y_{\mathbf{Q}}, \mathbf{G}_{\mathbf{C}_d - i}, y_i, G_{\mathbf{K}_i}, G_{\mathbf{C}_u}, \{G_{S_j}\}) \\
&= P(G_i|y_{\mathbf{Q}}, y_i, G_{\mathbf{T}-i})
\end{aligned}$$

Thus Gibbs sampling for members of the cutset involves only extracting the currently appropriate term from any R-function on any cutset of which the individual is a member. This is not a final solution; questions of irreducibility of the Markov chain on the core pedigree arise, and would have to be resolved in any specific example, just as they must in any case be resolved for any genetic model. However, given such irreducibility, the Gibbs sampler provides realisations for risk assessment or likelihood evaluation, just as before.

Acknowledgement

I am grateful to former students Charles J. Geyer and Nuala A. Sheehan and to current student Sun Wei Guo. Discussions with them have been a major contribution to the development of this work. The pedigree diagrams used software by Alun Thomas and by Charles Geyer. The computer programs used to obtain the Hutterite results in section 6 were based on previous programs by Nuala Sheehan. The programming and computing for the linkage example were done by Sun Wei Guo.

I am grateful to Ken Morgan for access to the Hutterite pedigree data base, and for many discussions about the medical genetics of the Hutterite population. Thanks also to Ken Morgan and Mary Fujiwara for permission to modify, and use for figure 2, the pedigree graphics file for figure 1 of Fujiwara et al. (1989).

References

Cannings, C., Thompson, E. A., and Skolnick, M. H. (1978) Probability functions on complex pedigrees. *Adv. Appl. Prob.*, **10**, 26–61.

Elston, R. C. and Stewart, J. (1971). A general model for the genetic analysis of pedigree data. *Human Heredity*, **21**, 523–542.

Fujiwara, T. M., Morgan, K., Schwartz, R. H., Doherty, R. A., Miller, S. R., Klinger, K., Stanislotovitis, P., Stuart N. and Watkins, P. C. (1989) Genealogical analysis of cystic fibrosis families and chromosome 7q RFLP haplotypes in the Hutterite Brethren. *Am. J. Hum. Genet.*, **44**, 327–337.

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, **6**, 721–741.

Guo, S-W. and Thompson, E. A. (1991a). Monte-Carlo estimation of mixed models on large and complex pedigrees. *In preparation.*

Guo, S-W and Thompson, E. A. (1991b). Monte-Carlo methods for the linkage analysis of complex genetic traits. *In preparation.*

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Heuch I. and Li F. M. F. (1972). PEDIG—A computer program for calculation of genotype probabilities using phenotypic information. *Clin. Genet.*, **3**, 501–504.

Hilden, J. (1970). GENEX—An algebraic approach to pedigree probability calculus. *Clin. Genet.*, **1**, 319–348.

Lauritzen, S. L. and Spiegelhalter D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems *(with Discussion). J.R.S.S. (B)* **50**, 157–224.

Ott, J. (1979). Maximum likelihood likelihood estimation by counting methods under polygenic and mixed models in human pedigrees. *Amer. J. Hum. Genet.*, **31** 161–175.

Thompson, E. A. (1986) *Pedigree Analysis in Human Genetics.* The Johns Hopkins University Press, Baltimore, MD.

Thompson E. A and Guo, S.-W. (1991). Monte Carlo evaluation of likelihood ratios. *Submitted*

Analysis of Pedigree Data Using Methods Combining Peeling and Gibbs Sampling

Augustine Kong*

Department of Statistics
University of Chicago
Chicago, IL 60637

Abstract

Peeling and Gibbs sampling are two computational tools for genetic pedigree analysis. While both are powerful methods, each has its limitations. There are problems where the application of either one technique alone will not lead to satisfactory results. For some of these problems, we propose methods which combine peeling and Gibbs sampling. The key idea is to take full advantage of the strengths of each method and eliminate the weaknesses.

KEY WORDS: Pedigree analysis, Peeling, Gibbs sampling, Monte Carlo, Markov chain, Likelihoods, Lod score, Bayesian inference.

1 Introduction

Peeling is a standard computational tool geneticists used for pedigree analysis (Elston and Stewart 1971, Lange and Elston 1975, Cannings et al 1978). While it is a powerful method. there are also limitations. For example, for problems which involve multiple loci or complicated genetic models with many parameters, simple application of the peeling algorithm can be infeasible or impractical due to the limitations of memory and speed of computations. Another technique, the Gibbs sampler, which had been used extensively in statistical physics and image reconstruction (see Geman and Geman 1984 and Gelfand and Smith 1990), has recently found its way into the genetics literature (Thompson and Wijsman 1990). The Gibbs sampler is an iterative technique which allows us to draw multiple, but dependent, realizations of the unobserved data (and sometimes parameters in a Bayesian setting) conditioned on the observed data. When applied to pedigree analysis, the drawn samples can be used to get estimates of likelihood ratios and, in a Bayesian setting, posterior distributions and posterior odds. A potential weakness of the Gibbs sampler is that the samples it generated can be too highly correlated so that the resulting Monte Carlo estimates can be very far from the actual values without the user noticing it. For a large class of problems which cannot be handled very well by either peeling or Gibbs sampling alone, we propose combining the two two techniques to achieve a satisfactory result.

In section 2 we will give a brief review of peeling and highlight its limitations. Section 3 gives a brief description of Gibbs sampling and also discusses the problem of convergence. Section 4 and 5 contain two examples which illustrate how peeling and Gibbs sampling can be combined. Section 6 has some final remarks.

2 Pedigree Analysis and Peeling

Pedigree analysis belongs to a class of problems in statistics common known as *missing data* problems. Consider a pedigree with n individuals. For $i = 1, \ldots n$, let g_i denote the genotype of person i and y_i the observed phenotype. Depending on the problem, both g_i and y_i may involve a single locus or multiple loci on the chromosomes. The joint distribution of the g_i's and y_i can usually be written as

$$p_{\theta,\eta}(\mathbf{g},\mathbf{y}) = p_\theta(\mathbf{g}) \prod_{i=1}^{n} p_\eta(y_i|g_i) \qquad (1)$$

where θ is the recombination fraction(s) between loci and η is the parameter vector associated with the genetic model relating y_i and g_i. (Thompson 1986 is a good reference for standard terminology.) When both $\mathbf{g}$ and

*Research supported in part by NSF grant DMS-89-02667. Computations for this document were performed using computer facilities supported in part by the National Science Foundation under grants DMS 89-05292, DMS 87-03942, DMS 86-01732, and DMS 84-04941 awarded to the Department of Statistics at The University of Chicago, and by The University of Chicago Block Fund.

$\mathbf{y}$ are given, (1), interpreted as a function of θ and η, is called the *complete data* likelihood. However, usually only $\mathbf{y}$ is observed and $\mathbf{g}$ is referred to as the *missing data*. The likelihood function based on the observed data only can be written as

$$l(\theta, \eta; \mathbf{y}) = p_{\theta,\eta}(\mathbf{y}) = \sum_{\mathbf{g}} [p_\theta(\mathbf{g}) \prod_{i=1}^{n} p_\eta(y_i|g_i)]. \quad (2)$$

The sum is over all genotype vectors that are compatible with the observed data. For any fixed values of θ and η, each term in the sum is trivial to compute. However, since in general there will be many genotype vectors $\mathbf{g}$ which are compatible with the observed data, summing by brute force is computationally infeasible except for very small pedigrees. Let $V = \{1, \ldots, n\}$ and $V' = \{i|\text{person } i \text{ has no parents in the pedigree}\}$, the later usually referred to as the set of *founders*. The joint distribution of the g_i's has the factorization

$$p_\theta(\mathbf{g}) = \prod_{i \in V'} p(g_i) \prod_{i \in V - V'} p_\theta(g_i|g_{f_i}, g_{m_i}) \quad (3)$$

where f_i and m_i denote the father and mother of person i respectively. This factorization reflects the fact that the genetic material of a person is inherited from his/her parents. By taking advantage of (3), for pedigrees without loops, peeling breaks down the *global* sum (2), which involves the joint outcome space of all the g_i's, into a sequence of *local* sums, each one involving at most the genotypes of three persons (two parents and an offspring). The basic idea is to sum out (peel) one person at a time. Peeling is related to the Kalman filter and has applications other than genetics (Lauritzen and Spiegelhalter 1988).

The method of peeling works very well for many problems, but can face difficulties in the following situations:

A. PEDIGREES WITH MANY LOOPS. Pedigrees with many loops due to inbreeding can be found in studies of rare recessive diseases. Loops create problems because it complicates the dependencies among relatives. Peeling can still be used, but instead of local computations involving three people at a time, sometimes computations have to be done on four or more people simultaneously. When there are too many inbreeding loops, the memory and computations requirements can reach a stage where peeling is either infeasible or at the least, impractical.

B. MULTIPLE LOCI. There are problems in genetics, such as multi-point linkage analysis, where many linked loci have to be handled simultaneously. Suppose we are dealing with loci 1 to k which have respectively $m_1, m_2, \ldots, m_k$ number of alleles. The number of states the composite (involving all the loci) genotype of an individual can have is approximately

$$\frac{1}{2} \prod_{j=1}^{k} m_j^2. \quad (4)$$

It can be easily seen that even if there is no inbreeding so that we only have to handle three people at a time, the amount of computations required, which is proportional to the cube of (4), will quickly exceed our capability as the number of loci increases. We propose in Section 4 a method to handle this class of problems.

C. MODELS WITH MANY PARAMETERS. Sometimes the genetics models can be very complicated and involve many parameters. Even if it is possible to peel the pedigree for any given set of parameter values, which sometimes is not the case, this only gives you one point of the likelihood function defined on a high dimensional space. It may require many of these point by point evaluations for us to get the maximum likelihood estimate and other inference tools such as lod scores (basically generalized likelihood ratios). This can become impractical, if not infeasible, for problems with many parameters. For problems of this type, peeling has to be coupled with other techniques such as the EM algorithm (Lander and Green 1987) or the Gibbs sampler. An example of the later is presented in Section 5.

3 Monte Carlo Approximations of Likehood Ratios and Gibbs Sampling

Instead of evaluating likelihoods exactly using methods such as peeling, for many problems, it is often adequate if multiple realizations of the unobserved genotypes can be drawn jointly conditioned on the observed data. For example, as suggested by Thompson and Wijsman (1990), suppose we can simulate realizations of $\mathbf{g}$ from the the conditional distribution

$$p_{\theta_0,\eta_0}(\mathbf{g}|\mathbf{y}) \quad (5)$$

where θ_0 and η_0 are some chosen values of the parameters. Let $\mathbf{g}(t), t = 1, \ldots, T$, be T simulated values of $\mathbf{g}$. For any other parameter vector (θ, η), the likelihood ratio $l(\theta, \eta)/l(\theta_0, \eta_0) = p_{\theta,\eta}(\mathbf{y})/p_{\theta_0,\eta_0}(\mathbf{y})$ can be approximated by

$$\frac{1}{T} \sum_{t=1}^{T} \frac{p_{\theta,\eta}(\mathbf{g}(t), \mathbf{y})}{p_{\theta_0,\eta_0}(\mathbf{g}(t), \mathbf{y})}. \quad (6)$$

This approximation is justified because of the identity

$$\frac{p_{\theta,\eta}(\mathbf{y})}{p_{\theta_0,\eta_0}(\mathbf{y})} = \sum_{\mathbf{g}} \frac{p_{\theta,\eta}(\mathbf{g},\mathbf{y})}{p_{\theta_0,\eta_0}(\mathbf{g},\mathbf{y})} p_{\theta_0,\eta_0}(\mathbf{g}|\mathbf{y}). \qquad (7)$$

Expression (6) is essentially a Monte Carlo estimate based on importance sampling. Note that (6) is easy to compute because each of the complete data likelihoods are suppose to be simple. For other situations where we will like to simulate the missing genotypes conditioned on the observed data, see Kong (1991) and Lange and Sobel (1990). If the pedigree can be peeled, the peeling algorithm can be modified for the simulations described here (Ploughman and Boehnke 1989, Ott 1989 and Kong 1991). Now suppose the pedigree cannot be peeled because of reasons given in Section 2. Here is where Gibbs sampling comes into play.

Gibbs sampling is a very general technique for simulating joint realizations of dependent variables. The idea is very simple. Suppose there is set of variables which we want to simulate jointly, possibly conditioned on some observed data. This set of variables is partitioned into a number of components so that each component consists of one or more variables. We start with some configuration of all the variables which is compatible with the observed data. Individual components are then visited based on some systematic or random scheme. When a component is visited, a realization of it is drawn conditioned on the *current* configuration of all the other components. The iterations set up a stationary Markov chain whose equilibrium distribution is the same as the joint distribution we want to simulate from. This implies that, after many iterations, which means that each component has been visited many times, the joint realization of all the components can be considered as a draw from the desired distribution (conditioned on the observed data and given parameter values). A key point to note is that the Gibbs sampler itself does not specify exactly how the variables should be partitioned. The later is a decision that the user has to make. There are two criteria for choosing an optimal partition:

(I) Drawing one component conditioned on the others is computationally simple.

(II) The Markov chain induced by the Gibbs sampler applied to this partition has to converge reasonably fast to its equilibrium distribution.

It is not difficult to see that (I) and (II) are usually conflicting criteria. For example, not partitioning the variables at all and just drawing them jointly is best under (II), but it is in general impossible to implement and is the reason that the Gibbs sampler was invented in the first place. In general, some compromise has to be made.

A natural way of applying Gibbs sampling to pedigrees is what I will refer to as *person by person* Gibbs sampling. In this case, each component of the partition is the composite genotype of an individual. This partition satisfies criterion (I) mainly because each person is related to all the other people in the pedigree only through his/her parents, spouse and children. This approach however can run into serious trouble as far as criterion (II) is concerned. The reason is that, in order for the Gibbs sampler to work, the induced Markov chain has to be irreducible, i.e. each point in the state space can be reached from another point through the Markov chain. A sufficient condition for irreducibility is the *positivity* condition introduced by Besag(1974). Basically, *positivity* means that although the variables are dependent in a probabilistic fashion, any joint configuration of the variables are logically possible. The later is clearly violated in our setting because given the genotypes of the parents, some genotypes for the offspring are logically eliminated. Because of this, for general genetic problems, a naive application of *person by person* Gibbs sampling can fail completely. While there are tricks (see for example Sheehan and Thomas 1991) which can turn a non-irreducible chain into an irreducible one, the efficiencies of such approaches are still in question. More investigation in this direction is warranted.

4 Locus by Locus Gibbs Sampling

In this section, we consider a multiple loci problem where peeling is infeasible. The task here is to simulate the unobserved genotypes, jointly for all the loci and all the individuals in the pedigree, conditioned on the observed data and possibly some fixed values of the parameters. Instead of doing *person by person* Gibbs sampling, we propose an alternative method which will be called *locus by locus* Gibbs sampling.

For each locus and each non-founder, we define two identity by decent (IBD) variables, one on the mother side and one on the father side. Each IBD variable is binary and indicates whether the allele at a particular locus is inherited from the grandfather or grandmother. In a pedigree, the main reason that the genotypes at different loci are dependent is because the IBD's are correlated. In particular, for the same individual , two IBD variables corresponding to a single parent and two linked loci are positively correlated. As pointed out by Lander and Green (1987) and Kong (1991), under the assumption of no interference, these IBD variables form a Markov chain. For example, suppose we have five or-

dered loci A,B,C,D and E, then given the states of the IBD variables associated with C, the IBD variables associated with loci A and B are independent of those IBD variables associated with loci D and E. However, even if interference is allowed so that the IBD variables do not form a Markov chain, the conditional distribution of the IBD variables of a particular locus given the IBD variables of the other loci can still be easily computed. Noting this key fact, instead of simulating all the composite genotypes jointly, we can construct a Gibbs sampling scheme which simulates the genotypes and IBD variables one locus at a time. When a locus is "visited", we draw a sample of its genotypes and IBD variables, jointly for all individuals, conditioned on the observed data of that locus and the current imputed values of the IBD variables of the other loci. Computationally, each simulation step requires peeling. However, since only a single locus is handled each time, peeling is usually possible. As long as the loci are not right on top of each other, it is trivial to show that *locus by locus* Gibbs sampling always lead to an irreducible Markov chain. This method is similar in spirit to one proposed by Lange and Sobel (1990), but their approach has the limitation that each locus must have only two alleles. The strength of our approach is that it does not require no interference, a condition that is crucial for the method proposed by Lander and Green (1987), and has no restrictions on the number of alleles a marker may have. The implementation of *locus by locus* Gibbs sampling is currently underway.

5 A Model with Many Parameters

As discussed earlier, for genetic models which have many parameters, point by point evaluation of likelihoods can be highly inefficient. Here we consider one model of this type. Suppose we are doing a linkage analysis with the gene locus of a quantitative trait and a single marker locus. Apart from the genetic effect, suppose there is an observed categorical covariate with three levels (0,1,2) which may also has an effect on the observable trait. Following Bonney et al (1988), we consider a regression model for the quantitative trait:

$$z_i = \alpha + \beta G_i + \gamma_1 X_{i1} + \gamma_2 X_{i2} + +\epsilon_i. \qquad (8)$$

Here z is the quantitative trait, i is the indicator for individuals, α is the mean for individuals who does not carry the disease allele and whose covariate belongs to level 0, $\gamma_j, j = 1,2$, is the difference between level j and level 0, $X_{ij}, j = 1,2$, is the indicator variable for level j of the covariate, β is the genetic effect, G is the indicator for whether an individual carries the disease allele and the ϵ_i's are assumed to be noise which are iid $N(0,\sigma^2)$. Following the notations established in Section 2, θ denotes the recombination fraction between the gene locus and the marker locus, and η denotes the vector $(\alpha,\beta,\gamma_1,\gamma_2,\sigma)$. Here g_i denotes the composite genotype which includes both the gene locus and marker locus. The observed data y_i will include z_i, the X's and the single locus genotype of the marker. (For some individuals in the pedigree, even y_i may be missing.)

Instead of simply computing likelihoods, we set up a Bayesian model where the parameters are also treated as random variables. Standard conjugate priors (Box and Tiao 1973) are assigned to the parameters. This implies that if both the missing data $\mathbf{g}$ and the observed data $\mathbf{y}$ are given, the complete data posterior distribution $p(\theta,\eta|\mathbf{g},\mathbf{y})$ can be obtained in closed form and is easy to draw from. With this setup, we do Gibbs sampling by iterating between the parameters (θ,η) and the unobserved genotype vector $\mathbf{g}$. Starting with some initial configuration $\mathbf{g}(0)$ which is compatible with $\mathbf{y}$, we draw a realization $(\theta(1),\eta(1))$ of the parameters from the conditional distribution $p(\theta,\eta|\mathbf{g}(0),\mathbf{y})$. We then draw a realization $\mathbf{g}(1)$ conditioned on $(\theta(1),\eta(1))$. In general, at time t, we draw a sample $(\theta(t),\eta(t))$ from the conditional distribution

$$p(\theta,\eta|\mathbf{g}(t-1),\mathbf{y}) \qquad (9)$$

and then draw a sample $\mathbf{g}(t)$ from

$$p(\mathbf{g}|\mathbf{y},\theta(t),\eta(t)) \qquad (10)$$

Drawing from (9) is simple because of the conjugate setup. Drawing from (10) requires a modification of peeling which was mentioned in Section 3. Based on the theory described in Section 2, for t large, $(\theta(t),\eta(t))$ and $\mathbf{g}(t)$ can be considered as draws from the desired conditional distributions

$$p(\theta,\eta|\mathbf{y}) \qquad (11)$$

and

$$p(\mathbf{g}|\mathbf{y}). \qquad (12)$$

where (11) is the posterior distribution of the parameters and (12) is the predictive distribution of the missing data $\mathbf{g}$. Because of that, histograms of the drawn parameter values can be used as approximations of the posterior distribution (11).

Using a particular pedigree structure (see Kong et al 1991 for details), we simulated one set of data based on $\theta = 0.005$ and $(\alpha,\beta,\gamma_1,\gamma_2,\sigma) = (10,1,1,2,1)$. We then apply the Gibbs sampler to the simulated data. In the

analysis, flat prior distributions are used for the parameters $(\alpha, \beta, \gamma_1, \gamma_2, \sigma)$. For the prior distribution of the recombination fraction θ, we use a probabilistic mixture of a delta function at $\theta = 1/2$ and a uniform distribution between 0 and 1/2. The delta function at 1/2 has weight .957 to reflect the 1 : 22 prior odds against the marker and disease gene being located on the same chromosome. A total of 5000 iterations were run. The histograms displayed in Figure 1 are constructed based on the last 4000 samples of the drawn parameter values. The posterior probability supporting linkage ($\theta < 1/2$) is approximately .91. This is very substantial considering the prior distribution we used for θ. We note here that, apart from posterior distributions, the Gibbs sampler described here also give excellent estimates of more traditional inference tools such as lod scores. For details, see Kong et al (1991).

Instead of using the histograms to approximate the posterior distributions of the parameters, there is a better alternative. Note that

$$p(\theta, \eta|\mathbf{y}) = \sum_{\mathbf{g}} p(\theta, \eta|\mathbf{g}, \mathbf{y}) p(\mathbf{g}|\mathbf{y}) \qquad (13)$$

This suggests, in our example, approximating $p(\theta, \eta|\mathbf{y})$ by

$$\frac{1}{4000} \sum_{t=1001}^{5000} p(\theta, \eta|\mathbf{g}(t), \mathbf{y}) \qquad (14)$$

assuming that each of the 4000 complete data posteriors can be obtained in closed form. Expression (14) is called the mixture approximation of the posterior distribution. Liu et al (1991) has proved that the mixture approximation is always superior to the histogram approximation in the sense that it has smaller variance. The smooth curve in Figure 1a is the mixture approximation. With a little of work, similar approximations can be obtained for the other parameters.

6 Remarks

Using two examples, we illustrated how the methods of peeling and Gibbs sampling can be combined. Potentially there are many problems where the same idea can be applied. It can be problems which have the characteristics of both of our examples. In some cases, it may make sense to combine *locus by locus* Gibbs sampling with *person by person* Gibbs sampling. For example, consider a linkage analysis where the genetic model relating the phenotype to the genotype is too complicated so that the gene locus cannot be peeled. Here we probably will still want to simulate the marker genotypes and IBD's jointly using peeling. However, for the gene locus, we may be forced to apply *person by person* Gibbs sampling.

Computational issues aside, there is also the question of statistical inference. Bayesian inference provides an alternative to traditional inference which is based mainly on profile likelihoods and lod scores. The relative merits of these different approaches warrant further research.

References

Besag, J. (1974) Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. R. Statist. Soc. B* **36**, 192–236.

Bonney GE, Lathrop GM and Lalouel Jean-Marc. (1988) Combined linkage and segregation analysis using regressive models. *Am. J. Hum. Genet.* 43:029–037.

Box GEP and Tiao GC. (1973) *Bayesian Inference in Statistical Analysis.* Addison-Wesley:Reading.

Cannings, C., Thompson, E. A. and Skolnick, M. H. (1978), "Probability functions on complex pedigrees," *Advances in Applied Probability 6*: 26-61.

Elston, R. C. and Stewart, J. (1971), "A General Model for the Genetic Analysis of Pedigree Data," *Hum. Hered. 21*: 523-542.

Gelfand A. E. and Smith, A.F.M. (1990), "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association 85*: 398-409.

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, **6**, 721–741.

Kong, A. (1991), "Efficient Methods for Computing Linkage Likelihoods of Recessive Diseases in Inbred Pedigrees," to appear in *Genetic Epidemiology*, Vol 8, Number 2.

Kong, A., Frigge M., Cox N. and Wong W. H. (1991), "Linkage Analysis with Adjustment for Covariates: A Method Combining Peeling with Gibbs Sampling," presented at the seventh Genetic Analysis Workshop. To appear in *Cytogenetics and Cell Genetics*.

Lander, E. S. and Green, P. (1987), "Construction of Multilocus Genetic Linkage Maps in Humans," *Proc. Natl. Acad. Sci. USA 84* :2363-2367.

Lange, K. and Elston, R. C. (1975), "Extensions to Pedigree Analysis. I. Likelihood Calculations for Simple and Complex Pedigrees ," *Hum. Hered. 25*: 95-105.

Lange, K. and Matthysse S. (1989), "Simulation of Pedigree Genotypes by Random Walks," *Am. J. Hum. Genet 45*: 959-970.

Lange, K. and Sobel E. (1990), "A Random Walk Method for Computing Genetic Location Scores," manuscript.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988), "Local Computation with Probabilities on Graphical Structures and Their Application to Expert Systems," with discussion *J. R. Statist. Soc. B 50*: 157-224.

Liu, J., Wong W. and Kong A. (1991), "Correlation Structure and Convergence Rate for Gibbs Sampling," submitted to *Biometrika.*

Ott, J. (1989), "Computer-Simulation Methods in Linkage Analysis," *Proc. Natl. Acad. Sci. USA 86* :4175-4178.

Ploughman,L.M. and Boehnke, M. (1989), "Estimating the Power of a Proposed Linkage Study for a Complex Genetic Trait," *Am. J. Hum. Genet. 44* : 543-551.

Sheehan, N. and Thomas A. (1991), "On the Irreducibility of a Markov Chain Defined on a Space of Genotype Configurations by a Sampling Scheme," to appear in *Biometrics*

Thompson, E. A. (1986), *Pedigree Analysis in Human Genetics* , Johns Hopkins University Press.

Thompson, E. and Wijsman E. (1990), "The Gibbs Sampler on Extended Pedigrees: Monte Carlo Methods for the Genetic Analysis of Complex Traits," Technical Report 193, Department of Statistics,University of Washington.

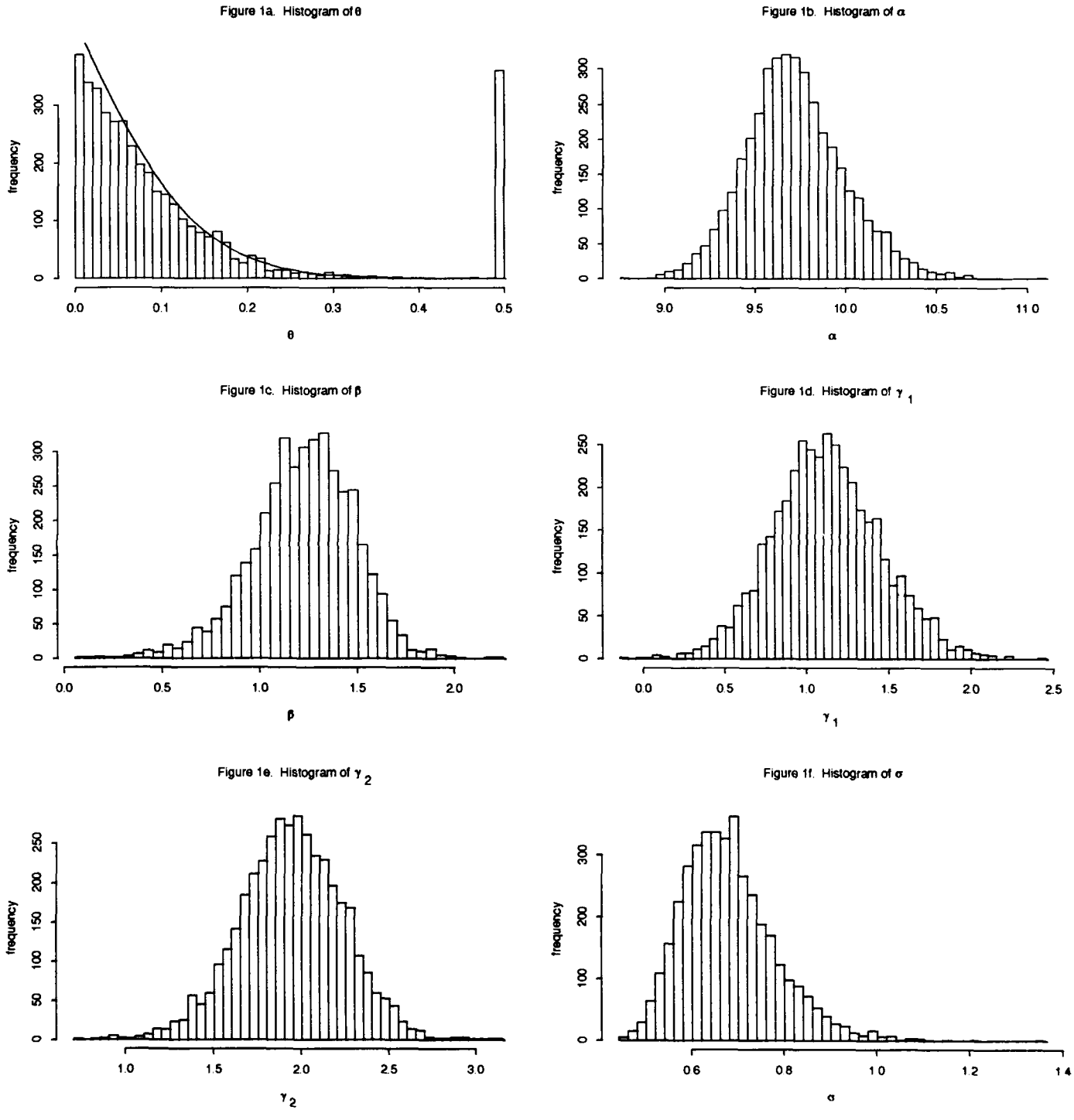


Figure 1. Histograms of Simulated Parameter Values

An overview of the affected-pedigree-member method of linkage analysis

Daniel E. Weeks[1] and Kenneth Lange[2,§]

[1]Department of Human Genetics, University of Pittsburgh, Crabtree Hall A310, 130 DeSoto Street, Pittsburgh, PA 15261

[2]Department of Biomathematics, UCLA School of Medicine, Los Angeles, CA 90024-1766. Present address: Harvard University, Department of Statistics, Science Center, One Oxford Street, Cambridge, MA 02138.

Summary

Human geneticists have been extremely successful in the past decade in mapping rare disease genes. For common diseases with a substantial genetic component, the payoff for human health is larger, but the mapping problems are harder. There is a need for robust statistical techniques that require minimal assumptions about the mode of inheritance of the disease studied. The affected-pedigree-member (APM) of linkage analysis makes virtually no assumptions about disease transmission except that it is independent of marker transmission in a pedigree. We discuss here an extension of the APM method from single markers to multiple closely linked markers. This extension should improve the power of the APM method to detect linkage.

Introduction

Chromosomes are not passed intact from generation to generation. During the formation of gametes (eggs and sperm), homologous chromosomes align and recombine. This produces gamete chromosomes that alternate between maternal and paternal sources. The further apart two loci are, the more likely it is that recombination will occur between them. Recombination has the effect of separating an allele at one locus from an allele at the second locus. The frequency of this reshuffling, whether directly observable or not in the offspring of a parent, is termed the recombination fraction between the loci. Conventional linkage analysis aims at estimating the recombination fraction. In order to measure recombination between a disease locus and a neutral marker locus, there must be an explicit model for the phenotypic expression of the disease locus. Such a model permits the inference of underlying disease genotypes from observed disease phenotypes. For rare Mendelian diseases, the model is usually clear, and linkage analysis has proved to be an extremely powerful tool for mapping these diseases (e.g., Kerem et al. 1989; Riordan et al. 1989; Rommens et al. 1989).

For more complex, common diseases such as schizophrenia (Weeks et al. 1990), no one knows the correct genetic model. This quandary is hardly resolved by selecting a simple model inconsistent with the known pedigree data. In fact, if the genetic model is misspecified, then this may mask the evidence for linkage (Baron 1990; Clerget-Darpoux et al. 1990; Martinez et al. 1989; Weeks et al. 1990a). For example, Figure 1 displays a pedigree in which the unaffected daughter is almost certainly a recombinant under an incorrect model but more likely a nonrecombinant under the correct model. Incorrect inferences about recombination events are disastrous for lod scores.

In the current paper we present an alternative to computing lod scores under questionable models. The APM method, which uses all the affected individuals in a pedigree, was preceded and inspired by the earlier affected-sib-pair methods, which used only affected siblings (Day and Simons 1976; de Vries et al. 1976; Fishman et al. 1978; Green and Woodrow 1977; Haseman and Elston 1972; Lange 1986a; Lange 1986b; Lange and Weeks 1990; Penrose 1935; Suarez and Hodge 1979; Suarez et al. 1978; Thomson and Bodmer 1977; Weeks and Lange 1988; Weeks and Lange 1991). These methods are motivated by the fact that affected individuals will tend to be alike at markers closely linked to

§ This work supported in part by: the University of California, Los Angeles; Harvard University; the University of Pittsburgh; and USPHS grant CA 16042.

a disease locus. Marker similarity between unaffected individuals is less predictable and is therefore ignored. The APM method seeks to answer in a model-free way the simple question: Are the affected individuals more similar than expected by chance at the marker locus (or at a group of markers closely linked to each other)? To answer this question, we need a measure of marker similarity among the affecteds of a pedigree and some idea of the statistical distribution of the measure.

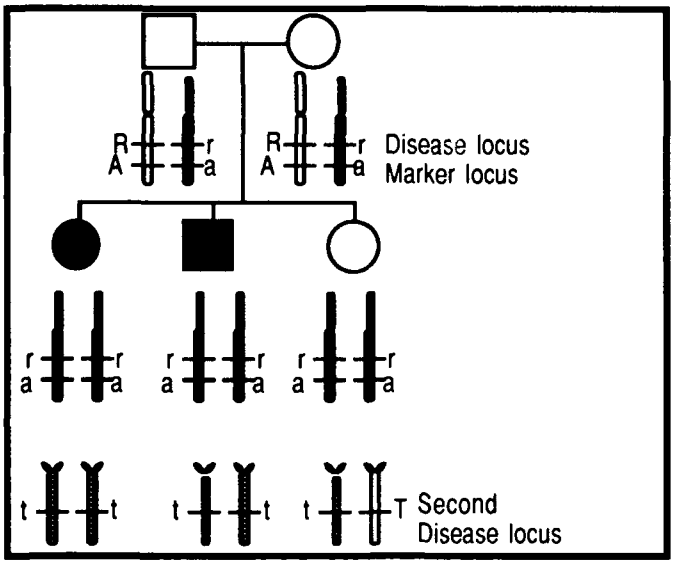


Figure 1: If a single-locus recessive model is postulated, then at least one recombination event must be invoked to explain the sib phenotypes. If a two-locus doubly recessive model is postulated, then the sib phenotypes are consistent with no recombination. Note that parental phases are, in fact, unknown.

A similarity measure

We will first discuss how to define marker similarity at a single marker locus. One natural measure of marker similarity is simply to count the number of matches of marker alleles between two affected individuals i and j, letting each match contribute a fourth to the statistic Z_{ij} (Table 1). In other words, we look at the identity-by-state status of the marker alleles. Two alleles are *identical-by-state* if they are the same allele, regardless of ancestral origin. Note that our measure ignores the common sense notion that, given two distantly related affecteds, a match between two common alleles is less striking than a match between two rare alleles. Thus, we modify our measure of marker similarity so that each match in state contributes a fourth times f(p), where f is a weighting function of the marker allele frequency p. For example, if f(p)=1/p and the allele frequencies of allele A and allele B are 0.10 and 0.90, respectively, then two individuals i and j who are both A/B at the marker locus would have a similarity measure Z_{ij} = (1/4)(10) + (1/4)(1.11) = 2.78. Since the magnitude of this similarity measure is meaningless by itself, we now turn our attention to the calculation of the mean and variance of Z_{ij} under the null hypothesis of independent segregation of disease and markers.

Table 1: Possible values of the unweighted similarity statistic Z_{ij} for a pair of affected individuals.

Marker Genotypes		
i	j	Z_{ij}
A/A	A/A	1
A/B	A/A	1/2
A/B	A/B	1/2
A/B	A/C	1/4
A/B	C/D	0

Mean of the similarity measure

First, let us rewrite the similarity measure Z_{ij} as the conditional expectation

$$Z_{ij} = E\left[U_{ij} \mid \text{marker genotypes of i and j} \right] \quad . \quad (1)$$

To define the random variable U_{ij} appearing on the right of (1), imagine sampling random gametes from i and j. If the two sampled gametes bear the same allele a_r at the current marker locus, then set $U_{ij} = f(p_r)$, where p_r is the population frequency of a_r. If the two alleles do not match in state, set $U_{ij} = 0$. As a consequence of (1), $E(Z_{ij}) = E(U_{ij})$.

To better understand the random variable U_{ij}, it is helpful to introduce the notion of identity-by-descent. Two genes are *identical-by-descent* if one is a copy of the other or they are both copies of the same ancestral gene. Identity-by-descent obviously implies identity-by-state, but not conversely. It is quite easy to calculate the probability that a gene drawn at random from one person is identical-by-descent to a gene at the same locus drawn at random from another person. This probability Φ_{ij}, known as the *kinship*

oefficient, depends only on the relationship between the wo individuals i and j, *i.e.*, on the graphical structure of the ɔedigree connecting them.

The mean of U_{ij} may be calculated by conditioning ɔn the identity-by-descent status of the alleles being :ompared. If the alleles are identical-by-descent (with ɔrobability Φ_{ij}), then U_{ij} takes on the value $f(p_r)$ with ɔrobability p_r. If the alleles are not identical-by-descent (with probability $1-\Phi_{ij}$), then U_{ij} takes on the value $f(p_r)$ with probability p_r^2 since the two alleles enter the pedigree of i and j independently. These considerations lead to

$$E(Z_{ij}) = \Phi_{ij}\sum_r p_r f(p_r) + (1-\Phi_{ij})\sum_r p_r^2 f(p_r).$$

We define the similarity measure Z for a pedigree to be the sum of the similarity measures Z_{ij} between all possible affected pairs. In other words,

$$Z = \sum_{i<j} Z_{ij}.$$

The expectation of Z is simply the sum of the expectations of the Z_{ij}'s. To calculate the variance of Z, it is necessary to compute terms such as $E(Z_{ij}Z_{kl})$, involving up to four distinct individuals. Fortunately, $E(Z_{ij}Z_{kl})$ may be calculated by a conditioning approach very similar to that used to calculate the mean of Z_{ij}. Instead of conditioning on whether two genes are identical-by-descent, we now condition on the identity-by-descent relationships among the genes drawn from the four individuals i, j, k, and l. The probability of each of the 15 possible identity-by-descent partitions of these four genes can be easily calculated using the *generalized kinship coefficients* of Karigl (1981; 1982), as extended by Weeks and Lange (1988). In short, it is possible to calculate exactly the theoretical mean and variance of the similarity measure Z for any pedigree. We then standardize Z by subtracting off its mean, dividing by its standard deviation, and weighting by $\sqrt{r-1}$, where r is the number of affected individuals in the pedigree:

$$W = \sqrt{r-1}\,\frac{Z - E(Z)}{\sqrt{Var(Z)}}$$

For several pedigrees, we form a grand APM statistic T with mean zero and variance 1 by dividing the sum of the standardized measures W_t from each pedigree t by the appropriate sum of weights:

$$T = \frac{\sum_t W_t}{\sqrt{\sum_t (r_t - 1)}}.$$

The statistic T is asymptotically standard normal, provided the number of pedigrees is large. A one-sided test based on the observed T is appropriate since linkage acts to increase marker sharing among affecteds.

Extension to multiple marker loci

As marker maps of the human genome become denser, investigators are more likely to type disease pedigrees at several closely linked marker loci. A set of closely linked markers might, collectively, provide a more accurate measure of marker similarity than any one marker alone. Thus, we have extended the APM method to use simultaneously information from several marker loci (Lange and Weeks 1990). For clarity, we will consider only two marker loci A and B below. The results easily generalize to several marker loci. The most obvious definition of similarity at several marker loci is to take the sum of the individual marker similarities. That is, for marker loci A and B,

$$Z_{ij} = Z_{ij}^A + Z_{ij}^B.$$

For a pedigree, we then define

$$Z = \sum_{i<j} Z_{ij} = \sum_{i<j} Z_{ij}^A + \sum_{i<j} Z_{ij}^B$$

$$= Z^A + Z^B.$$

The mean is easily computed as

$$E(Z) = E(Z^A) + E(Z^B),$$

but the variance poses more difficulties since

$$Var(Z) = Var(Z^A) + Var(Z^B) + 2\,Cov(Z^A, Z^B).$$

Because of the single locus results it suffices to compute

$$Cov(Z^A Z^B) = E(Z^A Z^B) - E(Z^A)E(Z^B).$$

Notice that since

$$E(Z^A Z^B) = E\left[\left(\sum_{i<j} Z_{ij}^A\right)\left(\sum_{i<j} Z_{ij}^B\right)\right],$$

we must evaluate terms such as $E(Z_{ij}^A Z_{km}^B)$.

As before,

$$Z_{ij}^A = E\left[U_{ij}^A \mid \text{marker genotypes of i and j}\right],$$

and similarly for Z_{km}^B and U_{km}^B.

Since U_{ij}^A and U_{km}^B are conditionally independent given the marker genotypes of i, j, k, and m,

$$Z_{ij}^A Z_{km}^B = E\left[U_{ij}^A U_{km}^B \mid \begin{matrix}\text{marker genotypes} \\ \text{of i, j, k, and m}\end{matrix}\right].$$

Thus, $E(Z_{ij}^A Z_{km}^B) = E\left[U_{ij}^A U_{km}^B\right]$.

The only way in which U_{ij}^A and U_{km}^B can influence one another is for identity-by-descent sharing at one locus to increase the chance of identity-by-descent sharing at the other locus. Thus, we can compute the expectation $E[U_{ij}^A U_{km}^B]$ by conditioning on the combined identity-by-descent states of the i and j sampled gametes at locus A and the k and m sampled gametes at locus B. The probabilities of the four possible combined identity-by-descent states can be found using the two-locus kinship coefficients of Thompson (1988). For details, see Weeks and Lange (1991).

Application to Simulated Data

Using the simulation program SLINK (Ott 1989; Weeks et al. 1990b), we simulated two markers flanking the tuberous sclerosis disease locus, conditional on the structure and affection status of the nine tuberous sclerosis pedigrees from Janssen et al. (1990). The recombination fraction between the left marker M1 (3 alleles, heterozygosity = 0.53) and the disease locus was taken as 0.01, and the recombination fraction between the right marker M2 (4 alleles, heterozygosity = 0.65) and the disease locus was taken as 0.02. While the maximum multipoint lod score using marker data on the affecteds alone was only about 0.90, the APM method detected significant linkage (Table 2). When the intermediate weighting function $f(p) = 1/\sqrt{p}$ is used, the multilocus statistic is slightly more significant than either of the single locus statistics. In the two examples given in Weeks and Lange (1991), the superiority of the multilocus statistic is much more evident.

In order to investigate the distribution of the multilocus APM statistic under the null hypothesis of no linkage, we simulated the segregation of the two markers M1 and M2, independently of disease status. Assuming no interference, the recombination fraction between the markers is 0.0296. Table 3 summarizes results for the multilocus APM statistics. As we observed previously with the single locus statistic (Weeks and Lange 1988), the tails, skewness, and kurtosis of the APM statistic increase, as the influence of allele frequency increases. Figure 2 provides a histogram of the APM statistic for the intermediate weight $f(p) = 1/\sqrt{p}$.

Table 2: Application of the single locus and multilocus APM statistics to simulated tuberous sclerosis data; θ = 0.0296 between the flanking markers.

Function	M1	M2	Multilocus (P-value)
$f(p) = 1$	2.99356	-0.01646	2.27595 (0.0114)
$f(p) = 1/\sqrt{p}$	2.22309	1.13318	2.26211 (0.0118)
$f(p) = 1/p$	0.22274	1.83509	1.62172 (0.0524)

Table 3: Simulation results for the multilocus test statistic, based on 5,000 trials.

Function	Mean (Variance)	Skewness[a] (Kurtosis[a])	Empirical P-value
$f(p) = 1$	0.01390 (0.99221)	0.17897 (-0.05648)	0.01620
$f(p) = 1/\sqrt{p}$	0.01232 (1.00957)	0.25233 (0.20503)	0.01840
$f(p) = 1/p$	-0.00577 (1.00060)	1.05786 (2.38970)	0.06280

[a]For 5,000 trials, a skewness of 0.057 is significant at the 0.05 level, a skewness of 0.081 is significant at the 0.01 level. A kurtosis of 0.114 is significant at the 0.05 level, and a kurtosis of 0.161 is significant at the 0.01 level.

Function	Upper Fifth Percentile[b]	Upper First Percentile[b]
$f(p) = 1$	1.711	2.448
$f(p) = 1/\sqrt{p}$	1.714	2.622
$f(p) = 1/p$	1.754	3.073

[b]These are empirical percentiles. For a standard normal variate, the theoretical upper fifth and first percentiles are 1.645 and 2.326.

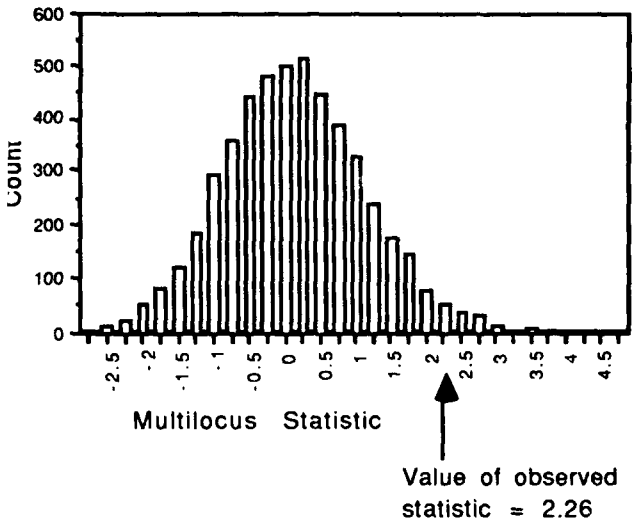


Figure 2: Tuberous sclerosis data: histogram of the multilocus test statistic, based on 5,000 trials, when $f(p) = 1/\sqrt{p}$. The theoretical standard normal P-value is 0.0118, while the empirical P-value is 0.01840.

Discussion

The affected-pedigree-member (APM) method of linkage analysis requires marker typing of only the affected individuals in a pedigree. More importantly, the APM method requires no assumptions about the mode of inheritance of the disease. This may provide an advantage over traditional methods of linkage analysis, which require an explicit, and often unverifiable, disease model. Although it is reasonable to expect the APM method to exhibit severely reduced power relative to conventional linkage lod score methods, in the presence of genetic heterogeneity it may, in fact, perform better. The multiple locus extension discussed here should make the APM method a more versatile and powerful tool for genetic epidemiologists.

References

Baron M (1990) Genetic linkage in mental illness. Nature **346**:618

Clerget-Darpoux F, Babron MC, Bonaiti-Pellie C (1990) Assessing the effect of multiple linkage tests in complex diseases. Genet Epidemiol 7:245-253

Day NE, Simons MJ (1976) Disease susceptibility genes - their identification by multiple case family studies. Tissue Antigens **8**:109-119

de Vries RRP, Fat RFM, Lai A, Nijenhuis LE, Van Rood JJJ (1976) HLA-linked genetic control of host response to *Mycobacterium leprae*. Lancet **ii**:1328-1330

Fishman PM, Suarez B, Hodge SE, Reich T (1978) A robust method for the detection of linkage in familial diseases. Am J Hum Genet **30**:308-321

Green JR, Woodrow JC (1977) Sibling method for detecting HLA-linked genes in a disease. Tissue Antigens **9**:31-35

Haseman JK, Elston RC (1972) The investigation of linkage between a quantitative trait and a marker locus. Behav Genet **2**:3-19

Janssen LAJ, Sandkuyl LA, Merkens EC, Maat-Kievit JA, Sampson JR, Fleury P, Hennekam RCM, Grosveld GC, Lindhout D, Halley DJJ (1990) Genetic heterogeneity in tuberous sclerosis. Genomics **8**:237-242

Karigl G (1981) A recursive algorithm for the calculation of identity coefficients. Ann Hum Genet **5**:299-305

Karigl G (1982) A mathematical approach to multiple genetic relationships. Theor Pop Biol **21**:379-393

Kerem B-S, Rommens JM, Buchanan JA, Markiewicz D, Cox TK, Chakravarti A, Buchwald M, Tsui L-C (1989) Identification of the Cystic Fibrosis gene: genetic analysis. Science **245**:1073-1080

Lange K (1986a) The affected sib pair method using identity by state relations. Am J Hum Genet **39**:148-150

Lange K (1986b) A test statistic for the affected-sib-set method. Ann Hum Genet **50**:283-290

Lange K, Weeks DE (1990) Linkage methods for identifying genetic risk factors. World Rev Nutr Diet **63**:236-249

Martinez M, Khlat M, Leboyer M, Clerget-Darpoux F (1989) Performance of linkage analysis under misclassification error when the genetic model is unknown. Genet Epidemiol **6**:253-258

Ott J (1989) Computer-simulation methods in human linkage analysis. Proc Natl Acad Sci USA **86**:4175-4178

Penrose LS (1935) The detection of autosomal linkage in data which consist of pairs of brothers and sisters of unspecified parentage. Ann Eugen **6**:133-138

Riordan JR, Rommens JM, Kerem B-S, Alon N, Rozmahel R, Grzelczak Z, Zielenski J, Lok S, Plavsic N, Chou J-L, Drumm ML, Iannuzzi MC, Collins FS, Tsui L-C (1989) Identification of the Cystic Fibrosis gene: cloning and characterization of complementary DNA. Science **245**:1066-1073

Rommens JM, Iannuzzi MC, Kerem B-S, Drumm ML, Melmer G, Dean M, Rozmahel R, Cole JL, Kennedy D, Hidaka H, Zsiga M, Buchwald M, Riordan JR, Tsui L-C, Collins FS (1989) Identification of the Cystic Fibrosis gene: chromosome walking and jumping. Science **245**:1059-1065

Suarez BK, Hodge SE (1979) A simple method to detect linkage for rare recessive diseases: an application to juvenile diabetes. Clin Genet **15**:126-136

Suarez BK, Rice JP, Reich T (1978) The generalized sib pair IBD distribution: its use in the detection of linkage. Ann Hum Genet **42**:87-94

Thompson EA (1988) Two-locus and three-locus gene identity by descent in pedigrees. IMA J Math Applied in Medicine & Biology **5**:261-279

Thomson G, Bodmer W (1977) The genetic analysis of HLA and disease association. In: HLA and disease. Munksgaard, Copenhagen, pp 84-93.

Weeks DE, Brzustowicz L, Squires-Wheeler E, Cornblatt B, Lehner T, Stefanovich M, Gilliam TC, Ott J, Erlenmeyer-Kimling L (1990) Report of a workshop on genetic linkage studies in schizophrenia. Schiz Bull **16**:673-686

Weeks DE, Lange K (1988) The affected-pedigree-member method of linkage analysis. Am J Hum Genet **42**:315-326

Weeks DE, Lange K (1991) A multilocus extension of the affected-pedigree-member method of linkage analysis. Submitted

Weeks DE, Lehner T, Squires-Wheeler E, Kaufmann C, Ott J (1990a) Measuring the inflation of the lod score due to its maximization over model parameter values in human linkage analysis. Genet Epidemiol **7**:237-243

Weeks DE, Ott J, Lathrop GM (1990b) SLINK: a general simulation program for linkage analysis. Am J Hum Genet **47**:A204

A RECURSIVE PARTITIONING ALGORITHM FOR CLUSTER ANALYSIS

Joseph S. Costa, Jr.
National Security Agency
9800 Savage Road
Fort George G. Meade, MD 20755-6000

I. INTRODUCTION

In 1965, A.W.F. Edwards and L.L. Cavalli-Sforza introduced a method for cluster analysis based on a recursive partitioning strategy over a minimum-variance clustering criterion. Although this method has been called "intuitively appealing", it was dismissed by Gower (1967) and others because of its computational infeasibility. It has been suggested on numerous occasions that some computationally efficient method be found to search an intelligently-chosen subset of the set of all possible partitions for a (hopefully) near-optimal solution. In this paper, one such method is introduced which borrows from the Classification and Regression Trees (CART) classification paradigm of Breiman, Friedman, Olshen and Stone (1984).

II. THE CLUSTERING ALGORITHM

1. Building the Clustering Tree

Consider a set S of n observations in p variables and represent any arbitrary observation by the vector $\underline{x} = (x_1, x_2, \ldots, x_p)$. As a first step, we would like to partition this set of n p-dimensional vectors into two subsets based solely on the observations given. The method used by this algorithm is an application of the *standard split* found in the CART classification scheme. In order to define the splitting rules, we generate all sets of the form

$$\{ x_j \leq c \}, \; j = 1, \ldots, p .$$

Geometrically, sets of this type are regions bounded by $(n-1)$-dimensional hyperplanes parallel to the co-ordinate axes which are selectively passed through n-space precisely in the center (with respect to variable j) of every pair of data points. Each of these hyperplanes will specify a partition of S into two new sets, S_1 and S_2.

The cs can be easily determined by sorting the j^{th} component of all the observations, then selecting values halfway between each successive pair of x_j s. In this fashion, the algorithm is guaranteed to create the least number of hyperplanes that will still produce every partition of S possible using such a method (in fact, for any j there are at most $(n-1)$ of them).

2. The "Goodness-of-Split" Criterion

Of course, some measure of the effectiveness of a split is needed in order to choose one of these $p(n-1)$ (at most) partitions as best. As in the original Edwards & Cavalli-Sforza algorithm, the minimum-variance criterion was selected. Although this measure can be somewhat sensitive to outliers, it has been tested by a number of researchers and shown to be effective in a wide range of clustering situations (Blashfield 1976, Milligan 1983).

More explicitly, from the standard set of splits generated as above, the "optimal" split is chosen to be that which maximizes the quantity

$$\mathrm{VAR}(S) - [\mathrm{VAR}(S_1) + \mathrm{VAR}(S_2)]$$

After the best split has been selected the algorithm proceeds recursively, splitting S_1 and S_2, then the best splits of these subsets, and so on.

3. Finding the "Optimal" Subtree

Certainly, if we allow the partitioning process to continue to completion, we will have an overspecification of the data structure, with each terminal node of the clustering tree containing a very few data points. A method must be found, therefore, to select the subtree of this complete tree that most accurately represents the gross structural characteristics of the data. This problem is exactly the "number of clusters" question that has been addressed repeatedly in the literature since the mid-1960s. Although no general analytic method has yet been found, a number of statistical or heuristic stopping rules have been employed with varying degrees of success (see Milligan 1985 for a comprehensive review and analysis).

As a hierarchical clustering method, recursive partitioning is amenable to the application of many of these stopping rules. Following a thorough search of the literature

and testing on constructed data sets, two such rules were found that seem to perform quite well in tandem with the recursive partitioning algorithm. These stopping rules are due to Calinski & Harabasz and Duda & Hart.

The approach of Calinski and Harabasz is to find that clustering of the data which maximizes the Variance Ratio Criterion (VRC)

$$\text{VRC} = \frac{\text{BGSS}}{(k-1)} \Big/ \frac{\text{WGSS}}{(n-k)},$$

where BGSS and WGSS are the between- and within-cluster sums of squares, n is the number of data points in the set, and k is the number of clusters in the current partition. This method was implemented by ordering all splits in the clustering tree according to the splitting criterion and computing the VRC for all subtrees of the complete tree created by recursively pruning away the lowest-rated remaining split. The subtree with the maximum VRC is then selected to represent the optimal clustering of the data.

The method of Duda and Hart is statistical in nature, and is applied during the initial "growing" of the clustering tree. The best split at each node (as defined in Section II.2) is examined and the ratio

$$\frac{J_e(2)}{J_e(1)},$$

where $J_e(1)$ is the WGSS of the node prior to the split and $J_e(2)$ is the WGSS over the pair of subsets resulting from the application of the split, is used as a test of the null hypothesis that the initial set of samples was drawn from a normal population with mean $\hat{\mu}$ and covariance matrix $\hat{\sigma}^2 I$. A rough estimate of the sampling distribution of the J_e s may be formulated, yielding the final test: Reject the null hypothesis (i.e., split the node) at the p-percent level of significance if

$$\frac{J_e(2)}{J_e(1)} < 1 - \frac{2}{\pi d} - \alpha\sqrt{\frac{2(1-8/(\pi^2 d))}{nd}},$$

where d is the dimensionality of the data, n is the number of data points in the node, and α is determined as usual by

$$p = 100\int_{\alpha}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-(1/2)u^2} du$$

Of course, the performance of the Duda-Hart rule will be directly related to the agreement of the data with the underlying assumptions of normality and form of the covariance matrix, and therefore some care should be exercised when using it with data with a wildly asymmetric or otherwise unusual distribution.

These two stopping rules were also found to work well in concert with each other. Although use of the Duda-Hart rule does require specification of the control parameter α, this requirement makes it useful for interactive examination of the stability of clustering solutions. In addition, the Duda-Hart criterion is able to test the one-cluster hypothesis while the Calinski-Harabasz VRC is not.

III. ALGORITHM PERFORMANCE

1. Experimental Design and Data Generation

In order to analyze the performance of this algorithm and compare it with other clustering methods, a series of Monte Carlo tests were undertaken. The structure of these tests followed very closely that used by Milligan (1983, 1985). Data sets containing 50 points each were generated in accordance with a structured experimental design. The design was defined by three factors: number of clusters, dimensionality of the data, and distribution of data points across clusters. The number of clusters in each data set ranged from two to five and each set was embedded in either four, six, or eight dimensions. In order to ensure testing over disparate cluster sizes, the distribution of points across clusters was varied according to the following schemes : Type A - points evenly distributed across clusters; Type B - 60% of points in one cluster; or Type C - 10% of points in one cluster.

The experimental design thus contained 36 cells, each of which were replicated three times for a total of 108 data sets.

The method of cluster generation was chosen to produce the characteristics of internal cohesion and external isolation noted (Milligan 1983, Everitt 1980) as being indicative of natural cluster structure. In order to satisfy the internal cohesion requirement, all clusters were composed of points drawn from truncated (at 1.5 standard deviations) multivariate normal distributions with standard deviations on each dimension chosen randomly from the range (0.25, 2.00). To ensure external isolation, clusters were not allowed to overlap in the first dimension; in fact, the separation between the means of two adjacent clusters in this dimension was defined by a function of the form $u(\sigma_1 + \sigma_2)$, where σ_1 and σ_2 were the standard deviations (in the first dimension) of the two clusters, and u was a constant drawn from a uniform (1.75 , 2.25) distribution. The positions of cluster centers in the remaining dimensions were selected randomly from within a range 2/3 as large as that of the first dimension, so cluster

overlap was possible and frequently did occur in these dimensions.

This method of cluster generation, when shaped by the factors comprising the experimental design, produced a body of data containing a wide variety of cluster shapes, sizes, and relative orientations that was felt to be a fair test of an algorithm's ability to discern true cluster structure.

2. Results I - Recovery of Cluster Membership

Each of the 108 data sets was analyzed for cluster structure with five different algorithms: single linkage, complete linkage, group average, k-means, and the author's implementation of the algorithm described in the previous section (henceforth called RPCLUS). For the k-means method, an average of results from three different runs was used for each data set, with a random ordering of the data for each run. The output of each of the methods was examined at the level of the correct number of clusters and this output was compared to the (known) true structure of the data by the use of the corrected Rand statistic (Rand 1971, Milligan 1983). This measure of similarity is defined as

$$R = \frac{\Sigma\Sigma N_{ij}^2 - (\Sigma\Sigma N_{i.}^2 N_{.j}^2)/N_{..}^2}{\Sigma N_{i.}^2/2 + \Sigma N_{.j}^2/2 - (\Sigma\Sigma N_{i.}^2 N_{.j}^2)/N_{..}^2}.$$

where N_{ij} is the number of data points placed in cluster i by the algorithm that are in cluster j of the actual solution, $N_{i.}$ and $N_{.j}$ are the marginal totals and N is the total number of data points.

The corrected Rand index assumes a value of 1.00 when the two clusterings are in total agreement. Its lower bound depends on the actual partition of the data, but is usually 0.00 or very slightly below. This index was chosen for reasons made clear in other studies (Milligan 1983): its high variability as compared to similar measures, as well as its consistency across different cluster scenarios. On the advice of such studies, a second index (Jaccard) was also used to evaluate the clusterings, but as it was in complete agreement with the corrected Rand statistic in regards to the relative performance of the algorithms, it was not felt necessary to include those values in the current report.

A table summarizing the results of the complete Monte Carlo testing with respect to each of the three design factors (number of clusters, dimensionality, and point distribution) is available from the author. For the data in this study, the complete linkage, RPCLUS and group average methods were clearly in a separate class from single linkage and k-means, with overall recovery means of 0.987, 0.986, 0.985, 0.955 and 0.909 respectively. There were strong similarity in the recovery means for RPCLUS, complete linkage and group average across almost all factors. The notable exceptions were the five-cluster and Type C distribution results, where RPCLUS paid the price for its minimum-variance characteristic of seeking evenly-sized partitions. However, even these differences were quite small; in fact, it was often one misclustered point that accounted for the discrepancies in the corrected Rand means.

3. Results II - Number of Clusters

Testing was also undertaken in order to measure the accuracy of the two stopping rules described in Section II.3 when used as constraints on the recursive partitioning algorithm. The same 108 data sets were used and for each of these sets a record was kept of the number of clusters indicated by each of the two rules. Overall, the Calinski-Harabasz rule exactly determined the number of clusters present in 91 data sets (84.2%), while it was within one cluster in either direction 105 times (97.2%). In the case of the Duda-Hart rule, experimentation revealed that optimum performance was obtained when α was set to 3.00 (corresponding to a 99.865% level of significance), and using this value the statistical rule produced the correct number of clusters 90 times (83.3%) and was within one cluster in 103 of the data sets (95.4%). Clearly both rules, when used in conjunction with the recursive partitioning algorithm, provide reliable information as to the number of clusters present in data with true cluster structure.

IV. ADVANTAGES

Monte Carlo testing thus has shown this new algorithm to be equivalent in performance to commonly-used techniques such as complete-linkage and the group average method. Why then should we be interested in another cluster analysis tool? Two reasons are readily apparent. First, cluster analysis is such an inexact science that it can never hurt to have a number of different approaches to use when beginning the analysis of a set of data. This is important because each type of algorithm is best suited to certain types of data. For example, the linkage methods used above are not likely to be effective for less spatially separated data, due to their tendency to string together adjacent clusters. Also, being a divisive method, RPCLUS would tend to yield different results from the agglomerative algorithms, results that may be more accurate at recognizing low-level cluster structure because the algorithm has had fewer steps in which to make irreversible mistakes. Another important consideration is that RPCLUS lends itself to very efficient implementation. The construction of standard splits boils down to a sorting operation which can be done very cheaply, and in addition,

the algorithm seems to be a natural for parallel processing. Other hierarchical clustering methods require comparisons across all data points at each step. RPCLUS, due to its recursive partitioning strategy, needs only to keep track of one portion of the data at a time and a parallel machine can readily farm out parts of the work to subsets of its computational resources.

V. SUMMARY

A new algorithm for cluster analysis has been introduced which draws both from earlier clustering efforts and recent techniques developed for use in classification problems. The algorithm makes good intuitive sense and has been shown in Monte Carlo tests to perform equally as well as many other methods used frequently in multivariate data analysis both for the recovery of cluster membership and for determining the number of clusters in a data set. Just as important, the analyses produced by this method are representable in a simple format that makes understanding of data structure easy and intuitive. In addition to its value as an alternative tool for the data analyst, this new algorithm also possesses computational advantages over some other popular methods that may make it more suitable for parallel implementations on very large data sets.

TECHNICAL NOTE : All data sets were created using random number generation routines contained in the S-PLUS data analysis software package (Statistical Sciences, Inc. - P.O. Box 85625 - Seattle, WA 98145). The complete linkage, single linkage and group average calculations were performed with subroutines also found in the S-PLUS package. K-means tests were run using software developed under DoD contract at Los Alamos National Laboratories. Recursive partitioning was done using an implementation developed by the author.

VI. REFERENCES

Blashfield, R.K. (1976). Mixture model tests of cluster analysis: accuracy of four agglomerative hierarchical methods. *Psychological Bulletin, 83*, 377-388.

Breiman, L. (1989). Suggested use of a tree approach to cluster speech vectors. Informal response to Department of Defense statistical problem.

Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.H. (1984). Classification and Regression Trees. Wadsworth : Belmont, CA.

Calinski, T. & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics, 3*, 1-27.

Duda, R.O. & Hart, P.E. (1973). Pattern Classification and Scene Analysis. New York: Wiley.

Edwards, A.W.F. & Cavalli-Sforza, L.L. (1965). A method for cluster analysis. *Biometrics, 21*, 362-375.

Everitt, B.S. (1980). Cluster Analysis (2nd ed.). Wiley : New York.

Gower, J.C. (1967). A comparison of some methods of cluster analysis. *Biometrics, 23*, 623-637.

Harding, E.F. (1967). The number of partitions of a set of N points in k dimensions induced by hyperplanes. *Proceedings of the Edinburgh Mathematical Society, 15*, 285-289.

Kaufman, L. & Rousseeuw, P.J. (1990). Finding Groups in Data : An Introduction to Cluster Analysis. Wiley : New York.

Milligan, G.W. & Cooper, M.C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika, 50*, 159-179.

Milligan, G.W. & Isaac, P.D. (1980). The validation of four ultrametric clustering algorithms. *Pattern Recognition, 12*, 41-50.

Milligan,G.W., Soon, S.C. & Sokol, L.M. (1983). The effect of cluster size, dimensionality, and the number of clusters on recovery of true cluster structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 5*, 40-47.

Mojena, R. (1977). Hierarchical grouping methods and stopping rules: An evaluation. *The Computer Journal, 20*, 359-363.

Rand, W.M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association, 66*, 846-850.

Scott, A.J. & Symons, M.J. (1971). On the Edwards and Cavalli-Sforza method of cluster analysis. *Biometrics, 27*, 217-219.

Improving Classification Trees with Simulated Annealing

Clifton D. Sutton*
George Mason University

Abstract

Classification trees produced by a recursive partitioning scheme such as CART are not guaranteed to be the best tree structured classifiers possible, partly because the sequential manner by which they are formed does not allow for "looking ahead". In some cases, altering trees produced by CART by shifting the partition boundaries results in improved prediction rules. Simulated annealing can be used to efficiently search for trees which may perform better than those produced by CART.

Introduction

In the general classification problem, it is known that each case in a sample belongs to one of a finite number of possible classes, and given a set of measurements for a case, it is desired to correctly predict to which class the case belongs. A classifier is a rule which assigns a predicted class membership based on a set of related measurements, $x_1, x_2, \ldots, x_K$. Taking the measurement space $\mathcal{X}$ to be the set of all possible values of $(x_1, \ldots, x_K)$, and letting $\mathcal{C} = \{c_1, c_2, \ldots, c_J\}$ be the set of possible classes, a classifier is just a function with domain $\mathcal{X}$ and range $\mathcal{C}$. It is normally desirable to use past experience as a basis for making new predictions, and so it follows that classifiers are usually constructed from a learning sample consisting of cases for which the correct class membership is known in addition to the associated values of $(x_1, \ldots, x_K)$.

Tree structured classifiers are constructed by making repetitive splits of $\mathcal{X}$ and the subsequently created subsets of $\mathcal{X}$ so that a hierarchical structure is formed, and a plurality rule can be used to assign a predicted class to each final subdivision of $\mathcal{X}$. The CART method of Breiman et al. [2] creates binary tree structured classifiers by recursively partitioning the measurement space $\mathcal{X}$ into disjoint subsets $A_1, A_2, \ldots, A_I$ as follows. The measurement space $\mathcal{X}$ is first divided into two disjoint sets by splitting it along a hyperplane. Next, one of the sets obtained from the first split is cut with a second hyperplane, resulting in the division of $\mathcal{X}$ into three disjoint sets. Successive splits can be similarly made until it is deemed that any further partitioning of $\mathcal{X}$ could not possibly result in a more accurate classifier. At each step, the selection of the splitting hyperplane results from an attempt to minimize overall tree impurity.

It is clear that after $I - 1$ splits have been made, the measurement space, as well as the learning sample, has been separated into I disjoint sets. Letting $A_1, \ldots, A_I$ be the subsets making up the partition of $\mathcal{X}$, these sets can be used in the following way to construct a classifier. If the value of $(x_1, \ldots, x_K)$ for an object to be classified belongs to the set A_i, then the predicted class for this object is the dominant class of the members of the learning sample which belong to A_i. That is, once the measurement space has been partitioned into sets $A_1, \ldots, A_I$, a classifier can be created by using a simple plurality rule to determine a mapping from $\{A_1, \ldots, A_I\}$ to $\mathcal{C}$.

One of the crucial issues that is addressed by the CART method is the decision of how finely $\mathcal{X}$ should be partitioned into a collection of disjoint subsets, since too few or too many subsets will result in a loss of class prediction accuracy. On one hand, if too small of a tree is chosen, not all of the information present in the learning sample will be fully utilized. Thus, the misclassification rate will be higher than the rate for a larger tree having a finer partitioning of $\mathcal{X}$. On the other hand, if a tree has too many terminal nodes, it may be "paying too much attention " to the specific features of the learning sample and may not accurately reflect the structure of the larger population from which the sample was taken. Although the resubstitution misclassification rate decreases as the complexity of the tree increases, it is not necessarily true that the probability of misclassification becomes smaller. For instance, it is possible that a tree with a large enough number of nodes can have a resubstitution misclassification rate of zero, but such a tree may do a very poor job of predicting class membership for new observations.

CART carefully selects a good value for the number of sets in the partition of $\mathcal{X}$ using either cross validation or the test sample method. To do this, CART first "grows" a tree having too many sets in the partition and then successively "prunes" this tree by recombining subsets of $\mathcal{X}$ that were previously split until the right sized tree

*The author gratefully acknowledges support from NSF Grant DMS-9002237. He would also like to thank Sarah Rosenblum, R. Duane King, and Kelly J. Buchanan for their assistance.

is obtained, using either cross validation or a test sample (which is a subset of the original learning sample that is not used to construct the tree, but is solely used to assess the accuracy of the various candidates) to select the best classifier from the sequence of candidates encountered. In other words, CART is a stepwise procedure which initially considers all of the variables present and creates a tree which is too complex. Then, the choice of which variable splits will be used in the final tree is based upon the tree encountered in the pruning process which has the smallest estimated misclassification rate.

Even though the CART method carefully selects the right sized tree, the classifier obtained isn't necessarily the best classification tree possible. This is partly due to the fact that CART employs a "greedy" algorithm which prescribes a sequence of stepwise optimal moves and does not allow for "looking ahead" in order to examine the effects that a current decision will have on the ability to create subsequent splits leading to a good classification rule. This means that if the CART method yields a classification rule based on a partition of $\mathcal{X}$ having four sets, then that rule is the best one that can be obtained by CART's recursive partitioning and pruning scheme, but it is not necessarily the globally optimal solution if *all* ways of partitioning $\mathcal{X}$ into four sets are considered. That is, CART's reliance on a descent algorithm employing a sequential decision process which doesn't allow for "looking ahead" results in the fact that the classification rule that it produces may not be the best tree structured classifier possible. However, it should be noted that if the CART method is altered to allow it to consider one or more succedent moves at each step, then the required computation time would be vastly increased, making this way of altering CART to search for improved tree structured rules not very practical.

A procedure that could use CART's output to search for an improved tree structured rule would be to let CART's tree stipulate how many subsets the partition of $\mathcal{X}$ should include, and then do an exhaustive search for the best classification rule amongst all sensible partitions having that same number of subsets. However, unless the learning sample is rather small and simple there can be far too many competitors to examine, and the required computation time could be excessive.

Another way to possibly improve a classification tree that is produced by a recursive method such as CART is to shift the locations of the partition boundaries while retaining the overall nested partitioned structure resulting from the recursive partitioning algorithm. For instance, if CART produces a tree having $\mathcal{X}$ partitioned into four subsets, then one could search for a better tree from among the class of four subset partitions of $\mathcal{X}$ that have each split defined using the same variables that CART used, but possibly different locations for the cutting hyperplanes. For example, if the first decision point in the CART produced tree happens to be "Is $x_2 \leq 98.6$?", then only trees having an initial decision of the form "Is $x_2 \leq \gamma$?" will be considered as possible candidates for improvement. Although limiting the search for a better tree to the set of trees having the same general structure as the CART produced tree decreases the size of the candidate pool from the number that would result if all partitions having the same number of sets as the CART partition were considered, for large data sets it may still be infeasible to do an exhaustive search for the best such partition of $\mathcal{X}$.

As an alternative to a brute force search, one could begin with CART's tree and then gradually shift the locations of the partition boundaries in order to search for an improved tree. One way to do this would be to randomly shift the locations of the partition boundaries and then determine if the new partitioning of $\mathcal{X}$ reduces the resubstitution misclassification rate. If so, then the associated rule can be adopted as the best rule so far. If the random shifting does not yield improvement, then the new partition can be discarded and another attempt at shifting the boundaries can be done starting from the same configuration as before. Repeated attempts to decrease the resubstitution misclassification rate by shifting the partition boundaries belonging to the best rule so far may result in iterative improvement.

Using Simulated Annealing

An undesirable feature of the preceding scheme is that it may yield a solution which is locally optimal without being globally optimal as well. That is, the overall best partition (having the same general form as the CART produced partition) may be separated from the initial CART partition by a "ridge", and it may not be possible to reach the globally optimal tree from the initial tree if only *downhill* moves are allowed.

Simulated annealing is an iterative method of optimization which makes use of a random number generator. If the method of simulated annealing is combined with the idea of randomly shifting the partition boundaries to search for a better tree, then it may be possible to avoid getting stuck at a solution which is only locally optimal. This is due to the fact that simulated annealing allows for an occasional *uphill* move. It is hoped that if the parameters associated with a simulated annealing algorithm are properly selected, then successive random pertubations of the partition boundaries will eventually result in a classification tree which may perform better than the CART tree.

The general scheme for the basic version of simulated

annealing employed for this project can be described in the following way. At the start of each new iteration, the current configuration of a system is slightly altered in a random way to obtain a neighboring trial configuration. If the trial configuration is better than the current configuration (the configuration that existed at the start of the iteration step prior to the random altering), then the trial configuration is automatically accepted and taken to be the current configuration for the next step. However, if the trial configuration is not better than the current configuration for the iteration step, then the trial configuration is neither automatically accepted nor rejected as being the new current configuration for the next iteration step. Instead, a pseudo random number generator will be used to randomly decide whether or not to accept the uphill move. The probability with which a less favorable configuration is accepted depends upon a couple of factors: the degree to which the trial configuration is worse and the position of the iteration step in the whole sequence of steps which make up the annealing process. The more unfavorable a trial configuration is, the less likely it is to be accepted. Also, a less favorable configuration is more likely to be accepted in an uphill move if it occurs near the beginning of the annealing process than it would if it occurs after the process has been run through many iterations. If an uphill move is rejected, then the current configuration remains the same for the next iteration.

The simulated annealing approach was originally developed by Metropolis et al. [5] as a way of minimizing complex energy functions associated with N particle systems, and both the general scheme and the associated terminology are related to statistical mechanics and the behavior of N particle systems which are acted upon by a heat bath. Although the approach has been successfully applied to many problems having little to do with physics (see [1, 3, 4]), it is common practice to retain the physicists' terminology of energy and temperature. Basically, the energy E is some function of an N particle system that one desires to minimize, and the temperature T is a control parameter that effects the probability $e^{-\Delta E/T}$ of an uphill move being accepted. Here ΔE is the increase in the energy function associated with the uphill shift under consideration.

It is common practice to lower the temperature (thus decreasing the probabilities of accepting uphill moves) as the annealing process continues. As typically implemented, this "cooling" is carried out using two additional parameters, the temperature length L and the cooling ratio r, along with the temperature T. Here L is a fixed positive integer, r is a fixed real number belonging to $(0, 1)$, and T is a variable parameter which takes on a decreasing sequence of positive real values $T_1, T_2, T_3, \ldots$. The temperature T is kept constant while a sequence of L trial configurations are considered. Then T is decreased by multiplying its value by the cooling ratio r ($T_{i+1} = rT_i$), and L additional iterations are done with the resulting lower temperature. This procedure continues until it is deemed that significant further improvement is unlikely, at which point the process is terminated according to an appropriate stopping rule. It is hoped that when the process reaches the "frozen state" (the point at which the temperature is not lowered and no additional shifts are tried), the current solution is close to being globally optimal.

In summary, the temperature is a control parameter which determines the likelihood of uphill moves being accepted. At the start of the annealing process, the temperature is high so that hopefully enough uphill moves will be made so that the configuration will not be trapped at a local minimum. As the annealing process continues, the temperature is lowered so that the series of trial configurations can move more efficiently towards a final configuration having low energy. In many applications of the simulated annealing algorithm, key issues are the determination of good choices for the initial temperature T_1, the temperature length L, and the cooling ratio r, along with developing a good method with which to randomly shift to new trial configurations.

In the classification tree problem, the members of the learning sample can play the role of the N particles, and E can be the resubstitution estimate of the misclassification rate, which is just the proportion of the learning sample that would be misclassified by the classification tree. It should be noted that while the resubstitution estimate of the misclassification rate is not a good criteria to use in the selection of the right sized tree, there seems to be nothing wrong with minimizing the resubstitution estimate of the misclassification rate when an attempt is made to find the best tree structured classification rule from among all those having the same number of subsets in the partition of $\mathcal{X}$ and the same nested structure of partition subsets.

Description of the Experiment

A waveform recognition problem due to Breiman et al. [2] is employed as a test bed for the improvement scheme. The data was constructed using random number generators so that a sufficiently large amount of independent observations could be made available for a satisfactory assessment of the performance of the simulated annealing technique.

The data points used are 21-dimensional vectors of the form $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,21})$. Each data point consists of a random convex combination of two fixed wave-

forms, to which Gaussian noise has been added. The fixed waveforms used were selected from **a**, **b**, and **c**, where

$$\mathbf{a} = (0,1,2,3,4,5,6,5,4,3,2,1,0,0,0,0,0,0,0,0,0),$$

$$\mathbf{b} = (0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,5,4,3,2,1,0),$$

and

$$\mathbf{c} = (0,0,0,0,0,1,2,3,4,5,6,5,4,3,2,1,0,0,0,0,0).$$

Data points belonging to Class 1 are of the form $\mathbf{x}_i = u_i\mathbf{a} + (1-u_i)\mathbf{b} + \mathbf{e}_i$, where u_i is a uniform (0,1) random deviate and $\mathbf{e}_i = (e_{i,1}, e_{i,2}, \ldots, e_{i,21})$ is a vector of observed values for 21 i.i.d. Gaussian random variables. Similarly, data points belonging to Class 2 are of the form $\mathbf{x}_i = u_i\mathbf{a} + (1-u_i)\mathbf{c} + \mathbf{e}_i$, and members of Class 3 are of the form $\mathbf{x}_i = u_i\mathbf{b} + (1-u_i)\mathbf{c} + \mathbf{e}_i$.

The task of identifying the proper class associated with one of the random vectors described above is made difficult due to two primary reasons. First, whenever u_i is close to either 0 or 1, it is not easy to distinguish between two possible classes. For instance, a Class 1 vector with u_i close to 0 may look very much like a Class 3 observation having u_i close to 1 — the general shape of both will resemble waveform **b**. The identification of the correct class is further hindered by the additive Gaussian noise. It should be noted that the standard deviation associated with the noise is rather large compared to the average magnitude of the waveform coordinates.

A total of twenty data sets consisting of 300 waveforms each were generated. The class for each observation was randomly selected, with the three possible classes all having the same likelihood of being chosen. CART was then applied to construct tree structured classifiers. Both the Gini and the twoing splitting criteria were used with each data set, and so altogether CART was used to produce forty classification trees. In all cases, linear combination splits using more than one variable were disallowed. By using only single variable splits, all of the decisions associated with the tree structured rules are of the form "Is $x_{i,j} \leq \gamma$?". This restriction on tree formation greatly simplified the programming of the annealing algorithm, and also led to decreased running times.

The number of nodes in the classification trees produced by CART ranged from 3 to 17. Since the level of difficulty of writing a program to perform the simulated annealing improvement scheme increases with the complexity of the tree, it was decided to only investigate the performance of the method on tree structured rules possessing three, four, or five nodes.

Of the fifteen trees having five or fewer nodes, some were identical, and there were only thirteen distinct trees to be examined. For each of the thirteen classification trees considered, numerous variations of the simulated annealing technique were investigated. One source of variation was due to using different values for the cooling schedule control parameters T_1, L, and r. Values tried for T_1 were 0.000625, 0.00125, 0.0025, and 0.005. The cooling ratio r was assigned the values 0.25, 0.5, 0.75, and 0.875, and 100, 150, and 200 were used for the temperature length L.

Various methods for obtaining trial configurations were also considered. One way to produce a trial configuration from the current one is to shift only a single boundary of the partition. Alternatively, more than one partition boundary could be shifted to create a new tree. Both the single shift approach and the multiple shifts approach were investigated. With the single shift method, a boundary is randomly selected to be moved, and each time a shift is to be made, each boundary in the tree is given an equal chance of being selected. Alternatively, for every perturbation in the multiple shifts scheme, each boundary is shifted with probability 0.5, independent of what occurs with the other boundaries. If no boundaries are selected for movement, additional attempts are made until at least one boundary shift occurs.

Another issue connected with the shifting procedure concerns the magnitude of the shifts. If it is decided that a boundary will be shifted, the shift could be slight so that only a small number of data points fall into a node different from the one that they were in previously, or the size of the shift could be much larger so that an appreciable proportion of the data points change their node membership.

In the algorithm used to perform the annealing, a parameter K is used to specify the greatest number of data points that can change node membership when a boundary shift is performed. When a boundary is selected to be shifted, the direction of the shift is first determined and then the number of points to change node membership is randomly chosen from $\{1, 2, \ldots, K\}$. Values tried for K were 5, 10, 20, and 30.

Another option that was explored deals with the initial configuration for the annealing process. The use of CART's tree as a starting point was investigated, as was using a randomly selected configuration having the same general structure as the CART partitioning.

A very basic version of the general simulated annealing algorithm was used in order to attempt to produce more accurate classification rules. It was hoped that shifting the boundary locations to configurations having a lower resubstitution misclassification rates would produce tree structured rules for which the misclassification rates would be lower when new, independent

data was applied, and this led to using the resubstitution misclassification rate as the energy function to be minimized. That is, the energy was taken to be E = (no. obs. misclassified)/300.

A geometric cooling schedule having fixed length was employed. This means that the temperature levels in the sequence $T_1, T_2, T_3, \ldots$ were related through the equality $T_{i+1} = rT_i$, and a fixed number, L, of trial configurations were considered at each temperature level in the sequence. The stopping rule utilized was as follows: the annealing process was terminated whenever none of the L trial configurations generated at a given temperature level resulted in an accepted shift and a new value of E.

The following description is a short summary of the algorithm. At each temperature level encountered, L trial configurations are generated by shifting one or more partition boundaries. For each trial configuration, the energy E (which is just the resubstitution misclassification rate) is determined. If E is reduced, the shift is accepted and another trial configuration is obtained by jiggling the boundaries of this newly accepted partition. If E is not lower for a trial configuration, then the configuration is accepted in an uphill move with probability $e^{-\Delta E/T}$ and rejected otherwise. Here ΔE is the increase in energy associated with the trial configuration, and T is the current temperature. If a trial configuration is not accepted, then the next trial configuration is obtained starting from the same tree that was altered previously; and a new random perturbation of the boundaries is produced. If one or more of the trial configurations at a given temperature T results in an accepted shift with a change in E, then the temperature T is lowered to rT, and L additional trial configurations are produced. Otherwise, the process is terminated and the current configuration is taken to be the classification rule.

The search for improved classification rules was pursued by applying the simulated annealing algorithm to the trees formed from the waveform data. The previously specified values of T_1 (0.000625, 0.00125, 0.0025, and 0.005), r (0.25, 0.5, 0.75, and 0.875), L (100, 150, and 200), and K (5, 10, 20, and 30) were used in the annealing process. Every possible combination of these parameter values was tried, making a total of 192 combinations for each tree. Also, four different variations of the annealing algorithm (random start/single shift, CART start/single shift, random start/multiple shifts, and CART start/multiple shifts) were tried with each combination of parameter values. So, in all, 768 distinct ways of doing the annealing were tried with each tree.

The annealing algorithm was implemented by running C programs on an Intel Hypercube concurrent computer. For each tree and each distinct case of the annealing scheme, eight annealing trials were performed (and so for each tree, a total of 6144 attempts were made to minimize the resubstitution misclassification rate using simulated annealing). Each of the eight trials used different seeds for the pseudo random number generator, and so the trials did not always result in the same tree configuration at the frozen state. Each trial was performed on a different node of the Hypercube, and the results from each of the eight trials were then sent to a host program where they were compared and summarized.

Results

A vast amount of computer time was required in order to carry out the experiment. Some jobs in which 6144 annealing trials were performed on a single tree took nearly half a day to complete. After all of the runs were made, the results from this experiment were closely examined to determine which combination of parameter values and which of the four variations of the annealing algorithm performed best. Also, the degree of improvement was assessed for the new trees produced.

The algorithm performance results will now be summarized by first considering each of the four variations of the annealing algorithm separately. For the variation which has the annealing process beginning with randomly chosen boundary locations and allows for only a single boundary to be shifted with each random perturbation, it turned out that for each tree the overall minimum misclassification rate was obtained on at least one trial for numerous combinations of parameter values. However, the following set of parameter values seemed to work best overall: $T_1 = 0.005$, $r = 0.875$, $L = 150$, and $K = 20$.

The second annealing scheme investigated in the search for optimal algorithm performance also used the single shift approach, but instead of a randomly chosen initial configuration, the CART boundaries were taken to be the starting point at which the annealing process began. For each tree, the minimum resubstitution misclassification rate obtained was exactly the same as it was for the "random start" variation. As before, numerous combinations of parameter values resulted in the minimum misclassification rate, but with this variation the following values were found to be most favorable: $T_1 = 0.00125$, $r = 0.875$, $L = 200$, and $K = 10$.

It can be noted that the best value for r is the same as it is for the "random start" scheme, but the optimizing values for K, L, and T_1 are slightly different. With the "CART start" variation, it seems to be better to use smaller values for K and T_1. These smaller values will lead to more moderate perturbations of the boundaries, as well as allow for fewer uphill moves. The result is a more controlled cooling, for which the energy function

decreases steadily towards a lower value. With the "random start" method, for which the starting point could be quite far from the optimal solution, a larger value for T_1 prevented the process from being stopped at a local minimum (which might be far from the overall minimum) by producing greater probabilities of escape via uphill moves. Also, a larger value for K allows for wilder movement of the boundaries, which seems appropriate if the starting points for the boundaries can be far from the minimizing locations.

The overall performance of the two single shift variations differed little. Both schemes reached the same minimum value for the resubstitution misclassification rates, and with the parameters set favorably, both schemes reached the minimum value with high frequency. However, the "CART start" variation was quicker for each tree examined, typically reducing the average number of perturbations required to reach the frozen state by a factor of about two or three.

For the third variation of the annealing scheme, the single shift procedure is replaced by the multiple shifts procedure, and a randomly selected initial point is employed. With regard to the best overall choice of parameters, this case leads to the selection of $T_1 = 0.005$, $r = 0.875$, $L = 200$, and $K = 10$. In all of the cases, the minimum value found was the same as it is for the other two schemes.

Comparing the set of parameters which worked best for this random start/multiple shifts scheme with the set that worked best for the random start/single shift scheme, it can be seen that T_1 and r are the same, but the values for L and K differ. When only a single shift is made for each perturbation, the best choice for K is 20, which can produce rather large boundary shifts. When multiple shifts are allowed, a smaller K works better. This observation may lead one to the tentative conclusion that the overall amount of shifting should not be allowed to be too large.

The final case is similar to the previous one in that it involves multiple shifts of boundaries, but instead of the randomly chosen starting boundaries, the CART tree is used as a starting point. This final case resulted in the the same minimum misclassification rates and yielded $T_1 = 0.0025$, $r = 0.75$, $L = 200$, and $K = 5$ as the best combination of parameter values. Overall, this variation did not work quite as well as the other three.

The lower initial temperature and the smaller value of K will result in a more controlled cooling than that which occurs with the larger parameter values of the random start/multiple shifts case (where $T_1 = 0.005$ and $K = 10$ worked best). When a comparison is made with the CART start/single shift variation (for which the best value for K is 10), it can be noted that the optimal value of K is smaller in this CART start/multiple shifts variation. This observation serves to reinforce the tentative conclusion reached earlier concerning the desirability of constraining the overall size of the maximum configuration shift.

To summarize, numerous ways of performing the simulated annealing resulted in the same minimum resubstitution misclassification rate for each tree, and so it appears that the technique is rather robust. However, with the randomly chosen starting points, the minimum rate is achieved with slightly greater frequency. On the other hand, with the CART starting points it generally took considerably less time to reach the minimum misclassification rate. Whether the "random start" or the "CART start" method was utilized, the single shift method performed a little better.

It seems that for all four variations a more gradual cooling, which results from a larger value of r, is superior to using a small value for r and obtaining a quicker cooling. Furthermore, with each of the four variations it was found that the performance deteriorated whenever T_1 was made too small. A small value of T_1 decreases the probability of uphill moves, and this resulted in a greater likelihood of the solutions being trapped at local minimums. It can also be observed that when a randomly selected starting point is used instead of the CART solution starting point, T_1 and K should be chosen to be larger in order to allow for a wilder shifting of the boundaries before the cooling severely limits the chance of escaping from local minimums. Also, it can be noted that the multiple shifts variations work better for smaller values of K than those which are preferred with the single shift variations. This leads one to conclude that it is best to limit the overall amount of shifting allowed. This notion is additionally supported by the fact that none of the four cases worked best when K was set as 30. To conclude these remarks concerning the parameter values, it should be stated that the experiment performed has not ruled out the possibility that the performance could be improved by using larger values for r and L. Unfortunately, larger values for r and L would require longer running times for the annealing programs.

The simulated annealing process produced new classification trees by randomly shifting partition boundaries until the resubstitution misclassification rate seemed to be lowered as much as possible. In each of the thirteen cases examined, the annealing experiment produced more than one tree having the lowest misclassification rate. In order to assess the typical amount of improvement due to the application of the algorithm to CART produced trees for this setting, it was decided to examine

the performance of the tree which was most frequently produced by the annealing process in each case.

For eleven of the thirteen cases considered, the resubstitution misclassfication rate for the classification tree produced by the annealing algorithm was lower than the resubstitution misclassification rate for the original CART produced tree, and for the other two cases the resubstitution misclassification rate was the same as it was for the CART rule. However, just because the resubstitution misclassification rate is typically reduced by the annealed trees, it is not necessarily true that the annealed trees are really more accurate. In order to assess the amount of improvement produced by the annealing procedure, the true misclassification rates of the resulting trees can be estimated using test samples consisting of observations which are independent of the observations that were used to create the trees. The accuracy of the CART produced trees can be assessed using the same test samples, and then the estimated misclassification rates can be compared.

Recalling that a total of twenty data sets of 300 observations each were originally generated in the same way, it is clear that test samples of independent observations can be produced in the following manner. To assess the accuracy of trees produced from a given data set, the observations in all of the other nineteen data sets can be combined to serve as a test sample. By doing this, test samples for each tree would consist of 5,700 observations, none of which were used in the construction of the tree being evaluated.

In ten of the thirteen cases, the unbiased estimate of the true misclassification rate provided by the test sample was lower for the tree which was sharpened by the annealing process. However, it should also be stated that the observed differences were typically quite small, and it is therefore natural to wonder whether or not all of the observed differences are indicative of any real difference in the predictive ability of the trees since it could be that the observed differences in the estimated misclassification rates can easily be attributed to chance. To investigate this query, hypothesis tests were performed (McNemar's test was used) in order to make inferences about the differences between dependent proportions. In one of the thirteen cases, the performance of the annealed tree is significantly better when tested at $\alpha = 0.01$, and in four of the thirteen cases, the performance of the annealed tree is significantly better when tested at $\alpha = 0.1$. None of the thirteen annealed trees were found to be significantly worse when $\alpha = 0.05$ tests were done, and only one was found to be significantly worse when $\alpha = 0.1$ tests were performed.

Overall, it seems reasonable to conclude that the application of the simulated annealing method is beneficial for the waveform data, and that if the resubstitution misclassification rate is decreased then the true misclassification rate may be slightly decreased. Of course, it could be argued that the typical amount of improvement is somewhat negligible since, on the average, the annealed trees were observed to do only a little better than the original CART trees. However, the small amounts of improvement could be largely attributed to the fact that the CART trees actually do a pretty good job with this data, and there simply wasn't a lot of room for improvement. In fact, when a brute force search for the lowest resubstitution misclassification rate was performed for all of the three and four node trees (with the searches limited to the class of trees having the same general struture as the CART trees), in each case it was determined that the annealing process reached the minimum rate possible. Furthermore, it was found that the minimum resubstitution misclassification rate was obtained by annealing much more quickly than by an exhaustive search — for the four node trees the average time required for a brute force search was greater than the time required for a set of eight annealing trials by a factor of about 650 (and for five node trees the difference would be much larger). All in all, the results of this simulated annealing experiment can be taken as encouragement that the method may be an efficient way to obtain improved tree structured classifiers in situations where CART leaves some room for improvement.

References

[1] Bonomi, E., and Lutton, J. L. (1984). The N-city travelling salesman problem: Statistical mechanics and the Metropolis algorithm. *SIAM Review* **26**, 551-568.

[2] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees.* Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, California.

[3] Johnson, D. S., Aragon, C. R., McGeoch, L. A., and Schevon, C. (1989). Optimization by simulated annealing: An experimental evaluation; Part I, Graph partitioning. *Oper. Res.* **37**, 865-892.

[4] Kirkpatrick, S., Gelatt, C. D., Jr., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* **220**, 671-680.

[5] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087-1092.

A Stratification Option for Regression Trees

Michael LeBlanc *

Department of Preventive Medicine and Biostatistics
University of Toronto, Toronto, Ontario, M5S 1A8

Abstract

A simple modification of the Classification and Regression Tree (CART) algorithm of Breiman, Friedman, Olshen and Stone (1984) that yields K-group stratifications is presented. Such stratifications can be useful for describing patient prognosis.

1 Introduction

Classification and regression trees have found applications in many fields including, pattern recognition, artificial intelligence and medicine. Trees have several advantages compared to classical methods; they are completely non-parametric and include powerful variable subset selection; they are robust to outliers in the covariate space and are easily used on a wide variety of data structures. In addition, they yield results that can be expressed as a binary decision tree that allows fast prediction and that is often easily interpreted. For applications in medicine it is the decision tree representation that is probably the greatest attraction to clinicians. The results are consistent with how some medical researchers think about certain problems, which can lead to easier interpretation and communication of statistical results.

Tree-based regression models are constructed by recursively partitioning the data and the covariate space into groups that minimize some measure of impurity, for instance residual sum of squares for continuous response data or binomial deviance for binary response data. The partitioning typically continues until there are only a few observations in each group and the binary tree representing the partitioning is large; this is done to avoid missing structure.

While tree-based methods have been available since Morgan and Sonquist (1963), advances in the methodology including not limiting the tree growth and using an optimal pruning algorithm with cross-validated estimates of prediction error to choose the size of the tree were introduced in the Classification and Regression Tree (CART) algorithm of Breiman, Friedman, Olshen and Stone (1984) (BFOS).

After choosing a tree of about the right size there can be a simplification of the description by further combining nodes that are close in terms of response and/or in the covariate space. These nodes need not be adjacent in the presented tree structure. In terms of studying a patients outcome, this recombination of many possible terminal nodes could lead to a few descriptive classes, say "good prognosis","fair prognosis", and "poor prognosis." Such prognostic stratifications can also be useful the development of staging schemes that can be used in the development of new clinical trials. The problem of development of prognostic stratification rules was the motivation for the technique presented here.

Below I outline a variation of the CART regression algorithm that recombines of possibly non-adjacent nodes, to yield a tree based stratification:

1. A tree is constructed and cost-complexity pruning is used, as in CART algorithm, to find the sequence of optimally pruned subtrees for any penalty α.

2. For each optimally pruned subtree a locally optimal $2, 3, 4, 5, \ldots$ group recombination of the nodes is found by a K-means type clustering algorithm.

3. The whole process is cross-validated as in CART. Therefore, the choice of number of strata can also be based on an estimate of prediction that is not overly optimistic.

The use of K-means like clustering to construct locally optimal K-ary splits of nominal covariates was investigated by Chou (1989). In addition, he proposes the development of "compound nodes" by using the terminal nodes of a tree to define a class variable. The algorithm presented here implements such a clustering scheme among all optimally pruned subtrees with the goal of finding good tree-based stratifications. Another tree-based stratification technique was implemented by Ciampi et. al (1988) for survival data.

*Research supported by the NSERC of Canada

2 Growing Trees

The data are assumed to consist of a vector of observations $(y_i, \mathbf{x}_i)$ $i = 1, ..., N$ observed from $(Y, \mathbf{X})$ where Y is the the response and $\mathbf{X}$ is a vector of covariates $\mathbf{X} = (X_1, X_2, ..., X_p)$.

Tree growing procedures recursively split the data and the covariate space into two groups. Splits are chosen based on the reduction in the impurity of a node or on a measure of dissimilarity in response between nodes. Define impurity at a node as the expected loss

$$i(t) = E[L(Y, \mu(t))|t],$$

where $\mu(t)$ minimizes the loss for node t. Let the expected cost a node t be

$$R^*(t) = i(t)P(t)$$

where $P(t)$ is the probability of falling into node t; an estimate of $R^*(t)$ is

$$R(t) = \int_{X \in B_t} L(Y, \hat{\mu}(t)) d\hat{F}_N$$

where B_t is the region corresponding to node t, $\hat{F}_N$ is the empirical distribution function and where frequently the loss functions $L(Y, \mu) = (Y - \mu)^2$ and $L(Y, \mu) = Y \log(\mu) - (1 - Y) \log(1 - \mu)$ used for continuous and binomial data respectively. Here, I follow Clark and Pregibon (1991) and Ciampi et al. (1987) in the use of the likelihood function for tree-based models.

Tree growing procedures calculate the reduction in impurity at all possible splits and choose a split that maximizes the reduction. The splitting continues until a large tree has been grown with only a few observations in each node. The entire partitioning process is usually represented by a tree T. Let $\tilde{T}$ denote the terminal nodes of tree T.

A tree-based regression model can be expressed in terms of a partition function $\tau(x) = t$ if $x \in B_t$ where B_t corresponds to a terminal region, and a decision rule $\nu(t) = \hat{\beta}_t$ where $\hat{\beta}_t$ is an estimate corresponding to that terminal node. Alternatively, the model can be expressed by step function regression function

$$\mu(\mathbf{x}) = \sum_{t \in \tilde{T}} \hat{\beta}_t I\{x \in B_t\}.$$

In the CART algorithm, the cost-complexity measure

$$R_\alpha(T) = \sum_{t \in \tilde{T}} R(t) + \alpha |\tilde{T}|,$$

where α is non-negative complexity parameter and $R(t)$ is the estimated cost of node t defined above, is used to assess the performance of a tree based model.

An optimally pruned subtree for any penalty α of the tree initially grown is T_1 if

$$R_\alpha(T_1) = \min_{T' \preceq T} R_\alpha(T'),$$

where " $\preceq$ " means "is a subtree of ", and it is the smallest optimally pruned subtree if $T_1 \preceq T''$ for every optimally pruned subtree, T''. Let $T(\alpha)$ denote the smallest optimally pruned subtree of T for complexity parameter α.

There is an efficient algorithm for obtaining $T(\alpha)$ for any α called the cost complexity pruning algorithm. It consists of finding the sequence of optimally pruned subtrees by repeatly removing branches for which the average reduction in impurity per split in the tree is small. The process yields a nested sequence of subtrees $T_m \prec ... \prec T_l \prec T_{l-1} ... \prec T_1 \prec T_0$, where T_m is the root node, and the sequence thresholds $\infty > \alpha_m > ... > \alpha_l > \alpha_{l-1} > ... > \alpha_2 > \alpha_1 > 0$, such that for the optimally pruned subtree $T(\alpha) = T(\alpha_l) = T_l$ for $\alpha_l \leq \alpha < \alpha_{l+1}$ (BFOS).

3 K-ary Stratification

A tree-based K-ary stratification will be defined to be a special case of the tree based model described in Section 2. That is we have a partition function $\tau(x) = t$ as before but now there is also the constraint that the decision rule $\nu(\cdot)$ must have only K values, where K is smaller than the number of terminal nodes. The tree-based regression model is then reduced to a piece-wise constant model with only K different prediction values.

One strategy is to find the best K-ary stratification among all subtrees of tree T. This scheme would by very computationally demanding because of the extremely large number of possible subtrees of even a moderate size tree. The proposed algorithm restricts the search to finding locally optimal recombinations of optimal trees obtained from the cost-complexity pruning algorithm. I denote the K-ary stratification of the optimally pruned subtree for parameter α by $S_K(T(\alpha))$. Note that some stratification of a sub-optimal tree may perform better.

Chou (1989) showed that a necessary condition for any K-ary partition $A_0, ..., A_{K-1}$, $A = \{t_1, ..., t_{|\tilde{T}|}\}$ to minimize the average impurity

$$I = \sum_{k=0}^{K-1} i(s_k) P(s_k)$$

is that $t \in A_k$ only if $k = \arg\min d(t, \mu(s_k))$ or if $P(t) = 0$, where $s_k = \{t \in A_k\}$ and where d is the divergence

$$d(t, \hat{\mu}) = E[L(Y, \hat{\mu}|t)] - E[L(Y, \mu(t)|t)],$$

which measures the increase in expected loss when $\hat{\mu}$ is used to represent Y instead of $\mu(t)$. This in general takes the problem of finding a optimal K-ary partition to polynomial time in N, $O(N^K)$. However, even with a moderate number of terminal nodes a fast approximate algorithm is useful. A K-means like clustering will be used as Chou (1989); the algorithm will always converge, but may only lead to a local optimum.

Below is an outline of the K-means algorithm that is applied to each optimally pruned subtree, T_l, subscripts indicating validation sample, and pruned subtree are omitted to simplify the description.

1. Pick some initial partition $S^0 = (A_0^0, ..., A_{K-1}^0)$. For example order the "mean"s for each node and dividing them up into K groups of approximately equal size.

2. Calculate the centroids, $\mu_0, ..., \mu_{K-1}$ where

$$\mu_k^j = \arg\min_{\mu_k} \sum_{t \in A_k^{j-1}} d(t, \mu_k)$$

3. Update the partition, $S^{j-1} = (A_0^{j-1}, ..., A_{K-1}^{j-1})$ let $t \in A_k^j$ if

$$k = \arg\min d(t, \mu_k^j)$$

Steps 2 and 3 are repeated until convergence of the partition. The algorithm yields a sequence of locally optimal K-ary stratifications $S_K(T(\alpha))$ for complexity parameter $\alpha_l \leq \alpha < \alpha_{+1}$, $l \geq 1$.

4 Examples

In this section I explore the stratification option and compare it to unstratified tree-based models on two simulated data sets. In both cases 10-fold cross-validation was used to calculate estimates of prediction error. The algorithms were implemented by modifying the tree-based tools of Clark and Pregibon (1991) in the S-programming Language.

4.1 Simple Regression

The regression function $f(\mathbf{x}) = 2 + 2x_1 + 2x_2 + 2x_3$ was used for this example. The response values were generated as

$$y_i = f(\mathbf{x}_i) + \epsilon_i$$

where the (x_1, x_2, x_3) were generated from the Uniform $(0, 1)$ distribution and ϵ_i were generated from a standard normal distribution. The sample size was $N = 250$. Figure 2 summarizes the estimated relative prediction errors for the unstratified tree and 2,3,4 and 5 group stratifications for each optimally pruned subtree. Relative prediction is the ratio of the prediction error to the null model prediction error. While it is clear in this situation that the 2, 3 and 4 group stratifications yield increased prediction error the 5 group stratification yields estimated prediction errors almost as small as the unstratified regression. However, in this example the stratification does not yield much simplification since the unstratified tree that minimizes the cross-validated estimate of prediction error has 7 terminal nodes.

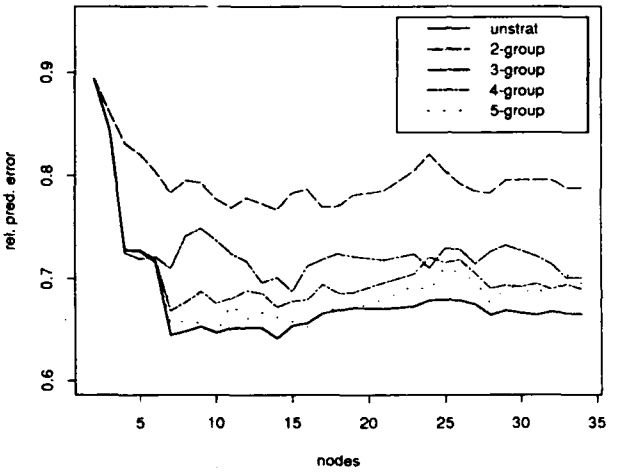


Figure 1: Example 1 - Cross-validated estimates of relative prediction error for the unstratified tree and 2,3,4 and 5 group stratifications.

4.2 Binary Response

Five hundred observations were generated from the model

$$\text{Binomial}(n = 1, p(\mathbf{x}_i) = \exp(f(\mathbf{x}_i))/(1 + \exp(f(\mathbf{x}_i)))$$

where th function

$$f(\mathbf{x}) = -2\text{sign}\{x_1 > .5\} + 2\text{sign}\{x_2 > .5\} - 2\text{sign}\{x_3 > .5\} + 2\text{sign}\{x_4 > .5.\}$$

The (x_1, x_2, x_3, x_4) were generated from the Uniform $(0, 1)$ distribution. In this model the conditional probability of success given $\mathbf{x}$ has only five different values. However, the the best fitting un stratified tree-based model has 13 nodes (with sufficient data one would expect a tree with 16 nodes, each node corresponding to

an area of constant success probability). Figure 2 shows that either the 3 or 4 group stratifications at tree sizes of 12 and 13 nodes perform similarly (slightly smaller estimated prediction error) for the unstratified tree with 12 terminal nodes. The 3-group stratification is presented in Figure 3. Note, Figure 2 also shows that for trees much larger than the optimal size the 2 and 3 group stratification perform better than the full unstratified trees.

The technique has also been applied data on prognosis after heart attacks. The data set analyzed included 1780 subjects collected by the Specialized Center for Research on Ischemic Heart Disease at the University of California, San Diego. A subset of this data set was analyzed in BFOS. The unstratified tree that minimizes the cross-validated estimate of weighted deviance had eight terminal nodes with a relative deviance of .883. However, the stratified tree with only 3 prognostic strata has similar relative deviance of .889. The analysis is presented in LeBlanc (1991, unpublished manuscript).

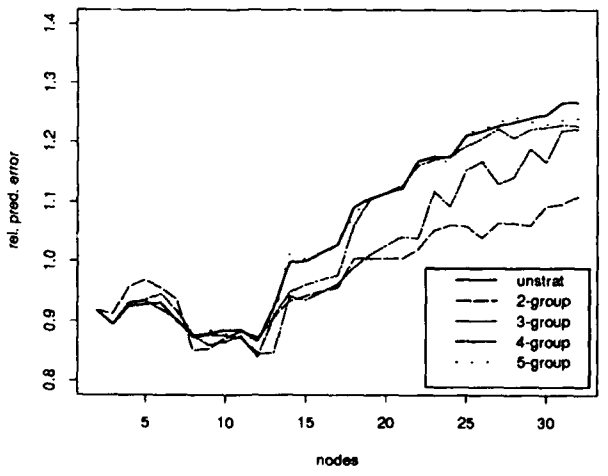


Figure 2: Example 2 - Cross-validated estimates of relative deviance for the unstratified tree and 2, 3, 4, 5 group stratifications.

The proposed procedure uses the K-means algorithm for all pruned subtrees; another possibility would be to use a combined approach. For small trees the optimal K-ary stratification could be found by the optimal partition theorem and for larger trees a locally optimal stratification could be found by the K-means algorithm.

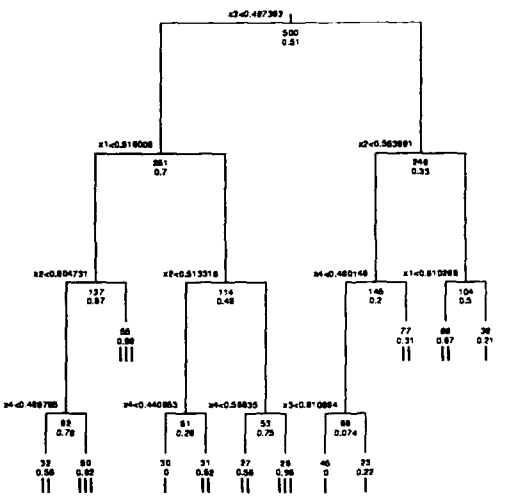


Figure3: Example 3 - 3-ary stratification tree. The number of observations and success probability for the unstratified tree are given below each node.

References

[1] Breiman, L., Friedman, J., Olshen, R., and Stone, C. *Classification and Regression Trees.* Wadsworth International Group, 1984.

[2] Chou, P.A. Optimal partitioning for classification and regression trees. Technical report, ATT Bell Laboratories, 1989.

[3] Ciampi, A., Change, C.H., and McKinney, S.M. Recursive Partition: a versatile method for exploratory data analysis in biostatistics, in *Biostatistics.* D. Reidel Publishing Company, 1987.

[4] Ciampi, A., Hogg, S., McKinney S., and Thiffault, J. RECPAM: a computer program for recursive partition and amalgamation for censored survival data. *Computer Methods and Programs in Biomedicine*, 26:239-256, 1988.

[5] Clark L. and Pregibon D. Tree-Based models, in *Statistical Models in S.* Wadsworth International Group, 1991.

[6] Morgan, J. and Sonquist, J. Problems in the analysis of survey data and a proposal. *Journal of the American Statistical Association*, 58:415-434, 1963.

The Relevance Density Method for Multi-topic Queries in Information Retrieval.

Y. Kane-Esrig, L. Streeter, S. Dumais, W. Keese
Bell Communications Research
331 Newman Springs Road, Red Bank, NJ 07701-7030

G. Casella
Cornell University, 337 Warren Hall, Ithaca, NY 14850

Abstract

A long standing problem in information retrieval is how to treat queries that are best answered by two or more distinct sets of documents. Existing methods average across the words or terms in a user's query, and consequently, perform poorly with multimodal queries, such as: "Show me documents about French art and American jazz". We propose a new method, the *Relevance Density Method* for selecting documents relevant to a user's query. The method can be used whenever the documents and the terms are represented by vectors in a multi-dimensional space, such that the vectors corresponding to documents and terms dealing with closely related topics are close to each other. We show that the Relevance Density Method performs better for multimodal as well as single mode queries than an averaging method. In addition, we show that retrieval is substantially faster for the new method.

Introduction

The task of an information retrieval system is to respond to a user's request for information (*a query*) by searching a collection of *documents* (e.g texts such as books, journal articles etc.) and selecting those documents that seem to be relevant to the topic(s) of the query. Usually, the documents in the collection are indexed by *terms* (keywords). It is assumed that the topic(s) of a document or of a query is adequately reflected by its collection of terms.

The relevance density method proposed in this paper can be applied whenever terms and documents are represented by vectors in the same multidimensional *document-term space* with similarity of terms and documents reflected by the closeness of their vector representations in that space. In other words, if two vectors are close together, then the corresponding terms or documents can be assumed to be closely related in their topics and vice versa. Methods for constructing such a space are presented in [1] and [2].

Currently, the method of selecting relevant documents used in conjunction with such vector representations of terms and documents is called *vector averaging* (VA). VA [2], [3] represents a query by a single vector in the document-term space. This query vector is a weighted average of the term vectors used in the query. Documents in the collection are ranked by the closeness (measured by the cosine or dot product) of their vectors to the query vector. The top ranking documents are selected as relevant and returned to the user.

Representing the query by a single vector works well when the vectors of the relevant objects (documents and terms) are clustered together in a single region of the document-term space, since the center of that region is a reasonable estimate of the query's content. However, if the vectors of the relevant objects fall into two or more clusters separated by regions of the space containing non-relevant documents, then averaging will perform poorly, since it will tend to retrieve documents between the two clusters of relevant documents. One proposed solution [4] was to identify multimodal queries and split them into sub-queries. However, this method was too computationally expensive and has not been used widely.

An additional drawback of vector averaging is computational expense. Typically, the query vector is compared to *every* document vector. If the document collection is large and the dimensionality of the document-term space high, computational demands can be quite significant. The proposed method can be implemented using table look-up, thereby trading space for time.

The Relevance Density Method

We propose a new method of ranking documents. The Relevance Density Method (RDM) can be used whenever documents and terms are represented by vectors in the document-term space.

We treat relevance as a continuous quantity and model its distribution by a probability density $\pi(\boldsymbol{D})$ over the document-term space. The documents in the collection are ranked in the order of the height of the density over their vector representations $\boldsymbol{D}$. In other words, the document that has the highest value of $\pi(\boldsymbol{D})$ is given rank 1 etc., with higher ranks reflecting greater similarity or relevance. Thus, this density should be high over areas of the

document-term space containing vectors to relevant objects and low over areas of nonrelevant objects. If there is more than one cluster of relevant objects, then the density should be multimodal.

To construct the density $\pi(\boldsymbol{D})$ we will start with a prior density $\pi_0(\boldsymbol{D})$ which reflects the system's a priori guess about the user's interests. If no prior information about the user is available, $\pi_0(\boldsymbol{D})$ is a constant and does not affect the ranking. We use Bayes' rule to update the density when the user's query is received. As in vector averaging, the query is treated as a collection of terms used in the query. Let $Q = \{\boldsymbol{T}^1, \ldots, \boldsymbol{T}^k\}$ be the set of vectors corresponding to the terms used in the query, where k is the number of terms used in the query. Then[1]

$$\pi_1(\boldsymbol{D} \mid Q) = f(Q \mid \boldsymbol{D}) \cdot \pi_0(\boldsymbol{D})$$

In some cases, *relevance feedback* can be obtained from the user after the initial query. The user is presented with a few top ranking documents and asked which of them s/he considers relevant to her/his query. If such relevance feedback is available, it can be used to update $\pi(\boldsymbol{D})$. Let $Q_1 = \{\boldsymbol{D}^1, \ldots, \boldsymbol{D}^m\}$ be the set of vectors corresponding to the documents that the user considered relevant, where m is the number of documents the user considered relevant. Then the relevance density after the feedback is:

$$\pi_2(\boldsymbol{D} \mid Q, Q_1) = f(Q_1 \mid \boldsymbol{D}) \cdot \pi_1(\boldsymbol{D} \mid Q)$$

We used:

$$f(Q \mid \boldsymbol{D}) = \sum_{j=1}^{k} w_j \cdot c(b_j) \cdot \exp\left[b_j \cdot \cos(\boldsymbol{T}^j, \boldsymbol{D})\right]$$

where $\cos(\boldsymbol{T}, \boldsymbol{D})$ is the cosine of the angle between the term vector $\boldsymbol{T}$ and document vector $\boldsymbol{D}$. The above density has the property of being unimodal when the term vectors are in a single cluster and multimodal when there is more than one cluster. This density is a sum of bell-shaped components. The ith bell is centered over the vector of the ith term used in the query. The bell is tall and narrow if the *parameter of concentration*, b_j is high, and low and wide, if b_j is low. The parameter of concentration differentiates highly specific terms from broad, less specific terms. For example, single word terms, such as *cable* tend to be less specific than multi-word terms, such as *fiberoptic cable* [3] [6]. The factor $c(b_j)$ normalizes the ith bell to integrate to 1, making it a proper density. The weights w_j can be used to express different amounts of importance associated with terms. For example, words can be weighted according to their information value; common or frequent words are weighted less heavily than rare words. (A list of desirable qualities of a sampling density, proofs that $f(Q \mid \boldsymbol{D})$ has these qualities, and an alternative sampling function are presented in [5].)

The values of $w_j \cdot c(b_j) \cdot \exp\left[b_j \cdot \cos(\boldsymbol{T}_j, \boldsymbol{D})\right]$ can be pre-calculated for every document and term and stored. Thus, when a user's query is processed, the system simply looks up the values corresponding to the terms used in the query and adds them up to compute $f(Q \mid \boldsymbol{D})$. This table look-up method of computation makes the RDM far less computationally expensive than the VA in terms of the number of operations required [5]. However, if the term by document matrix is large, having enough space to store the values becomes an issue.

Results of Testing

Both the RDM and VA methods were tested on Bellcore's ADVISOR system [3], [6]. The system responds to a query by identifying departments within Bellcore best suited to answer the query. (Bellcore is a large and diverse research and development company.) At the time of the first set of tests, the 104 departments were represented by abstracts of the technical papers they produced in 1987. There were 728 such documents indexed by 7,100 terms in the ADVISOR's collection. New abstracts were collected in 1987 and in 1989 and used as test queries. (We did not use as queries any of the abstracts in ADVISOR's collection.) In addition, to study the performance in cases where the query was likely to have at least two separate topics, we constructed "double" queries by joining the texts of pairs of abstracts produced by two different departments and treating these joined texts as a single query.

The measure of performance for each test query was the rank of the first retrieved "relevant" document. A document was considered relevant to the query if it was produced by the same department as the one that produced the query. In the case of the double queries, the documents produced by either one of the two departments were considered relevant. If the method of retrieval were perfect, the rank of the first correct document would be 1. On the other hand, if the the documents were ranked randomly, the rank would be on average 52.

Each query was ranked by each of the two methods, RDM and VA. VA was used with a root mean squared weighting of the terms and with the cosine as the similarity measure. This weighting scheme and similarity measure were chosen because they produced the best performance in previous tests on this collection. The RDM was used with a constant prior density (i.e. no prior information), with constant weights on the terms and with b_j=1 for

1. To make π_1 a proper density, a scaling constant is needed, but since it does not affect the ranking, we will omit it.

terms consisting of a single word and b_j=2 for multi-word terms.

The results of these tests are presented in Table 1. We observe that for 263 new abstracts produced in 1987, which were used as queries, both VA and RDM answered at least 25% of these 263 queries correctly on the first try, since the lower quartile of the ranks of the first correct documents is 1 for both methods. VA answered at least 50% of the queries on or before the third try (median rank=3), while RDM did better with a median rank of 2. Finally, the upper quartile of the ranks was 19 for VA and 9 for RDM. This indicates that RDM answered 75% of the queries correctly on or before the 9th try, whereas VA answered 75% of the queries correctly on or before the 19th try. From the user's point of view there is likely to be a big difference between looking at 8 versus 18 non-relevant documents before getting a relevant one. The statistical significance of the differences in performance was assessed using a Wilcoxon Signed Rank test. The value of the z statistic for the 263 queries was -2.14. The p value of the test against the two-sided hypothesis is 0.016.

Similar comparisons can be made for the two ranking methods based on queries from 1989 and on the "double" queries from 1987 and 1989. Both methods performed better on the 1987 queries. This is to be expected, since the work of the departments represented in ADVISOR's database is from 1987 documents, and undoubtedly departments' emphasis and work have shifted in two years.

The overall conclusion that can be drawn from the data in Table 1 is that the RDM performed better than the VA (had the rank of the first relevant document closer to 1). The Wilcoxon test statistic ranged from highly significant (p value < 0.0001) to moderately significant (p value < 0.018), but in all 4 tests the RDM was the superior method.

Recently, we compared the two methods in terms of their computational cost. We collected 316 actual queries submitted by the users at Bellcore to ADVISOR and found out how long it took to do the computations needed by each method for these queries. (We ignored the time it takes to do the I/O and the sort of the documents since this is the same for both methods.) The current version of ADVISOR represents documents and terms by 300 dimensional vectors and has 1023 documents in its collection. The computations were done on a DEC 5000/200 machine. The computation time (the sum of user and system time) is plotted against the number of terms in the query in Figure 1. It is obvious, that VA took substantially longer than RDM. The median of the VA time was 0.53 seconds, of the RDM time was 0.02 seconds.

Conclusions

The Relevance Density Method of ranking documents for retrieval was designed to overcome two problems of the currently used method, Vector Averaging. These problems are: (1) poor performance in the case of multimodal queries and (2) high computational cost. The proposed method was tested on Bellcore's ADVISOR system and performed faster and better than Vector Averaging in these tests.

References

[1] Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A. "Indexing by latent semantic analysis." *Journal of the Society for Information Science*, 1990, *41(6)*, 391-407.

[2] Salton G. and McGill M. J. *Introduction to Modern Information Retrieval.* McGraw Hill, New York 1983.

[3] Streeter L. A. and Lochbaum K. E. "An Expert/Expert-locating System Based on Automatic Representation of Semantic Structure." *Proceedings of the Fourth Conference on Artificial Intelligence Applications.* San Diego Ca., March 14-18, 1988, pp. 345-350.

[4] Borodin A., Kerr L. and Lewis F. "Query Splitting in Relevance Feedback Systems." *Scientific Report No. ISR-14,* Department of Computer Science, Cornell University, Ithaca NY, October 1988.

[5] Kane-Esrig Y. *Information Retrieval and Estimation with Auxiliary Information.* Dissertation, field of Statistics, Cornell University, Ithaca NY, 1990.

[6] Streeter L. A. and Lochbaum K. E. "Who knows: A System based on Automatic representation of Schematic Structure." *RIAO 88: User-oriented Content Based Text and Image Handling* Massachusetts Institute of Technology, Cambridge MA, March 21-24, 1988, pp 379-388.

TABLE 1
ADVISOR RESULTS

1987 Queries

Method	Lower Q	Median	Upper Q
VA	1	3	19
RDM	1	2	9

Z_{wilc}	p value	# of queries
-2.14	0.016	263

1987 PAIRS

Method	Lower Q	Median	Upper Q
VA	1	5	24
RDM	1	3	19

Z_{wilc}	p value	# of queries
-4.20	0.000	66

1989 QUERIES

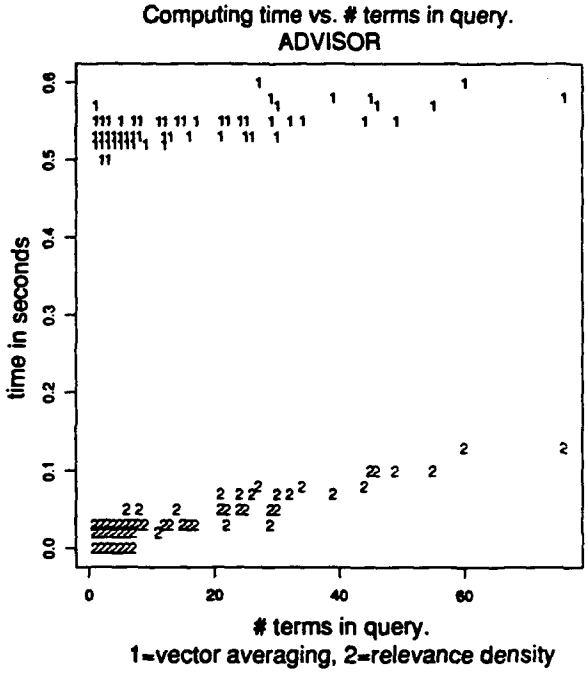
Method	Lower Q	Median	Upper Q
VA	2	8	51
RDM	1	5	29

Z_{wilc}	p value	# of queries
-0.920	0.018	43

1989 PAIRS

Method	Lower Q	Median	Upper Q
VA	3	10	31
RDM	1	6	33

Z_{wilc}	p value	# of queries
-0.918	0.018	98



Analysis of Data from Computer Linked Files

William E. Winkler, Bureau of the Census, Rm 3000-4, Washington, DC 20233

1. INTRODUCTION

Information that resides in two computer data bases can be useful for analysis and policy decisions. For instance, an epidemiologist might wish to evaluate the effect of a new cancer treatment by matching information from a collection of medical case studies against a death index that contains information about the cause and date of death. An economist might wish to evaluate energy policy decisions by matching a data base containing fuel and commodity information for a set of companies against a data base containing the values and types of goods produced by the companies. If unique identifiers such as verified social security numbers or employer identification numbers are available, then matching data sources is straightforward and standard methods of statistical analysis are applicable.

If such identifiers are not available, then matching must be performed using information such as company or individual name, address, age, and other descriptive information. Even when typographical variation and errors are absent, name information such as 'Smith' and 'Robert' may not uniquely identify an individual. Use of address information is often subject to error because parsing/standardization software do not effectively allow comparison of, say, a house number with a house number and a street name with a street name. The addresses of an individual we wish to match may differ because one is erroneous or because the individual has moved.

Fellegi and Sunter (1969) presented a formal mathematical model and showed the optimality of decision rules in a record linkage strategy. Pairs of records in a file are given a score. Those above a certain score are designated matches, those below a second, lower, score are designated nonmatches, and those with with scores between the higher and lower scores are held for clerical review. The scores, or computer matching weights, are based on a crucial likelihood ratio that is often difficult to estimate (see e.g., Winkler and Thibaudeau 1990, Belin and Rubin 1990, 1991).

With files of moderate size, several thousand pairs may need to be clerically reviewed. As such review often involves examining paper forms (if they exist) or use of additional data sources, it is expensive and subject to error. With large files, reviewing hundreds of thousands of pairs is likely to be prohibitively expensive.

Winkler and Scheuren (1991) introduced a model that provides a means of adjusting general regression analyses for matching error. The main purpose of the adjustment procedure is to reduce or eliminate the need for clerical review. At a minimum, the procedures tell us how much accuracy is improved via adjustment, whether estimates are sufficiently accurate for statistical analyses and policy decisions, and how much cost must be incurred (through targetted clerical review) to insure a given benefit in increased accuracy. The key to the adjustment procedure is estimating accurately the proportions of matches and nonmatches within a set of pairs for all ranges of scores. The method of estimating proportions of matches within weight ranges is due to Belin and Rubin (1990, 1991).

The paper presents an evaluation of the adjustment procedure for ordinary linear regression. The evaluation tool is an extension of Rubin's multiple imputation (see e.g., Rubin 1987, pp. 75-77). The empirical data base is constructed from two files for which true matching status of pairs is known. Very extensive review and verification of pairs was done to assure that matching status is accurate. Numerical data are constructed using known normal models. Different sets of seed numbers produce different samples.

The intuitive idea of multiple imputation is that the structure of data relationships and the model under which we impute places restraints on the statistical estimates being considered. For nonresponse (Rubin 1987), the set of data values associated with respondents, the pattern of nonresponse, and the imputation model all effect multiply imputed parameter estimates and their variances. For this paper, what records from one file are matched with what records from another file, the data associated with the matched records, and the model for adjusting for matching error all effect the multiply imputed estimates.

The outline for the remainder of the presentation is as follows. In the second section we present some of the theoretical background. In the third section we present brief results. The final section consists of discussion.

2. BACKGROUND

2.1 Theoretical Adjustment Model

This section provides a description of the regression framework and adjustment methodology for the simplest classes of univariate regression. The theory for general regression is given by Winkler and Scheuren (1991).

Let $Y = X + \epsilon$ be the ordinary univariate regression model for which error terms are independent with constant variance σ^2. If we were working with a single data base, Y would be regressed on X in the usual manner. For $i = 1, \ldots, N$, we wish to use (X_i, Y_i) but use (X_i, Z_i). Z_i is usually Y_i but may take some other value Y_j due to matching error.

For $i = 1, \cdots, N$,

$$Z_i = \begin{cases} Y_i & \text{with probability } p_i \\ Y_j & \text{with probability } q_{ij} \text{ for } j \neq i. \end{cases}$$

$$p_i + \sum_{j \neq i} q_{ij} = 1.$$

The probability p_i may be zero or one. We define $h_i = 1 - p_i$ and divide the set of pairs into N mutually exclusive classes. The classes are determined by records from one of the files. Each class consists of the independent x-variable X_i, the true value of the dependent y-variable, the values of the y-variables from records in the second file to which the record in the first file containing X_i have been paired, and computer matching weights. Some of the N classes may have zero matching weights By paired we mean two records from the two files that have been brought together during the record linkage process but for which no determination of matching status may have been made. Under an assumption of 1-1 matching, for each $i = 1, \cdots, N$, there exists at most one j such that $q_{ij} > 0$. We let ϕ be defined by $\phi(i) = j$.

To define regression properly, we need to find $\mu_z = E(z)$, σ_z^2, and σ_{zx}. We observe that

$$E(Z) = (1/N) \sum_i E(Z|i) = (1/N) \sum_i (Y_i p_i + \sum_{j \neq i} Y_j q_{ij})$$

$$= (1/N) \sum_i Y_i + (1/N) \sum_i [Y_i (-h_i) + Y_{\phi(i)} h_i] = \bar{Y} + B.$$

Similarly, we can represent σ_{zy} in terms of σ_{xy} and a bias term B_{xy} and σ_z^2 in terms of σ_y^2 and a bias term B_{zz}. We neither assume that the bias terms have expectation zero nor that they are uncorrelated with the observed data.

Different equations yield the adjustments that relate regression coefficients β_{zx} based on observed data with regression coefficients β_{yx} based on true values. Our assumption of 1-1 matching (which is not needed for the general theory) is done for computational tractability to reduce the number of records and amount of information that must be tracked during the matching process.

In implementing the adjustments, we make two crucial assumptions. The first is that, for $i = 1, \cdots, N$, we can accurately estimate the true probabilities of a match p_i. The second is that, for each $i = 1, \cdots, N$, the true value Y_i associated with independent variable X_i is the pair with the highest matching weight and the false value $Y_{\phi(i)}$ is associated with the second highest matching weight.

2.2. Empirical Data Base

The empirical data base is created from two files of 10,000 records having known matching status. Basic matching parameters (see e.g., Winkler and Thibaudeau 1990) are estimated that cause the curves of log frequencies versus matching weight for nonmatches and matches to separate (Figure 1). Matching probabilities are estimated using the Belin-Rubin methodology (Table 1). We see that the estimated probabilities agree quite closely in the tails (above weight 4 and below weight 2). For weight 3, the deviation is relatively large because the true proportion of false matches is 0.06 while the estimated one is 0.20.

Table 1. Probabilities and Counts of Matches and Nonmatches in Weight Ranges

weight	- Count - Mat	NM	Probability true	est
11	6950	0	.00	.00
10	785	0	.00	.00
9	610	0	.00	.00
8	439	3	.00	.00
7	250	4	.00	.01
6	265	9	.03	.03
5	167	8	.05	.06
4	89	6	.06	.11
3	84	5	.06	.20
2	38	7	.16	.31
1	33	34	.51	.46
0	13	19	.59	.61
-1	7	20	.74	.74
-2	3	11	.79	.84
-3	4	19	.83	.89
-4	0	15	.99	.94
-5	0	15	.99	.96
-6	0	27	.99	.98
-7	0	107	.99	.99

1/ In the first column, weight 10 means weight range from 10 to 11. Weight ranges 11 and above and -7 and below are added together separately. Mat is match and NM is nonmatch.

Each unique record in the merged data files has an independent x-variable that is generated according to a uniform distribution between 1 and 101 and a dependent y-variable that is generated via with a random normal distribution such that the slope is 2 and the R-square value is approximately 0.45. Error arises because the observed (x,y)-pair that is normally used in computation has a y-value from a record to which the record containing the x-value was falsely matched.

For the analysis we consider only those pairs having matching weights between 0 and 10 because all pairs above

weight 10 are true matches. Pairs between 0 and 10 contain both true and false matches. We do this to determine how much the adjustment improves the accuracy of the regression analyses in situations for which there is significant matching error. If we include pairs above weight 10, then then it is more difficult to judge the adjustment process because ordinary regression estimates based on observed data and adjusted regression estimates will both be relatively more accurate.

In the remainder of the paper, whenever we use true, we will mean estimates based on the true values. Similarly, when we use observed, we mean estimates based on observed data. Adjusted will always refer to estimates obtained via the adjustment methods of this paper.

3. RESULTS

The results of using the adjustment process are illustrated in Figure 2. Figure 2a provides a comparison of the relative coefficients of variation of the adjusted procedure versus the nonadjusted procedure. To get the plotted points, the coefficients of variations (cvs) computed via either procedure are divided by the true cv for weight class 8. The results show that both adjusted and nonadjusted procedures yield approximately the same cv estimates and that cvs decrease as sample size increases. The relative bias of the cvs for the adjusted procedure is substantially lower than the relative bias for the nonadjusted procedure (Figure 2b). The nonadjusted procedure uses ordinary linear regression on the observed data pairs.

Multiply imputed estimates for 25 samples (Table 2) show the relative cv estimates for both adjusted and nonadjusted procedures are about the same while the higher bias of the nonadjusted procedure yields higher quasi root mean square errors (qmrse). The term qrmse is used because we use an estimate of the variance component of root mean square error rather than the true value. We observe that for higher weight ranges, say between 6 and 10, both the adjusted procedure and nonadjusted procedure produce about the same qmrses, 0.056 and 0.058, resp. As weight ranges having more erroneous data are included, say between 0 and 10, qrmse under the adjusted procedure, 0.048, is substantially lower than under the nonadjusted procedure, 0.081.

4. DISCUSSION

The multiple imputation procedure adopted for analyzing the adjustment procedure was intended to dampen the influence of the regression-variable-creation procedure. Specifically, as individual samples showed significant variation from sample to sample, it was difficult to determine how much of an improvement the adjustment procedure yielded. Although not shown, the between sample component of the variance estimated via the multiple imputation procedure was roughly equal the within sample component. If we had considered only individual samples, we would have missed the additional source of variation.

Table 2. Comparison of Estimates Averaged over 25 Samples Coefficient Estimates

wgt class	size	true	est	obs
8	442	2.020	2.018	2.004
cv		0.082	0.082	0.082
qrmse			0.082	0.082
6	970	2.015	2.002	1.976
cv		0.053	0.056	0.056
qrmse			0.056	0.058
4	1240	2.010	2.006	1.956
cv		0.046	0.048	0.049
qrmse			0.048	0.055
2	1374	2.005	2.025	1.940
cv		0.044	0.047	0.047
qrmse			0.049	0.056
0	1473	2.007	1.976	1.870
cv		0.042	0.046	0.046
qrmse			0.048	0.081

Note: Weight class 2 means those pairs having weight above 2 and below 9.

This paper reflects views of the author and not necessarily those of the Census Bureau.

REFERENCES

Belin, T. and Rubin, D. (1990) "Calibration of Errors in Computer Matching for Census Undercount," Proc. of the ASA Section on Government Statistics, to appear.

Belin, T. and Rubin, D. (1991) "Recent Developments in Calibrating Error Rates for Computer Matching," 1991 Census Bureau Annual Research Conference, to appear.

Fellegi, I. and Sunter, A. (1969) "A Theory of Record Linkage," J. Amer. Stat. Assn. 1183-1210.

Rubin, D. B. (1987) Multiple Imputation for Nonresponse in Surveys, New York: J. Wiley.

Winkler, W. E. and Scheuren, F. (1991) "An Error Model for Regression Analysis of Data Files that arc Computer Matched," 1991 Census Bureau Annual Research Conf.

Winkler, W. E. and Thibaudeau, Y. (1990) "An Application of the Fellegi-Sunter Model of Record Linkage to the 1990 U.S. Census," Technical report.

Figure 1. Log of Frequency vs Weight Matches & Nonmatches

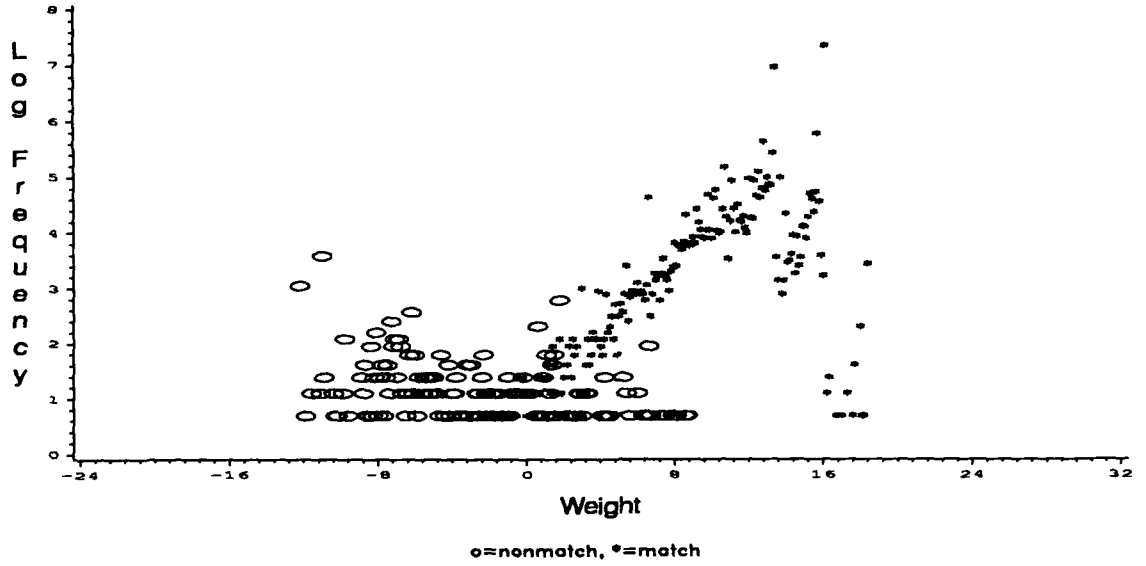


Figure 2a. Relative Coeff Var vs Weight, Estimated Probabilities R-square=0.45

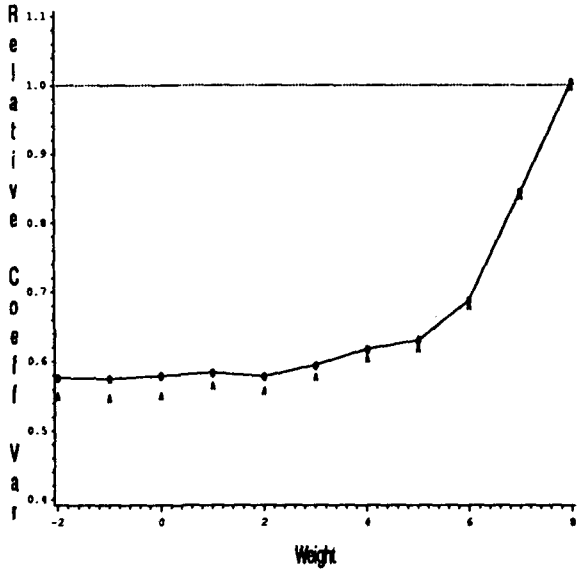
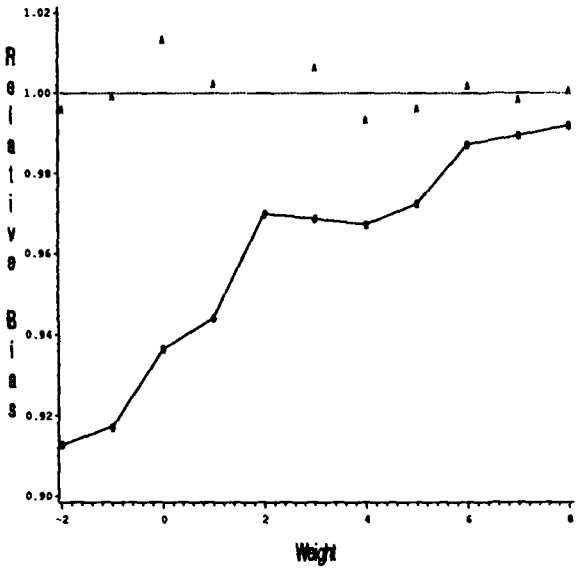


Figure 2b. Relative Bias vs Weight, Estimated Probabilities R-square=0.45



The Discrimination Power of Dependency Structures in Record Linkage

Yves Thibaudeau, U.S. Census Bureau, F.B.4, Room 3000, Washington, DC 20233.

Abstract

In record linkage, the correct statistical model underlying a particular application may present estimation difficulties. Often, a convenient model is substituted in place of the correct one. Naturally, the substitution induces an error and one can only hope that the error is negligible. This paper compares two models as they are applied to the data collected during the 1988 Dress Rehearsal of the Decenial Census.

Key Words: Record linkage; Decision rule; Conditional independence model;

1 Introduction

The paper compare three related techniques of record linkage. Sections 2 to 5 give a background on record linkage, while sections 6 to 8 apply the 3 techniques to a particular situation.

2 Record-Linkage Rules

Consider two populations of individuals: population A and population B. Denote the individuals of these two populations by a and b respectively. A and B may have some individuals in common. Consider the set of all possible ordered pairs (a,b). This set is the cartesian product $A\times B = \{(a,b) \mid a \in A, b \in B\}$ and it can be divided into two sets: $M=\{(a,b) \mid a \in A, b \in B, a=b\}$ and $U=\{(a,b) \mid a \in A, b \in B, a\neq b\}$. The pairs in M are the links, while the pairs in U are the non-links. Note that $M \cap U = \varnothing$ and $M \cup U = A \times B$. Let α be a record generating function on A and let β be a record generating function on B. These two functions produce $\alpha(a)$ and $\beta(b)$, the records of a and b respectively. γ is a comparison function over $\alpha(A) \times \beta(B)$ if for any individual $a \in A$ and for any individual $b \in B$, the record $\alpha(a)$ can be compared to the record $\beta(b)$ through the comparison function $\gamma(\alpha(a),\beta(b))$. Finally, the comparison space is the set $\Gamma = \{\gamma(\alpha(a),\beta(b)) | a \in A, b \in B\}$, the set of all possible comparison values.

In practice, the comparison function is a vector valued function. Each vector component $\gamma^i(\alpha(a),\beta(b))$, where $i=1,...,N$, corresponds to a specified field, such as last name or age. $\gamma^i(\alpha(a),\beta(b)))$ is assigned the value 0 if the records of the two indviduals disagree over field i and it is assigned 1 if they agree. The comparison space Γ is the set of all binary vectors (i.e. whose components are 0 or 1) of dimension N.

Consider a particular comparison vector denoted by γ^*. The probability that a pair of records (a,b) gives rise to γ^*, through the comparison function γ and given that the pair belongs to the set of links M, is defined as follows:

$$m(\gamma^*)= \sum_{(a,b) \in M} Pr[\gamma(\alpha(a),\beta(b)) = \gamma^* | (a,b)] Pr[(a,b) | M]$$

Similarly,

$$u(\gamma^*)= \sum_{(a,b) \in U} Pr[\gamma(\alpha(a),\beta(b)) = \gamma^* | (a,b)] Pr[(a,b) | U]$$

is the probability that a pair of records gives rise to γ^* given that the pair is a non-link.

The purpose of record-linkage is to determine which pairs are the links. In this respect, a decision rule is constructed. Let A_1 be the decision to declare a given pair a link, while A_2 is the decision to declare that same pair a possible link, and A_3 is the decision to declare the pair a non-link. Any one of these three decision is taken on the basis of γ. It is assumed that the comparison vector γ is sufficient. In this context a decision function is a triplet of probabilities, $d(\gamma) = (Pr[A_1 | \gamma], Pr[A_2 | \gamma], Pr[A_3 | \gamma])$, where $Pr[A_i|\gamma]$ is the probability of making decision A_i when observing comparison vector γ_3, where $i=1,...,3$. Naturally $Pr[A_i|\gamma] \geq 0$ and $\sum Pr[A_i|\gamma] = 1$. The definition of record linkage rule follows easily: A record linkage rule (linkage rule) is a mapping from the comparison space Γ onto a set of decision function $D = \{d(\gamma)\}$.

Two types of error may occur when applying a linkage rule. The type I error occurs whenever a pair declared a non-link is in fact a link. The type II error occurs when a pair is declared a link but is not. A linkage rule is said to be a linkage rule at the levels μ and λ, where $0<\mu<1$ and $0<\lambda<1$, if $Pr[A_1|U] = \mu$ and $Pr[A_3|M] = \lambda$. Here $Pr[A_1|U]$ is the Type II error and $Pr[A_3|M]$ is the type I error. Such a linkage rule is denoted by $L(\mu,\lambda,\Gamma)$. Furthermore, the rule $L(\mu,\lambda,\Gamma)$ is said to be optimal at the levels μ and λ if for any other linkage rule at the levels μ and λ, denoted by $L^*(\mu,\lambda,\Gamma)$, the following holds: $Pr[A_2|L] \leq Pr[A_2|L^*]$. That is the probability of declaring any pair a possible link is no greater under rule L than under rule L^*, while maintaining the same error levels.

3 The Fellegi-Sunter Theorem

Fellegi and Sunter (1969) formally show how to construct an optimal linkage rule. Let all the comparison vectors γ be ordered by decreasing order of the ratio $m(\gamma)/u(\gamma)$. If there are ties, order is assigned randomly among them. This ordering gives rise to a sequence $\{\gamma_i\}_{i=1}^{N_\Gamma}$, where N_Γ is the total number of comparison vectors. For given error levels μ and λ assume there exists n and n^* such that

$$\sum_{i=1}^{n-1} u(\gamma_i) < \mu \leq \sum_{i=1}^{n} u(\gamma_i)$$

and

$$\sum_{i=n^*}^{N_\Gamma} m(\gamma_i) \geq \lambda > \sum_{i=n^*+1}^{N_\Gamma} m(\gamma_i)$$

Consider the following linkage rule:

P_μ must satisfy $u(\gamma_n)P_\mu = \mu - \sum_{i=1}^{n-1} u(\gamma_i)$. This ensures the consistence of the randomisation rule. A similar constraint involves P_λ.

THEOREM 1 (Fellegi and Sunter, 1969): The linkage rule

$$d(\gamma_i) = \begin{cases} (1,0,0) & i \leq n-1 \\ (P_\mu, 1-P_\mu, 0) & i = n \\ (0,1,0) & n<i\leq n^*-1 \\ (0,1-P_\lambda,P_\lambda) & i=n^* \\ (0,0,1) & i \geq n^*+1 \end{cases} \quad (1)$$

defined in (1) is optimal at the levels μ, λ.

In order to make use of theorem 1, the ratio $m(\gamma)/u(\gamma)$ must be known for each observable value of the comparison vector γ. Of course, in practice, those ratios are unknown and must be estimated. To perform the estimation, a class of probabilistic models is established. Then estimation techniques are used. Before introducing some classes of models, more notation must be reviewed.

4 Notation for record-linkage

Let $\nu_{k,i_1,\ldots,i_N}$ represents the count of pairs with the following attributes: whenever $k = 0$ the corresponding pairs are non-links and whenever $k = 1$ they are. Furthermore, when $i_s = 0$, the corresponding pairs do not exhibit record agreement over comparison field s and whenever $i_s = 1$, the pairs do exhibit record agreement over comparison field s. Note that $s = 1,\ldots,N$, where N is the number of comparison fields.

It is important to realize that the counts $\nu_{k,i_1,\ldots,i_N}$ cannot be observed. Rather, what is observed are the aggregated counts, denoted by $\nu_{i_1,\ldots,i_N}$, where

$$\nu_{i_1,\ldots,i_N} = \nu_{0,i_1,\ldots,i_N} + \nu_{1,i_1,\ldots,i_N}$$

This notation is usefull. The next section present a class of record linkage models.

5 The Conditional Independence Model

Goodman (1974) gives a thorough analysis of the conditional independence model. It is best described by its log-linear representation:

$$\log(v_{k,i_1,\ldots,i_N}) = \mu + \lambda_k + \sum_{j=1}^{N} \alpha_{i_j}^{j} + \sum_{j=1}^{N} \zeta_{k,i_j}^{j} \qquad (2)$$

There are constraints on the parameters involved on the right hand side of (2). These can easily be deduced and are left out here.

The expression on the right-hand side of (2) includes one term corresponding to the effect of the link status (link/non-link) of the counted pairs (λ_k) and one term for the effect of each comparison field ($\alpha_{i_j}^{j}$). It also includes terms for the interaction effects between the link status and the fields(ζ_{k,i_j}^{j}). However, there are no interaction terms between the fields and this implies that, conditional on the latent class, the fields are independent. In many cases, because of dependency relationships, it is necessary to include interactions terms between certain fields. In those case, selective models must be used. The following situation is such a case.

6 Applications: The St. Louis data

These data were collected in 1988 during a dress rehearsal, in preparation for the Decennial Census operations. Two files were created, based on two surveys of the individuals living in a defined geographical area within the city of St. Louis. Those surveys are the census and the post-enumeration survey. In both cases, for each individual reported at the time of the survey, a record is created and various characteristics of the individual are recorded. The objective is to link the records of the Census file with the records of the Post Enumeration Survey. The comparison fields are indexed 1 to 11 and are in order: Surname, house no., street name, phone number, first name, middle initial, marital status, age, race, sex and relationship with the respondent.

7 Two Models for the St. Louis data

In the case of the St. Louis data there are dependencies between some fields. Particularly among the non-links, between the houseold fields. These are surname, street no., street name and telephone. When two individuals agree on some of the houseold fields, they are more likely to be living under the same roof and therefore they are more likely to agree on the rest of the houseold fields.

In this section, two explanatory models are proposed for the St. Louis data. The first model is the conditional independence model in (2) with $N = 11$. The second model includes interaction terms between the houseold variables. The log-linear representation of this model is similar to that in (2), but with the addition of 2-nd, 3-rd and 4-th order interaction terms between the household fields, among the non-links.

8 Linkage Performances

In this section, the models are fitted to the St-louis data. The Fellegi-Sunter rule is applied under the conditional independence model and under the model with interactions between the houseold fields.

In section 2, the Type II error is defined as the proportion of non-links actually declared links. For the St. Louis data it is known which pairs are the links a priori. This information was obtained through tedious follow-up operations. With this information, the Type 2 error can be controled when applying the Fellegi-Sunter decision rule.

There are exactly 9823 links among the pairs. Table 1 contains the number of links that were actually recovered, applying the Fellegi-Sunter rule, under the 2 models presented previously and under an ad-hoc model, for 3 different controled Type II errors. The ad-hoc model is based on informal advice from W. E. Winkler (1989). The principle behind it is to improve the performance of the conditional independence model by adusting its parameters, rather than using a more elaborate model. The adjustments are largely based on experience and past knowledge of similar process. One such adjustment for example, is the increase of the value of the term corresponding to the effect of the first name in (2) to ensure that pairs of records agreeing on the first name be weighted heavily. The advantages of this method is that it does not require estimation procedures beyond those used for the conditional independence model.

Clearly, the model including interactions is the best when the tolerated error is at its smallest (.01). At that error level the conditional independence model is poor, but the ad-hoc model does fairly well. If the tolerated error goes up to .02, then both the independence model and the ad-hoc model catch-up on the model with interactions. This trend continues as the error is allowed to climb to .03. At that point the independence model is only 36 links behind the model with interactions, whereas the ad-hoc model and the model with interactions are virtually the same.

9 Conclusion

The model with interactions clearly gives the best performance when the tolerated Type II error is small. When the tolerance on the type II error is relaxed, the other methods may be just as good, especially the ad-hoc procedure, in this type of situation.

Table 1: Links Recovered For Three Error Levels

	Independence	Interactions	Ad-hoc
Error	.01	.01	.01
Links	7273	9712	9562
Pairs	7346	9808	9659
Error	.02	.02	.02
Links	9636	9758	9765
Pairs	9824	9952	9960
Error	.03	.03	.03
Links	9740	9776	9783
Pairs	10038	10062	10097

References

Fellegi, I. P. and Sunter, A. B. (1969), "A Theory for Record Linkage," *j. Am. Statist. Assoc.*, **40**, 1183-1210.

Goodman, L. A. (1974), "Exploratory Latent Structure Analysis Using Both Identifiable and Unidentifiable Models," *Biometrika*, **61**, 2, 215-231.

Haberman, S. J. (1979), *Analysis of Qualitative Data, Vol. 2*, Academic Press ed.,

Haberman, S. J. (1976), "Iterative Scalling Procedures for Log-Linear Models for Frequency Tables Derived by Indirect Observation," *1975 ASA Proc. of the Statist. Comp. Sec.* 45-50.

Thibaudeau Y. (1989). "Fitting Log-Linear Models in Computer Matching," *1989 ASA Proc. of the Satist. Comp. Sec.* 283-288.

Winkler, W. E. (1989). "Methods for Adjusting for a Lack of Indepence in an Application of the Fellegi-Sunter Model of Record Linkage." *Survey Methodology*, Vol. 15, No. 1, 101-117.

This paper reflects views of the author and not necessarly those of the Census Bureau.

Optimal Allocation for the Estimation of Attributable Risk

R. K. JAIN
Department of Mathematics and Statistics
Memorial University of Newfoundland
St. John's, Newfoundland, Canada A1C 5S7

Abstract

This paper derives an expression for the optimum sampling allocation under the minimum variance criterion of the estimated attributable risk for case-control studies. Various optimal strategies are examined using alternative exposure-specific disease rates.

KEY WORDS: Odd Ratio, Relative Risk and Attributable Risk.

1 Introduction

Mullooly (1987) derived expressions for the optimal number of cases and controls that minimize the total sample size and ensure the required level of precision for exposure-specific disease rates. Unequal sample allocation rule for various types of clinical studies was discussed by Gail et. al (1976). They suggested the so-called 'square root rule' to the case when the response variable has a different variance in each group. They presented various techniques to determine the optimal number of subjects. Brittain and Schlesselman (1982) examined the problem of optimal allocation for comparing proportions, p_1 and p_2, in two groups of clinical trial or follow-up studies. The criterion chosen was the precision of the estimator. In a series of papers, Walter (1975, 1976, and 1978) discussed the estimation procedures for estimating attributable risk and its role in epidemoilogical research. Walter and Morgenstern (1985) stressed the importance of optimal sampling plan which is essentially dependent on the choice of measures for summarizing the data. The purpose of this paper is to derive expression for optimal strategies in determining allocation rules under the minimum variance criterion of the estimated variance of attributable risk for case-control studies.

2 Optimal Allocation

The odd ratio as an approximation to the Relative Risk of disease in a group of people exposed to a certain risk factor, compared to those not exposed, has been widely used since its introduction by Cornfield (1951). Epidemiologists and public health officials suggested the so-called 'Attributable Risk.' The measure of 'Attributable Risk' suggests the potential impact on disease frequency of eliminating the exposure in the population.

Consider the following 2×2 contingency Table 1 for possible association between a dichotomous study factor (A = exposed or unexposed) and a dichotomous disease outcome (B).

Table 1. Data Layout

A	B		
	Cases	Controls	
Exposed	a_1	b_1	
Unexposed	a_0	b_0	
Total	n_1	n_2	n

Denman and Schlesselman (1983) estimated the attributable risk which is given by

$$\hat{\lambda} = \frac{a_1 b_0 - b_1 a_0}{n_1 b_1} \tag{1}$$

which can be expressed as follows:

$$\hat{\lambda} = p_1 - \frac{q_1}{q_2} \cdot p_2 \tag{2}$$

where $p_1 = \frac{a_1}{n_1}, p_2 = \frac{b_1}{n_2}$ such that

$$p_1 + q_1 = 1, \text{ and } p_2 + q_2 = 1.$$

It is assumed that p_1 and p_2 are independently binomially distributed. Walter (1976) has shown that the variance of $\hat{\lambda}$ may be estimated by

$$\text{Var}\,(\hat{\lambda}) = \left(\frac{a_0 n_2}{b_0 n_1}\right)^2 \left[\frac{a_1}{a_0 n_1} + \frac{b_1}{b_0 n_2}\right]. \tag{3}$$

The equation (3) can be rewritten as

$$\text{Var}\,(\hat{\lambda}) = \left(\frac{q_1}{q_2}\right)^2 \left[\frac{p_1}{nF(1-p_1)} + \frac{p_2}{n(1-F)(1-p_2)}\right] \tag{4}$$

where F denotes the proportion of the total sample subjects which are assigned to group 1 (cases), i.e., $F = \frac{n_1}{n}$.

Minimization of Var $(\hat{\lambda})$ requires differentiating equation (4) partially with respect to F and then equating to zero. Solving, one gets

$$F = \frac{\sqrt{\frac{q_2}{p_2}}}{\sqrt{\frac{q_2}{p_2}} + \sqrt{\frac{q_1}{p_1}}}. \tag{5}$$

The optimal sample size allocation to cases and controls can be obtained by using Equation (5) for various combinations of p_1 and p_2.

3 Concluding Remarks

It is often required to choose a particular combination of n_1 and n_2 that maximizes the precision of the estimator. Walter and Morgenstern (1985) emphasized that the optimal sampling strategy depends on the choice of function of p_1 and p_2. For example, Mullooly (1987) discussed the optimum sampling strategies based on precise estimation of disease rate in the exposed population. However, expression (5) minimizes the variance of attributable risk given by equation (4) for a given fixed total number of subjects. The choice of estimator for summarizing the data dictates the appropriate allocation rule.

References

Brittain, E. and Schlesselman, J.J. (1982). Optimal allocation for the comparison of proportions. *Biometrics* 38, 1003-1009.

Cornfield, J. (1951). A method of estimating comparative rates from clinical data: Applications to Cancer of the lung, breast and cervix. *J. of the National Cancer Institute* 11, 1269-75.

Denman, D.W. and Schlesselman, J.J. (1983). Internal estimation of the attributable risk for multiple exposure levels in case-control studies. *Biometrics* 39, 185-192.

Gail, M.; Williams, R.; Byar, D.P. and Brown, C. (1976). How many controls? *J. Chron. Dis.* 29, 723-731.

Leung, H. M. and Kupper, L.L. (1981). Comparisons of confidence intervals for attributable risk. *Biometrics* 37, 293-302.

Mullooly, J.P. (1987). Sample sizes for estimation of exposure-specific disease rates in population-based case-control studies. *American Journal of Epidemiology* 125, 1079-1084.

Pentico, D.W. (1981). On the determination and use of optimal sizes for estimating the difference in means. *The American Statistician* 35, 40-42.

Walter, S.D. (1975). The distribution of Levin's measure of attributable risk. *Biometrika* 62, 371-375.

Walter, S.D. (1976). The estimation and interpretation of attributable risk in health research. *Biometrics* 32, 829-849.

Walter, S.D. (1978). Calculation of attributable risks from epidemiological data. *International Journal of Epidemiology* 7, 175-182.

Walter, S.D. and Morgenstern, H. (1985). A note on optimal sampling for the comparison of proportions or rates. *Statistics in Medicine* 4, 541-542.

Bandit Strategies for Ethical Sequential Allocation

Janis P. Hardwick[1]
Statistics Department

Quentin F. Stout[2]
EECS Department

University of Michigan, Ann Arbor, MI 48109

Abstract

The problem of allocating patients in a two treatment clinical trial with dichotomous response is considered. The trial goal is to determine the better treatment while incurring as few patient losses as possible. Several allocation rules are compared and it is found that *bandit* strategies perform well on both criteria in that they achieve nearly optimal power while keeping expected trial failures nearly minimal. The rules are also evaluated according to their computational complexity.

1 Introduction

Researchers designing clinical trials often encounter difficulties when trying to determine the best way to allocate patients to treatments so that trial goals may be achieved and the costs to all concerned kept at a minimum. Conventional designs, in which subjects are allocated to groups in equal or predetermined proportions, have good decision making properties but lack the flexibility to incorporate other desirable design goals. Adaptive designs, in which allocation strategies may depend on data observed during the trial, have more flexibility. The consideration of adaptive techniques raises the question of what an *optimal* allocation rule is for a problem where statistical merit is not the only measure of the quality of a design. This question is complex and intriguing, and it deserves more attention than it is given here, where only a simple trial set-up is examined. What we can show, however, is that adaptive designs based on optimal strategies for *bandit* problems perform well according to *multiple* criteria, which include but are not restricted to the ability to make a good terminal decision. In particular, these rules are evaluated according to ethical and computational criteria and then compared with standard fixed allocation techniques.

Now, consider a clinical trial in which we wish to compare two treatments and determine, if possible, which has the higher efficacy rate. The patients, who enter the trial sequentially, are to be allocated to one of the two therapies in such a way that trial goals are met as well as possible. While any complete description of a clinical trial design should address all aspects of trial protocol (e.g., eligibility criteria, interpretation of responses, data analysis, etc.), we focus on the effects of changing allocation rules within otherwise fully specified designs.

It is assumed that the sample size for the trial is a fixed number, n, but that the sample sizes for the treatment groups, n_1 for T_1 and n_2 for T_2, may be random. The response variables, X and Y from T_1 and T_2 respectively, are independent Bernoulli random variables such that

$$(1)\quad X_1, X_2, \cdots \sim B(1, P_1); \quad Y_1, Y_2, \cdots \sim B(1, P_2)$$

where $(P_1, P_2) \in \Omega$, for $\Omega = (0,1) \times (0,1)$.

An *allocation rule*, γ, is defined to be a sequence $(\gamma_1, \ldots, \gamma_n)$ such that,

$$\gamma_i = \begin{cases} 0, & \text{if } T_1 \text{ is used for patient } i; \\ 1, & \text{if } T_2 \text{ is used at patient } i, \end{cases} \qquad i = 1, \ldots, n.$$

It is required that the decision, γ_i at stage i, depend only on the information available at that time.

The parameter of interest is the mean difference in responses, $\Delta = P_2 - P_1$, and T_1 is said to be *superior* to T_2 if $\Delta > 0$, and *inferior* if $\Delta < 0$. The *terminal decision rule* depends on the maximum likelihood estimate for Δ which, after n observations, is given by

$$\hat{\Delta}_n = \hat{\Delta}_n(\gamma) = \overline{Y}_{n_2} - \overline{X}_{n_1},$$

where $n_1 = \gamma_1 + \ldots + \gamma_n$, $n_2 = n - n_1$, and

$$\overline{X}_{n_1} = \frac{1}{n_1} \Sigma_{j=1}^{n} \gamma_j X_j; \quad \overline{Y}_{n_2} = \frac{1}{n_2} \Sigma_{j=1}^{n} (1 - \gamma_j) Y_j.$$

2 Design Characteristics

With the primary goal being to select the better of two competing therapies, the decision rule has been formulated to test the hypothesis

$$(2)\qquad H_0 : \Delta < 0 \quad vs. \quad H_1 : \Delta > 0,$$

and it specifies

$$(3)\qquad \begin{array}{lll} \text{Reject } H_0 & \text{if} & \hat{\Delta}_n > 0; \\ \text{No decision} & \text{if} & \hat{\Delta}_n = 0; \\ \text{Fail to reject } H_0 & \text{if} & \hat{\Delta}_n < 0. \end{array}$$

[1]Research supported in part by National Science Foundation under grant DMS-8914328.

[2]Research supported in part by National Science Foundation/DARPA under grant CCR-9004727.

An informative measure of how well a test performs is given by its *power*. For this problem, the power is simply the probability, as a function of $\mathbf{P} \in \Omega$, of correctly identifying the superior treatment. In practice, a rule allowing the *no decision* option should not be used without a null hypothesis of equality and corresponding acceptance region. We would prefer, in fact, a test that not only recognizes similar treatment effects with high probability, but also one that has maximum power at the *smallest clinically* significant difference between the parameters. The testing regions here, however, have been established so that we may study the behavior of the allocation rules over the entire parameter space and obtain lower bounds for the power of (3). In [3], we examine problems incorporating both type I and II errors.

It is not difficult to show that, for any $\mathbf{P} \in \Omega$, the probability of making an incorrect decision based on (3) is minimized by allocating patients to therapies in equal proportions. This may be achieved via alternating assignments or by constrained or blocked randomization. Since an equal allocation rule guarantees that fully half of the patients are assigned to the inferior treatment, designs utilizing them tend to incur more failures than may be necessary for the decision process. Our evaluations of allocation rules are based on three criteria:

1. The probability of making a 'correct' decision at the end of the trial,
2. The expected number of failures during the trial,
3. The complexity of the computations required to utilize the design.

Due to space limitations, the manner in which these criteria are assessed is quite simplistic. While each of these items can be viewed from many angles, the results (Section 4) seem to be representative of the behavior of the allocation rules in more general settings as well.

2.1 Bandit Problems

The sampling plans that we propose are based on optimal rules for multi-armed bandit problems. In a bandit problem, the goal is to maximize the sum of weighted outcomes arising from a sequence of experiments from *arms* whose outcomes follow the laws of a specified Bayesian model. A *bandit allocation rule* is thus one that utilizes prior information on unknown parameters together with incoming data to determine optimal selections at each stage of the experiment. The weighting of returns is known as *discounting* and it consists of multiplying the payoff of each observation by the corresponding element of a discount sequence. The properties of any given bandit allocation rule will depend upon the associated discount sequence and prior distribution.

Here we have only a *two-armed bandit* (TAB), but these techniques generalize easily to problems with several arms. Let the outcomes for the two treatment arms be given by (1), and model the prior information on the success rates, p_1, p_2, as independent beta distributions

$$p_1 \sim \mathrm{Be}(a_0, b_0) \quad \text{and} \quad p_2 \sim \mathrm{Be}(c_0, d_0).$$

At any stage $m \leq n$, the posteriors for p_1 and p_2 are

$$(4)\quad (p_1 \mid k, i, j) \sim \mathrm{Be}(a, b);\ (p_2 \mid k, i, j) \sim \mathrm{Be}(c, d)$$

where $k = \Sigma_{i=1}^{m} \gamma_i$, $i = \Sigma_{i=1}^{k} X_i$, $j = \Sigma_{i=1}^{m-k} Y_i$, and

$$a = i + a_0, \quad b = k - i + b_0,$$
$$c = j + c_0, \quad d = m - k - j + d_0.$$

The posterior means of p_1 and p_2 at m are simply $\mathbf{E}_m[p_1] = a/(a+b)$ and $\mathbf{E}_m[p_2] = c/(c+d)$, where $\mathbf{E}_m$ denotes expectation in the model (4).

Typically, the choice of a prior distribution will depend, somewhat subjectively, on the knowledge of the investigator preceding the trial. We use independent uniform priors here, $a_0 = b_0 = c_0 = d_0 = 1$, because they contain no initial bias and little information, and because the parameters of the beta posteriors concisely summarize the relevant study data to date.

It is worthwhile to note that these allocation rules, which arise within a Bayesian framework, are being evaluated according to frequentist standards. In Section 4, the Bayesian design is seen to have had little effect on the results of the trial from this viewpoint. However, if desired, the design may be set up to impact the trial and its results more heavily, since investigators can strengthen and/or bias the parameters of the beta distributions to reflect a preferred level of information.

2.2 Ethical Criteria

An advantage of using bandit problems to model clinical trials is that elements of the discount sequence can be selected to represent an ethical decision regarding the relative importance of the patient outcomes both during the trial and in the future. At each stage of the sequential decision process, a bandit allocation rule is a function both of the effort to gather information and of the effort to gain immediate reward. Here, we consider two discount sequences, $\{1, \beta_1, \beta_2, \ldots, \beta_n\}$: the *n-horizon uniform* sequence with $\beta_i = 1$, $i = 1, \ldots, n$, and the *geometric* sequence, $\{1, \beta, \beta^2, \beta^3, \ldots\}$, $0 < \beta < 1$.

In the uniform, finite horizon case, the optimal strategy will begin by emphasizing the gathering of information with the result being that the first patients will be treated rather like patients in an equal allocation trial where one assumes throughout that the treatments offer the same prognosis. Toward the end of the study,

with a decision imminent, the emphasis on immediate reward is increased until, at the last stage, a completely myopic rule is used. In the geometric case, it is assumed that that there will always be more patients, so the need for information is never completely absent as in the last stage of a finite horizon problem. However, as more and more patients are treated, the need to sacrifice immediate reward to gain information will decrease. Since the sample size in the present problem is fixed at n, we truncate the allocations after n observations. Thus bandit allocation strategies for problems with geometric discounting are not exactly optimal for the truncated case. As we see, however, these rules still provide good model strategies for the problem at hand. See Hardwick [2] for further discussion of the incorporation of geometric bandit strategies in clinical trial designs.

2.3 Computational Criteria

Ethical attributes aside, an experimental design must be straightforward to carry out if it is to be useful. For computational purposes, this means that the rules should use reasonable amounts of time and space (memory), and be sufficiently easy to program. We distinguish here between the computational requirements to set design parameters and those needed to carry out the trial. In general the former will be significantly greater than the latter, but can be carried out on large computers without significant deadline pressure. The latter may require timely response, and may often be performed on personal computers. The latter will be analyzed here in the next section, while the former will be discussed in [3].

3 Allocation Rules

The following three allocation rules were evaluated with respect to the given criteria:

TAA = Truncated Alternating Allocation,
UB = Uniform Bandit, and
TGLB = Truncated Gittins Lower Bound.

The "truncation" in TAA and TGLB refers to a rule whereby, if a state is reached such that the final decision can not be influenced by any further outcomes, then the treatment with the best success rate will be used for all further patients.

3.1 Uniform Bandit

By definition, the n-horizon uniform TAB uses prior and accumulated information to minimize the number of failures during the trial. We can determine the optimal strategy for this bandit problem using dynamic programming. Let $\mathcal{F}_m(i,j,k,l)$ denote the minimal possible expected number of failures remaining in the trial, if m patients have already been treated and there were i successes and j failures on T_1, and k successes and l failures on T_2. (Note that one parameter can be eliminated since $m = i+j+k+l$.) The algorithmic approach is based on the observation that if T_1 were used on the next patient, then the expected number of failures for patients $m+1$ through n would be

$$\begin{aligned}\mathcal{F}_m^{T_1}(i,j,k,l) &= \mathbf{E}_m[p_1]\cdot\mathcal{F}_{m+1}(i+1,j,k,l) + \\ &\quad \mathbf{E}_m[1-p_1]\cdot(1+\mathcal{F}_{m+1}(i,j+1,k,l))\end{aligned}$$

while if T_2 were used then we would get

$$\begin{aligned}\mathcal{F}_m^{T_2}(i,j,k,l) &= \mathbf{E}_m[p_2]\cdot\mathcal{F}_{m+1}(i,j,k+1,l) + \\ &\quad \mathbf{E}_m[1-p_2]\cdot(1+\mathcal{F}_{m+1}(i,j,k,l+1)).\end{aligned}$$

Therefore $\mathcal{F}$ satisfies the recurrence

$$\mathcal{F}_m(i,j,k,l) = \min\{\mathcal{F}_m^{T_1}(i,j,k,l), \mathcal{F}_m^{T_2}(i,j,k,l)\}$$

which can be solved by dynamic programming, starting with patient n and proceeding toward the first patient.

For the m^{th} patient there are $\Theta(m^3)$ possible values of i,j,k,l, so to evaluate all possible combinations of m, i, j, k, and l requires $\Theta(n^4)$ computations. A clever implementation might not evaluate all possible values, but a straightforward implementation, as used here, needs to do so, and empirical evidence indicates that, in fact, $\Theta(n^4)$ values must be computed. The space requirements can be kept at $\Theta(n^3)$ (see [3]).

3.2 Gittins Lower Bound

According to a theorem of Gittins and Jones [1], for bandit problems with geometric discount and independent arms, for each arm there exists an index with the property that, at any given stage, it is optimal to select, at the *next* stage, the arm with the higher index. The index for an arm, the *Gittins Index*, is a function only of the posterior distribution and the discount factor β. While the existence of the Gittins Index removes many computational difficulties associated with other bandit problems, the only known technique for computing the index involves an iterated dynamic programming approach which is computationally intensive when β is close to 1 (see [1]). Unfortunately, these are the β values needed to produce tests of suitable power.

Here we show that very good results can be achieved by utilizing an easily computed approximation. For an arm with posterior distribution $\mathrm{Be}(a,b)$, a lower bound for the Gittins Index is given by (see [1,2])

$$\Lambda_r = \frac{\frac{\Gamma(a+1)}{\Gamma(a+b+1)} - b\sum_{i=1}^{r}\beta^i\frac{\Gamma(a+i)}{\Gamma(a+b+i+1)}}{\frac{\Gamma(a)}{\Gamma(a+b)} - b\sum_{i=1}^{r}\beta^i\frac{\Gamma(a+i-1)}{\Gamma(a+b+i)}}.$$

Parameters →		$\Delta = 0.1$			$\Delta = 0.3$		
↓	Criteria	TAA	TGLB	UB	TAA	TGLB	UB
n=20	Power	0.671	0.667	0.647	0.913	0.906	0.874
$\beta = .999$	Average Failures	9.947	9.774	9.768	9.505	8.330	8.217
n=50	Power	0.760	0.754	0.708	0.985	0.982	0.947
$\beta = .9999$	Average Failures	24.828	24.148	24.117	23.489	19.673	19.214
n=100	Power	0.841	0.835	0.771	0.999	0.996	0.980
$\beta = .99999$	Average Failures	49.614	47.779	47.642	46.762	38.051	36.984
n=150	Power	0.890	0.885	0.811	1.000	0.998	0.989
$\beta = .999999$	Average Failures	74.393	71.243	70.890	70.031	56.367	54.611

Table 1: Comparisons of Discrimination and Ethical Criteria

Because Λ_r is a unimodal function of r, the best such lower bound is Λ_{r^*}, where $r^* = \min\{r : \Lambda_r - \Lambda_{r+1} \geq 0\}$. Each Λ_r can be computed from the previous one in a constant number of steps, so the total time to compute the best lower bound is proportional to $r^* + 1$.

The computational requirements of the TGLB approach are difficult to analyze since they depend upon the value of r^* and upon the successes and failures encountered. In the simplest implementation, the approximate indices for both treatments are computed at each stage and compared to determine the best choice. However, computation can be saved by noting that a "play the winner" property holds, in that if the indices resulted in treatment i being chosen for the previous patient, and the outcome was a success, then they will again choose treatment i. Therefore an index needs to be computed only when a failure has occurred, and then only for the treatment that failed since the posterior distribution of the other treatment is unchanged.

4 Results

The results of our investigations are summarized in Tables 1 and 2. The computational techniques used are explained in [3].

Table 1 shows that TAA, which is optimized to make the correct selection, incurs a large ethical cost, while UB, which is optimized to minimize failures, has a poor discrimination ability. The TGLB rule is a compromise with nearly the power of TAA and nearly the ethical behavior of UB. Note that TGLB has an extra parameter, β, which must be adjusted to optimize its performance. One can show that β must converge to 1 as n increases in order to obtain increasing power. The specific values of β used have been indicated.

Table 2 compares UB and TGLB on computational grounds. TAA was not included since the total computation time is merely proportional to n, i.e., $\Theta(n)$. For

Parameters		UB	TGLB	
n	β		$\Delta = 0.1$	$\Delta = 0.3$
20	0.999	8,855	180	174
50	0.9999	292,825	611	597
100	0.99999	4,421,275	1,705	1,687
150	0.999999	21,947,850	4,124	4,109

Table 2: Comparisons of Computational Time

UB, the value presented is the number of evaluations of $\mathcal{F}$ which occur, each of which takes a constant amount of time. Thus the computational time for a clinician to utilize UB is proportional to the value presented and may be prohibitive. For TGLB, the value also represents a quantity which is proportional to the total computational time needed to utilize TGLB during a trial. The value presented is the average, over all trials, of the total number of Λ_r values which must be computed for index calculations throughout the trial. While space requirements were not tabulated, recall that UB needs $\Theta(n^3)$ space and TGLB needs only $\Theta(1)$ space.

References

[1] Berry, D. and Fristedt, B. (1986), *Bandit Problems: Sequential Allocation of Experiments.* Chapman and Hall, New York.

[2] Hardwick, J. (1986), *The Modified Bandit: an approach to ethical allocation in clinical trials.* Ph.d. thesis, University of California at Los Angeles.

[3] Hardwick, J. and Stout, Q. F. (1991), Computational aspects of sequential allocation for testing with multiple criteria. In progress.

Dynamic Graphics: Linked Points, Lines and Regions with Applications to Spatial Data Modelling[1]

John Haslett, Ronan Bradley,
Department of Statistics, Trinity College, Dublin 2, Ireland.

Abstract

The convergence of Geographical Information Systems (GIS) and Statistical Dynamic Graphics has led to the development of a number of new concepts in spatial data analysis. This paper discusses how such ideas may be extended in the context of spatial modelling. The software environment we use - REGARD - incorporates the GIS ideas of point, line and region 'layers of data' pertaining to an area under study.

The spatial variation here is modelled by a variogram. The model may be used to generate new statistical views of the data, which may be regarded as diagnostics. Aspects of the model may be decomposed spatially in the linked-views environment. Many such aspects are naturally viewed as being defined point-wise. Aspects which refer to data pair-wise can naturally be associated with line-objects. Many analyses are oriented to defining regions of anomalous behaviour. The paper will illustrate these intert-twined ideas.

1 Introduction

The linked windows concept in dynamic graphics supports a number of views of the data. Each view focuses on some relatively simple aspect of the data; dynamic linking of these provides a platform for understanding the variation. We introduce below a new generalisation of such views for spatial data and exploit it here for the very specific purpose of studying diagnostics for spatial models.

Spatial data may best be thought of as data which are defined on objects which have location. A central view in spatial data analysis is therefore that of the objects and of their physical locations; we call this a Map View. The simplest objects are point objects. Stream sediment data are naturally thought of as multivariate data on geochemical composition associated with small samples which may be thought of as points. More generally one may think of data on lines or on regions. Regional data may be administrative or geological. Disease rates are naturally defined on regions. Lines may be roads or streams: the variables may be stream width or traffic on the road. Many projects in the environmental sciences in particular in fact involve integrating, within one study, data of various types associated with different types of objects. Mineral exploration thus naturally involves assembling data on satellite imagery (pixels - regions or points), geology (regions) geochemistry (points) and geological faults (lines). Many other examples abound.

One way to approach the task of integrating is through Geological Information Systems (Burrough,1986). This supports layers of information which may be superimposed both visually and logically. A complementary proposal, using linked views, is to treat each layer as a data matrix comprising a list of objects with associated attributes (variables) and to support a Map View with visually superimposed point line and region objects, each of which is separately associated with linked views of the attributes. See Figure 1 and Haslett and Cameron (1990) for further aspects of this study. At the simplest level, selecting a case in, for example, a scatterplot of two variables causes that case to be highlighted in other views of that case, including its object in the Map View. At a more sophisticated level, selecting objects in one layer can cause objects in other layers to become selected. We refer to this as cross-layer linking. Thus a histogram of the attributes in a region layer can be used to select one or more regions of interest; these regions can then cause points within them, but in a point layer, to become selected; these in turn will be highlighted in a view of the attributes of the points. Such ideas have been implemented in REGARD, experimental software under development in Trinity College, Dublin. See Haslett et al (1990), Haslett et al (1991). One important type of data which lends itself well to such analysis is data defined on a network. Such data are commonplace (traffic flow on a telephone network, airline traffic between cities, trade flows between counties). In this context we may use inter-connected point and line layers to analyse the data defined on such a network; lines connect pairs of nodes.

Thus views of the different objects and of their attributes can be linked to each other. This provides one avenue towards the integration of such data. An alternative use of such a platform is in the spatial decomposition of aspects of the data.We illustrate this by reference to a spatial model of univariate data defined on points. We will see that a network can be a useful vehicle to study together the pairwise interactions and the point wise data.

[1] The support of Apple and of EOLAS (Dublin) and of CSIRO (Sydney) are gratefully acknowledged. REGARD is experimental software, written by Graham Wills of the Department of Statistics, TCD.

2 Spatial Decomposition

We begin by decomposing the mean and variance of data on the proportion of stone to be found at various point locations in a field (Burgess and Webster, 1980).

2.1 Point-wise decomposition of the mean and Pair-wise decomposition of the variance.

Consider data $z(\underline{x}_i)$ defined at points, $\underline{x}_i$, i=1,..n. Trivially, the mean of these is

$$\bar{z} = \frac{1}{n}\sum_{1}^{n} z(\underline{x}_i)$$

Thus the histogram view of $z(\underline{x}_i)$ provides the possibility of a spatial decomposition of the mean $\bar{z}$ by cross referring to the physical locations in the Map View. See Figure 2. As the smaller values are to be found in one part of physical space (and the larger values in another) there is certainly spatial structure. The mean is a very useful summary of unimodal bell shaped distributions; in other cases this is less clear. In Figure 2 for example there is a suggestion that there are in fact two modes. This is reinforced by the Map View.

Less trivially, the variance may be written

$$s_z^2 = \frac{1}{n(n-1)}\sum_{i>j} (z(x_i) - z(x_j))^2$$

Thus a histogram view of $(z(x_i) - z(x_j))^2$ provides the possibility of a spatial decomposition of the variance by cross referring to the physical locations in the Map View. Since $(z(x_i) - z(x_j))^2$ involves a pair of points, it is natural to associate it with a line from $\underline{x}_i$ to $\underline{x}_j$. Further it is more natural to view $(z(x_i) - z(x_j))^2$ through a scatterplot against $h_{ij} = |\underline{x}_i - \underline{x}_j|$. See Figure 3. This in fact a plot of the 'variogram cloud' (Chauvet, 1982) and is closely related to the empirical variogram of the geostatistical family of models of spatial variation.In Figure 3 a selection has been made of the pairs which have relatively high $(z(x_i) - z(x_j))^2$ for the separation. The corresponding lines are shown in the Map View; all other lines (n(n-1)/2 in total) remain invisible. The variance (and more specifically the empirical variogram) has been spatially decomposed pairwise. Certain pairs, in a band in the SE and associated with a single point in the NW, contribute strongly. See Bradley and Haslett (1990) for further discussion.

It is clear that there is a local outlier in the NW and some suggestion of a discontinuity in the SW. Below we see that by viewing the variation through a simple model these issues become more clearly defined.

2.2 Pair-wise decomposition of the likelihood.

If we model the data as being a partial realisation of a spatial stochastic process we may develop model derived measures of the data which may similarly be decomposed. Specifically consider the geostatistical model in which the data above are taken to be a partial realisation of a Gaussian stochastic process with given isotropic variogram $\gamma(h)$ (Journel and Huijbregts, 1978) in which

$$E\left\{\frac{1}{2}(Z(\underline{x}_i) - Z(\underline{x}_j))^2\right\} = \gamma(|\underline{x}_i - \underline{x}_j|) = \gamma(h_{ij})$$

Clearly, under the model, the data are a single realisation of a multivariate Normal distribution. Consequently the likelihood of the data can be written as a quadratic form in the $\{z(\underline{x}_i)\}$. A convenient representation for pair-wise decomposition is:

$$\text{-2LogLikelihood} = \text{constant} + \sum_{i>j} \omega_{ij}\{z(\underline{x}_i) - z(\underline{x}_j)\}^2$$

where the ω_{ij} terms have an interpretation close to the partial covariance of $(Z(\underline{x}_i), Z(\underline{x}_j))$ given the rest of the data. More specifically they can be seen by considering leave-one-out cross validation, the estimation of each data point in turn from all the others. Thus, by seeking the maximum likelihood estimator of the unknown $Z(\underline{x}_i)$ from $\{z(\underline{x}_j), j \neq i\}$ we find on differentiation that

$$\hat{Z}(\underline{x}_i) = \sum_j \omega_{ij} z(\underline{x}_j) / \sum_j \omega_{ij} = \sum_j \lambda_{ij} z(\underline{x}_j)$$

and that the variance of this estimator is $\sigma^2(\underline{x}_i) = \sum_{j \neq i} \omega_{ij}$.

The ω_{ij} are thus proportional to the kriging weights λ_{ij} used in cross-validation. They may also be interpreted as proportional to the correlations between cross-validation residuals at $\underline{x}_i$ and $\underline{x}_j$.In this context they have another interpretation, that of pair-wise leverage on the likelihood. More simply stated, for a pair of data values to contribute significantly to the (un)likelihood, which is in fact a measure of lack-of-fit, it is necessary not only that $\{z(\underline{x}_i) - z(\underline{x}_j)\}^2$ be large, but that ω_{ij} be large.

The model thus suggests another plot, that of $\{z(\underline{x}_i) - z(\underline{x}_j)\}^2$ against ω_{ij}. Pairs which are high in both are important; if their are also located in one part of space that part of space is contributing significantly to the lack of fit. Thus for lack of fit, ω_{ij} is a more useful metric than h_{ij} against which to judge the separation of two points. In Figure 4, such a plot is presented; a few pairs have been selected and are presented as lines in the Map View. These have been computed using

a variogram fitted to the data. They are seen to communicate the same message as previously, but much more crisply. Note the cross layer linking: selected lines in the line layer cause their end points to be selected in the point layer. We see, naturally, that these pairs of points are associated with the upper and lower ends of the distribution of the proportion of stone. Point-wise decomposition of the likelihood is also possible and attractive. In particular, it is possible to show that:

$$-2\text{LogLikelihhood} = \text{constant} + \sum_{i>j} z(\underline{x}_i)\{\hat{z}(\underline{x}_i) - z(\underline{x}_i)\}/\sigma^2(\underline{x}_i)$$

A scatterplot of $\{\hat{z}(\underline{x}_i) - z(\underline{x}_i)\}/\sigma(\underline{x}_i)\}$, the standardised cross-validation residual, vs $\{z(\underline{x}_i)/\sigma(\underline{x}_i)\}$ can therefore provide the basis for a point-wise decomposition of the likelihood. See Bradley and Haslett (1991).

3 Discussion

This paper has indicated one of a number of new possibilities for the use of linked windows. Here we have concentrated on its use as a platform for research on diagnostics in spatial modelling. New pairwise views of the data can be supported; it is thus possible to investigate a number of pairwise diagnostics. Two such have been offered; the decomposition of the empirical variogram may be perhaps described as a pre-modelling diagnostic, and the decomposition of the likelihood as a post modelling diagnostic. Spatial decomposition is a general principle: one can often ask " from where does the evidence come that .....". The likelihood can be expressed as a sum of terms defined pair wise, as above, or point-wise; see Bradley and Haslett (1991). Other pairwise and point-wise diagnostics can be created and, using our platform, can be investigated. Such procedures are likely to be particularly valuable in multivariate spatial processes.

The possibilities of integrating data of different spatial support are also important. In this context the ideas of cross layer linking provide a natural vehicle with which to examine such data. See Haslett et al (1991) for further examples.

References:

Bradley, R. and Haslett, J. (1990) Interactive Graphics for the Exploratory Analysis of Spatial Data - The Interactive Variogram Cloud CODATA Geostatistics meeting Leeds, Sept 1990

Bradley, R. and Haslett, J.(1991) High Interaction Diagnostics for Geostatistical Models of Spatially Referenced Data, Conference on Applied Statistics in Ireland, May 1990

Burgess, T.M. and Webster, R. (1980) Optimal Interpolation and Isarithmic Mapping of Soil Properties 1: the Semi-Variogram and Punctual Kriging', Journal of Soil Science, 31,2, pp315-331.

Burrough, P. (1986) Principles of Geographic Information Systems for Land Resource Assessment, Clarendon Press, Oxford

Chauvet, P.(1982) The Variogram Cloud', Proc. 17th APCOM Symposium, Colorado, 19-23 April, 1982, Colorado School of Mines, Golden Colorado, pp757-764.

Haslett, J. , Wills, G. and Unwin, A.R. (1990) SPIDER - An Interactive Statistical Tool for the Analysis of Spatial Data, Int. J Geog. Infor. Syst., 4, p285-296.

Haslett, J. and Cameron, M. (1990) Interactive Graphics for the Exploratory Analysis of Spatial and Spatial-Temporal Data, Karlsruhe Hydrology meeting, July 1990

Haslett, J., Bradley, R., Craig, P.S, Wills, G., Unwin, A.R.,(1991) Dynamic Graphics for Exploring Spatial Data, with application to Locating Global and Local Anomalies, to appear in American Statistician.

Haslett, J., Bradley, R., Dillon, M. and Wills, G. (1990) Interactive Graphics for the Exploratory Analysis of Spatial Data - Application to Data of Different Support CODATA Geostatistics meeting Leeds, Sept 1990

Journel, A.G. and Huijbregts, Ch. J. (1978) Mining Geostatistics, Academic Press

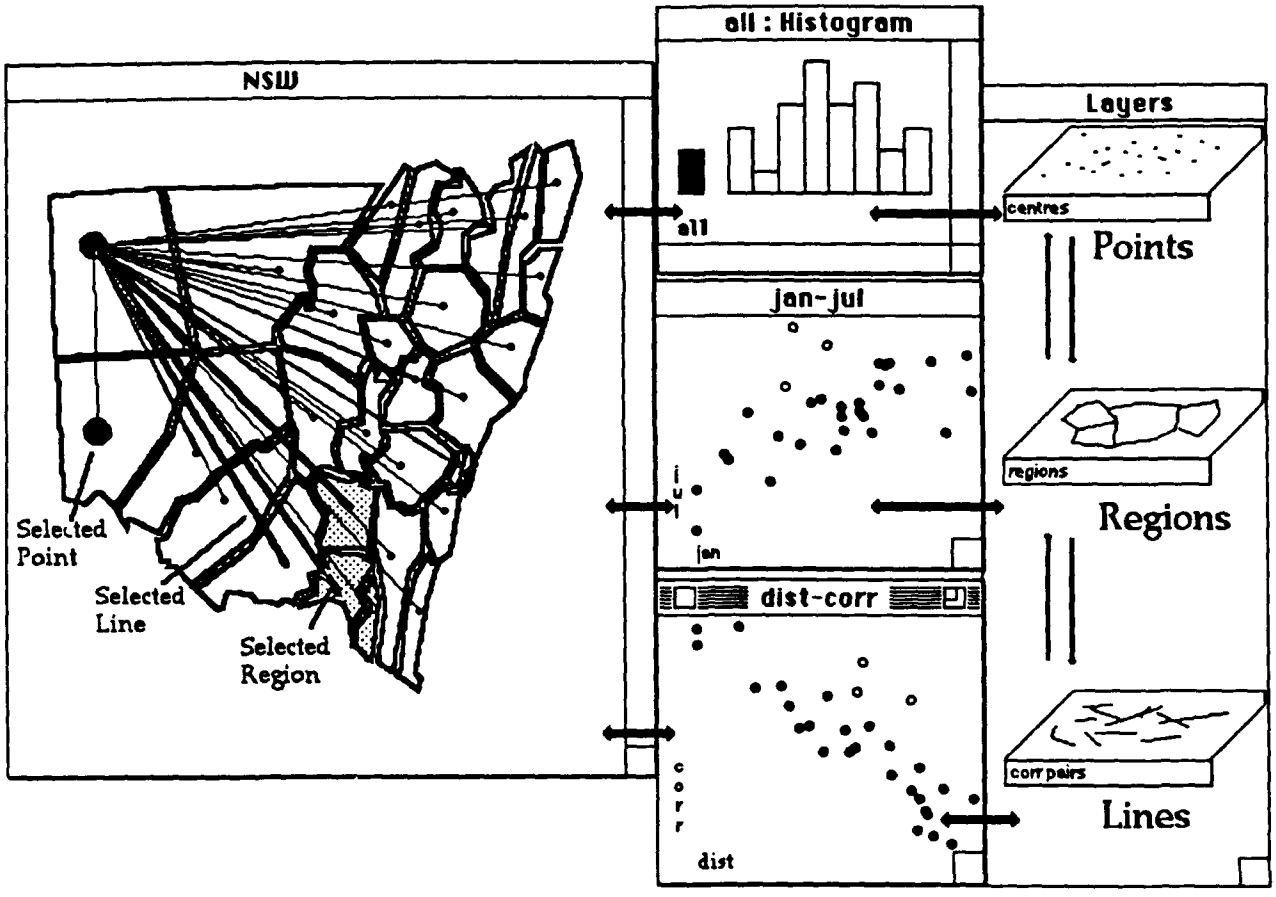


Figure 1: Layers of data. Shown are three layers of data, defined on points, regions and lines. The data pertain to rainfall data in 30 meteorological regions of New South Wales. Variables of interest are defined on the regions, on points within the regions and on pairs of points. The objects for each layer are visible in the Map View: statistical views of some of their attributes are shown. In each layer a few objects have been selected. Here there is no formal cross-layer linking; since the objects occupy the same physical space they are visually cross referred. The use of colour in the Map View is recommended.

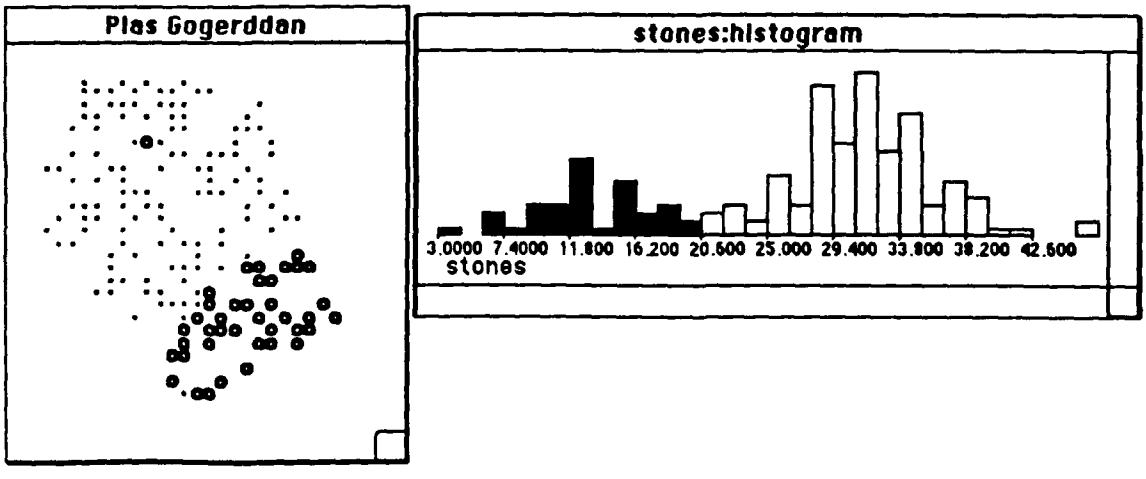


Figure 2: Data on the distribution of the proportion stone in soil samples in a field. Selecting the left hand tail of the distribution shows that there is a clear spatial structure to the variation. One unusual point has been identified

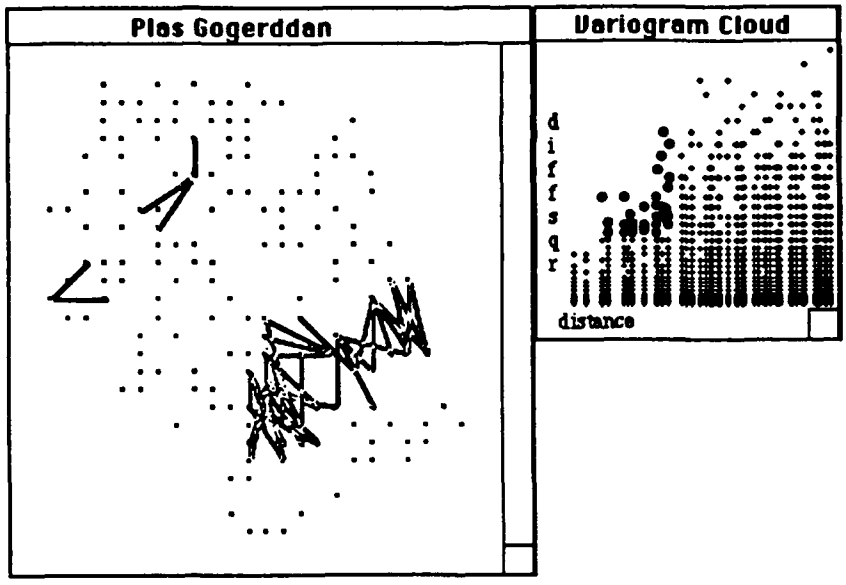


Figure 3: With each pair (line) is associated the distance between the pair and the squared difference between the two proportions. A scatter plot of these is shown. Some of these have been selected. These correspond to pairs which are unusually different, given their separation. The corresponding lines have been highlighted. The remainder of the (very many) lines remain invisible. It is clear that many such pairs are to be found in a region in the SW and associated with a single point in the NW.

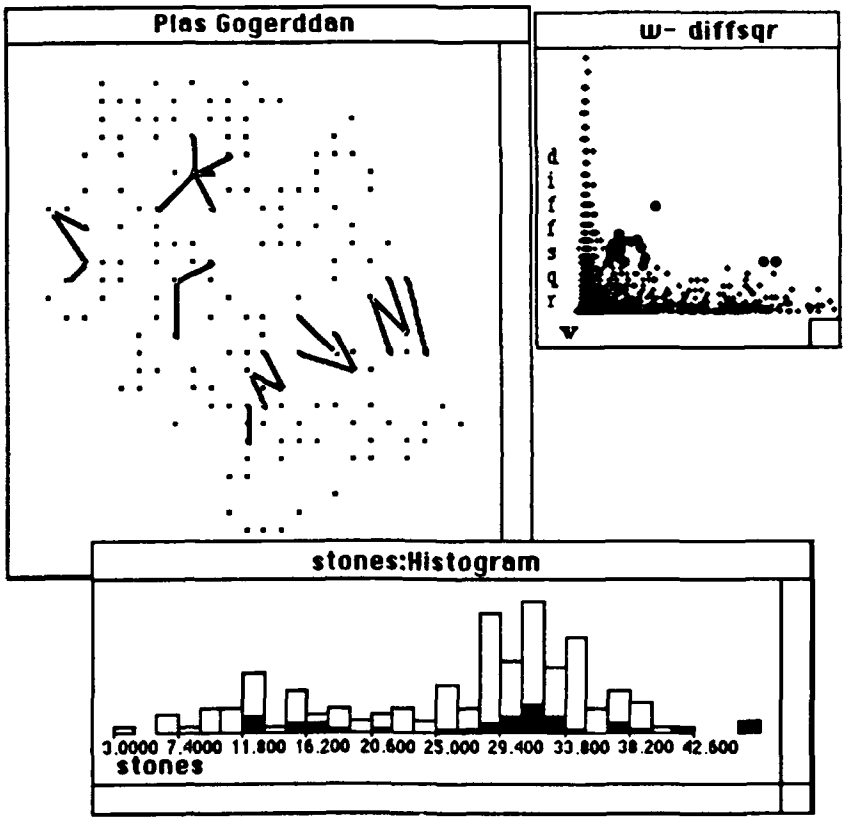


Figure 4: A few pairs simultaneously having high w and large squared difference have been selected. These correspond to the same features as in Figure 3, but much more clearly defined. In this case cross layer linking is active: thus the points at the ends of the selected lines have been selected and the point data values are shown in the histogram view of the data in the point layer.

D.F. Swayne, A. Buja, and N. Hubbell

XGobi Meets S: Integrating Software for Data Analysis

Deborah F. Swayne, Bellcore, dfs@bellcore.com
Andreas Buja, Bellcore, andreas@bellcore.com
Nancy Hubbell, University of Wisconsin and Bellcore

ABSTRACT

This paper describes an approach to integrating various computing tools used in data analysis. Integration is accomplished by creating direct manipulation panels which control and link disparate software. The linked programs can perform data manipulation, numerical analysis, static or dynamic graphics.

The two prototypes described here are integrated systems that are used to control data analysis sessions. XSmooth coordinates a smoothing session; it consists of a control panel and a plotting window and has a link to S. XClust coordinates a clustering session; it uses a panel to control an S process and one or more instances of XGobi, an interactive dynamic graphics program. The prototypes run in the X Window System™.

1 Introduction

There is a great deal of software for statistical computing available for UNIX®workstations, but no single system can do everything. A system which is rich in data manipulation functionality may lack dynamic graphics; a dynamic graphics package may not be easily programmed by a user; a system which is easily programmable may lack data analytic methods.

An analyst might then wish to use a variety of software on a single problem. This can prove difficult and cumbersome, because each system has its own command syntax and its own data representation.

One solution is to use a control program to manage the communication between these different elements. Its own command syntax should be quite simple, so that it does not burden the user with another language to learn.

We are exploring a method in which we create a control panel which becomes the user's means of interacting with several other pieces of software. The panel itself has a direct manipulation interface which allows the user to communicate with the panel by selecting buttons or menu items with the mouse.

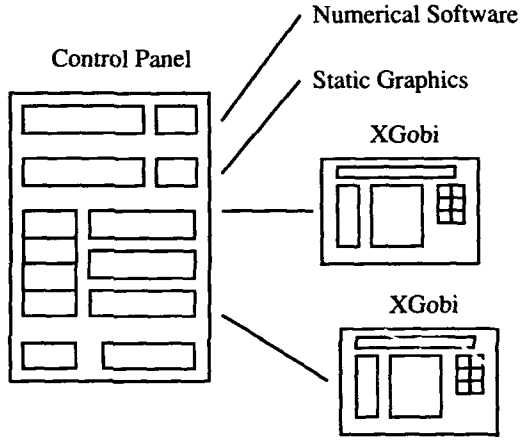


Figure 1: General Model

Each control panel manages a single application which includes some or all of the following: analytical and data manipulation software, static graphics displays, dynamic graphics software. The analytical software can be written by the user or it can be an independent interactive program. The static graphics software can be independent routines or part of a package. The dynamic graphics software is to be XGobi. Figure 1 shows a sketch of this approach.

The panel can control these elements in a few different ways. In the simplest case, it can use direct function calls, and even write directly into data structures. In other cases, it would use UNIX interprocess communication methods, sending and receiving data over pipes or sockets.

Elements in this model can sometimes communicate directly, without using the control panel as a translator. For example, instances of XGobi can share data using an X interprocess communication method; in fact, that is how linked brushing and identification are implemented

X Window System is a trademark of MIT.
UNIX is a registered trademark of UNIX System Laboratories.

in XGobi. Another sort of communication between elements occurs if the analytical software is S and the static graphics window is an S plotting window. In that case, the control panel sends plotting commands to S, and relies on S to communicate with the S plotting window.

The use of a direct manipulation interface to coordinate and link disparate software could be applied to several areas of statistical computing. Two examples are optimization problems and iterative cycles of data analysis.

In an optimization problem, the control panel could continuously print the value of the function to be optimized, using plots to enrich the feedback to a user. The user could interactively adjust parameters in response to this information. For example, a display could indicate that the routine is stuck at a local maximum, and the user could increase a step size, allowing the program to keep searching the solution space. The projection pursuit methodology developed by Cook et al. (1991) for XGobi provides an illustration of this approach.

During an iterative cycle of data analysis, an analyst executes a command that applies the initial model, then studies some numerical and graphical output to evaluate the model. After examining this output, the analyst adjusts the model and re-executes the command. In regression, for example, the analyst evaluates the model using statistics such as the residual sum of squares and the t-test for each coefficient, and uses graphical output such as residual and influence plots. In response to this evaluation, the analyst adjusts the model by adding or removing a term, eliminating outliers, and so forth. The regression may be repeated many times before the analyst is finally satisfied with the model.

In such a data analysis session, a user wants several kinds of information readily available at the same time: all the printed values returned by the regression function, various diagnostic plots, and plots of the raw data. The analyst's work can be made easier by a direct manipulation interface: the recomputation of the model is reduced to a couple of operations, key presses or clicks of a mouse button.

2 XSmooth

Smoothing was chosen for the initial prototype, for two reasons: first, a scatter plot with an added line or lines is the only graphical output required, and second, a smoothing session is well captured by the iterative formulation just described. The data analyst repeatedly adjusts one or more smoothing parameters and looks at plots of the smoothed curve and scatter plots of the raw data.

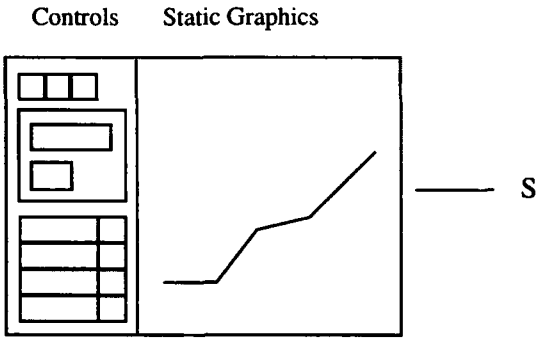


Figure 2: XSmooth Model

XSmooth is an integrated system with only two elements, as shown in Figure 2. There is a window containing both a control panel and a plotting region, and this window communicates with S, which performs the smoothing computations.

Three smoothing functions can be found in S (`lowess`, `smooth`, and `spline`), and each has at least one smoothing parameter. An S user who wants to find a smoothed curve for a pair of vectors `x, y` is likely to experiment with at least one of these routines several times. First, the scatter plot is generated:

```
plot(x, y)
```

Then a smoothed line can be added to the plot using `lowess`, where the argument `f` is the fraction of the data used for smoothing at each `x` point:

```
lines(lowess(x, y, f=.3))
```

At this point the user may generate several different smoothed lines, then decide the plot has become too noisy, regenerate the scatter plot, and continue to add smoothed lines until the preferred value of `f` is found.

Using XSmooth, the user executes an S function, passing it the name of the data to be smoothed:

```
xy <- cbind(x, y)
xsmooth(xy)
```

An XSmooth window appears, initially displaying the smoothed curve generated by using `lowess` with the default value of `f`. A user can now choose to work with `lowess` or to select the `smooth` or `spline` function. If `lowess` is chosen, the `f` argument is adjusted using a scrollbar, clicking on the arrows at either end of the

scrollbar for fine control. A single click on the "Send" button causes a new smoothed curve to be generated and plotted. A user can control various features of the plot with single button clicks: whether or not the raw data values are plotted, whether or not axes and axis labels are shown. A history of the most recent three smoothed curves is kept, and a user can include or exclude each of the three.

To communicate with S, XSmooth uses UNIX interprocess communication code. In response to a button click, XSmooth assembles a command in S syntax and sends it to S. Then it captures S's response and copies it into C data structures.

3 XClust

The second integrated system is used for clustering. This is another example of iteration in data analysis, but it uses a greater variety of graphical output than smoothing. One would want to see different views of the cluster structure, such as a dendrogram in a static graphics window and a scatter plot in a high-dimensional motion graphics system with brushing.

XClust, as shown in Figure 3, has a panel which controls one or more instances of XGobi and an S process with an S plotting window.

To perform clustering in S, a user is likely to repeat a sequence of operations a number of times, using diagnostic plots to guide the iterative procedure. First, a distance matrix is calculated, using one of several distance metrics:

```
d <- dist(x, metric="euc")
```

Then the hierarchical clustering tree is determined, using one of the clustering methods available in S:

```
tree <- hclust(d, method="compact", sim)
```

Now, the dendrogram itself can be plotted and studied, and there are several parameters to the plotting function:

```
plclust(tree, hang, unit, level, ...)
```

At this point, other diagnostics can be performed. For example, one can use the `cutree` function to cut the tree, specifying either a height or a number of clusters. The function returns a vector of cluster membership:

```
ind <- cutree(tree, k, h)
```

Using this vector, one can make pairwise scatter plots of the variables, plotting each cluster with a different color or glyph. One might plot the data in the space of the discriminant coordinates, or use other diagnostic tools.

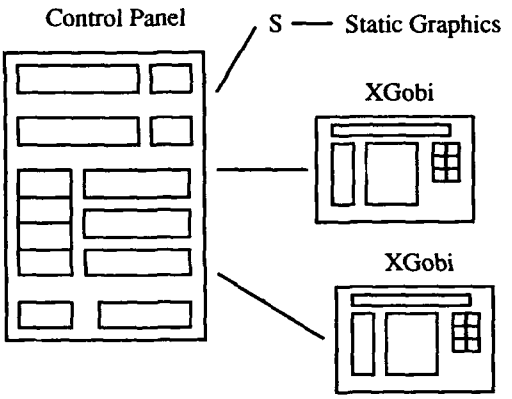


Figure 3: XClust Model

XClust is initiated from the UNIX command line. The XClust control panel appears, an S process is started, and an S graphics window appears. Figure 4 shows the XClust control panel and the S graphics window, as well as a Variable Selection Menu, which will be described later.

To start a clustering session with XClust, the user types in the name of the S data to be used, selects a button, and the functions `dist`, `hclust`, and `plclust` are applied to that data and the tree is plotted in the S graphics window. All the arguments to those functions can be adjusted using menus, buttons, or text windows. The user selects another button to initiate an XGobi window using the same data.

To define a clustering scheme based on this tree, the user can click on the S graphics window, specifying the height at which to cut the tree. This action has two results: a line is drawn on the S window indicating the height of the cut, and the vector of cluster membership is passed to XGobi, which then redraws each point using a different color and glyph for each cluster. The result of this action is shown in Figure 5.

To investigate the validity of this clustering scheme, the user can select another button called "xgobidiscr()." This action initiates a new XGobi window containing a plot of the data in the space of the discriminant coordinates. After examining the scatter plots and the discriminant coordinate plots, a user may decide that one or more variables are not contributing to the clustering among the data. The Variable Selection Menu allows these variables to be eliminated from the computation

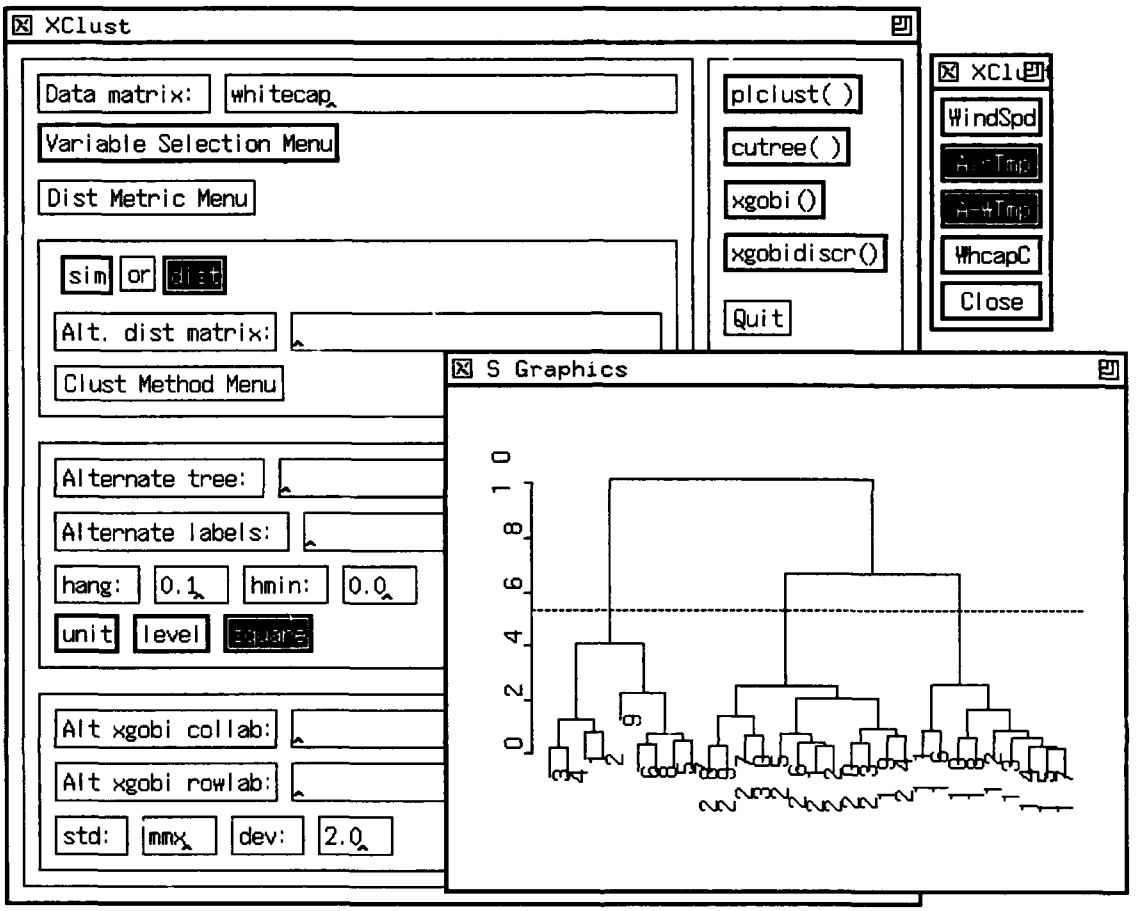


Figure 4: XClust: Control Panel, S Graphics Window, and Variable Selection Menu

of the distance matrix. If a change is made in that menu, a new tree is calculated and plotted.

An additional form of interconnection between program elements is used in XClust: instances of XGobi communicate with each other using an X interprocess communication method. Selecting a button in one XGobi window causes the color and glyph characteristics of each point to be sent to other linked XGobi windows. In this way, the points in the XGobi window which represents the data plotted in the space of the discriminant coordinates reflect the cluster identities shown in the first window.

4 Conclusions

In this work, we often encountered the question of when to write new software and when to use existing code. For example, we wrote the code for the static graphics window in XSmooth but used an S graphics window in XClust. When we write our own code, we have greater control over it. It would be easy to link the plotting window in XSmooth to an XGobi window, for instance, as XGobi windows are linked, and such a window could respond to mouse events in a very flexible way. On the other hand, when we use existing code, we have the benefit of previous authors' work. We saved time by using the S code for drawing a clustering tree, at the cost of some limitation on the user's ability to interact with that window.

In future work, we expect to encounter that same question again, in choosing each element of the integrated system. We will make the decision by balancing those two factors: the amount of work we expect to save by using existing code, and the amount of control we want over the element of the system.

We think this model has wide applicability, and we plan to work with it further. We would like to find out whether it can be made easy to program. There are

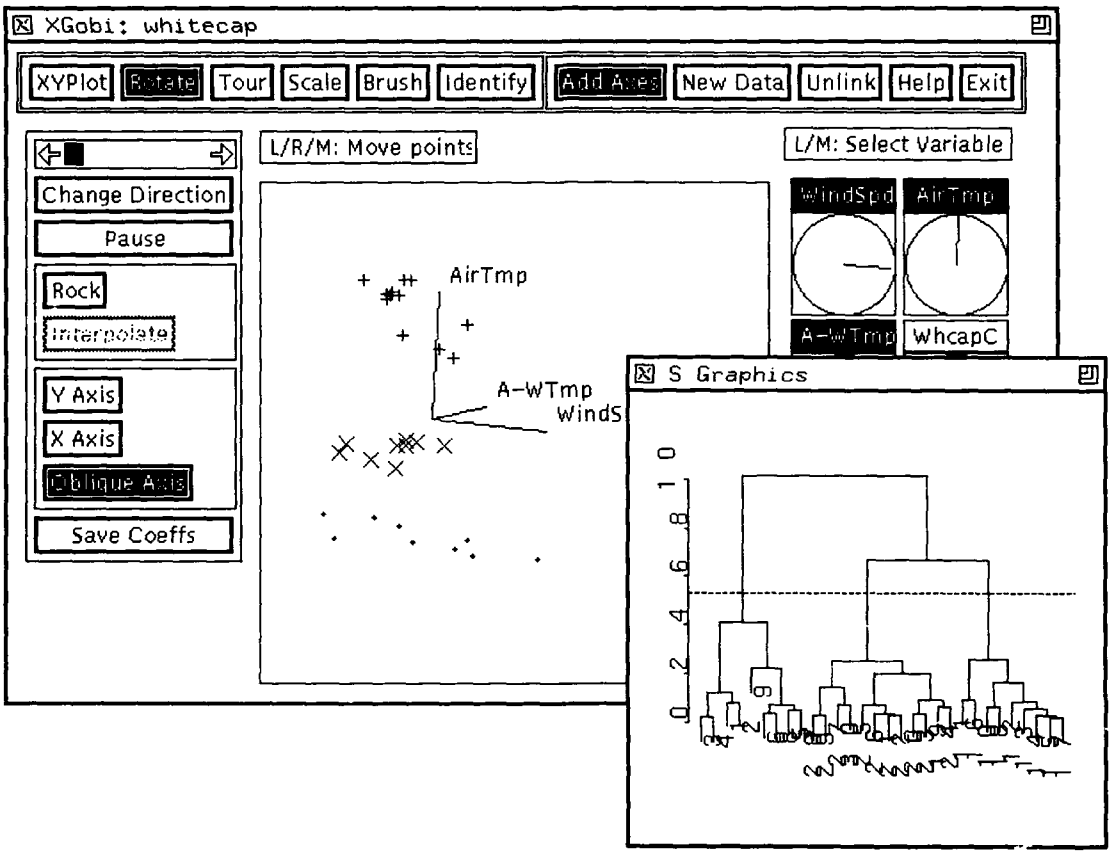


Figure 5: XClust: XGobi and S Graphics Window

tools that are intended to make it easy for users of the X Window System to create windows such as these control panels and to attach functionality to them. We want to find out whether these tools could be used in our context.

5 Acknowledgements

We would especially like to thank Daniel Nachbar, whose interprocess communication code we use in XSmooth. Allen Mcintosh and Tom Raleigh were also helpful in advancing our understanding of UNIX interprocess communication methods. We would like to thank Dianne Cook, Diane Duffy, Elaine Keramidas, Jon Kettenring, and Martin Maechler for useful comments as well.

References

Becker, Chambers, and Wilks (1988). *The New S Language: A Programming Environment for Data Analysis and Graphics.* Wadsworth & Brooks/Cole, Pacific Grove, California.

Cook, D., Buja, A., and Cabrera, J. (1991). Direction and Motion Control in the Grand Tour. In *Proceedings of the 23rd Symposium on the Interface.* Springer/Verlag.

Swayne, D. F. and Cook, D. (1990). Xgobi: A Dynamic Graphics Program Implemented in X with a Link to S. In *Proceedings of the 22nd Symposium on the Interface.* Springer/Verlag.

Swayne, D. F., Cook, D., and Buja, A. (1990). User's Manual for Xgobi, a Dynamic Graphics Program for Data Analysis Implemented in the X Window System. Bellcore Technical Memorandum.

Using Multiple Views for Data Analysis

Ron Baxter
Murray Cameron
Nicholas Fisher
Branka Hoffmann

CSIRO Division of Mathematics and Statistics
PO Box 218 Lindfield, NSW, 2070 Australia

Commonwealth Scientific and Industrial Research Organization

Abstract

When a data analyst meets a complex dataset, graphics displays giving overall summaries are examined first, then more specific displays that highlight observed features are studied. Frequently, this involves selection of subsets, and point-and click-methods are intuitive and effective. Sometimes the observed features are investigated by altering details of the analysis, and then an interactive command interface (like S) can be more useful.

A rainfall dataset with geographic and time components is used as an example. Graphics displays are done in a modified version of S that permits multiple graphics windows, and this is compared with xlispstat, xgobi, and datadesk.

1.0 Introduction

Five years ago the S Language offered two examples of dynamic graphics (brushing and spinning). These methods were supported on special graphics terminals that never achieved great popularity, but the concepts of these graphics techniques inspired many. Since then these ideas have been extended to more general notions of how linked views of data, and animation can be helpful to the data analyst. Much of this work has been done on the Macintosh. It is probably fair to say that these systems have not been widely used by data analysts, because the software systems provide too many restrictions.

We are interested in displaying multiple views on a workstation screen to

- select subsets interactively
- investigate relationships by highlighting
- setting of parameters interactively

We wish to explore how well these concepts of dynamic graphics and linked windows fit into a realistic working environment for data analysts. We have chosen an example dataset that does not fit the mould of either brush or spin, and is not trivially small.

2.0 The Rainfall Dataset

The dataset to be examined is monthly rainfall for 70 years (1913 - 1982) for 30 regions of the state of New South Wales. The Great Dividing Range is parallel to, and close to the coast, giving high rainfalls along the coast, and low rainfalls in the west. The north coast is sub-tropical with high summer rainfall, while in the south the mountain

range is higher (Snowy Mountains) and has a more prominent winter/spring rainfall. This is indicated in Figure 1.

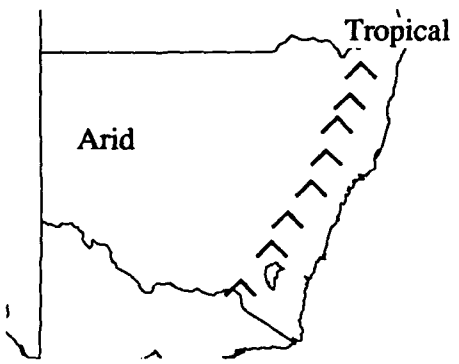


Figure 1

Our aim is to explore the data looking for patterns that may be interesting. Clearly, plotting the map will be useful, and comparing patterns for rainfall between regions will be one feature to explore.

3.0 DataDesk

Datadesk on the Macintosh has gone a long way with the concept of linked views. It can display scatterplots, lineplots, barcharts, piecharts, boxplots, and probability plots. Any number of these can be displayed with linkages between them.

However, it did not seem that we could plot a map of the regions, so we did not pursue DataDesk any further. Of course, we went out of our way to identify a dataset that did not fit the *brush/spin* mould.

4.0 Xgobi

Xgobi is similar. It has addressed the needs of brushing and spinning, and done a really excellent job of it. The user controls are well designed and operate smoothly. Color is used effectively, and re-scaling and rotation are beautiful to watch.

However, it does not appear that our requirement for a map of regions fits in at all.

5.0 Lisp Stat

5.1 Learning a New System

Learning any new system can be a hassle whether it be a new editor, a new word-processing system, or a new data analysis system. At the most elementary level it is the difference between

```
fun(x)
```

and

```
(fun x)
```

That is just the beginning. How do you list a function? What is a sensible operating environment for this system?

So given these likely problems, it came as a surprise to discover that Lisp-Stat is fun, and a challenge. This is largely because of the graphics that can be achieved, and is helped by the book which guides you between learning by doing, then absorbing new concepts. To give an example, by page 62 your are presented with an example of a dozen lines of code. This generates a scatterplot and a slider control (Figure 2) which sets the parameter of a Box-Cox power transformation. As you move the slider, the plot shows the effect.

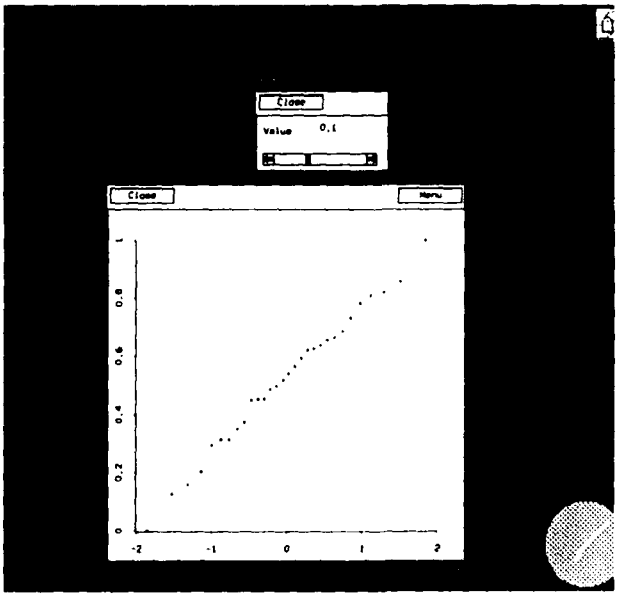


Figure 2

5.2 Using Lisp-Stat to look at rainfalls

Plotting the map of regions is quite straightforward. Once the data variables have been set up all that is needed is

```
; PLOT-MAP - draws a map of NSW:

(defun plot-map()
   (def plotmap (plot-points
      mgns-x mgns-y :title
      "Map of NSW" NIL))
   (send plotmap :x-axis nil nil 5)
   (send plotmap :y-axis nil nil 5)
   (send plotmap :clear-points)
   (send plotmap :size 380 270)
   (send plotmap :location 3 80)
   (dotimes (i 30)
      (send plotmap :add-lines
         (select reg-x i)
         (select reg-y i))
   )
   (send plotmap :add-points
      centres-x centres-y )
   (send plotmap :linked t)
)
```

Instead of trying to add a label to each region, Lisp-Stat makes it very easy to have the list of names as a linked window, so that as you point at regions, the corresponding names highlight (Figure 3). Finally Lisp-Stat offers a range of statistical functions that can be used to explore the data.

6.0 S-PLUS

6.1 The versions we used

We started this work using a version of New S (June 89 tape) that had been modified to permit multiple graphics windows simultaneously. We then received a beta copy of S-PLUS 3.0 which has this same facility. We did not use any other new facilities of S-PLUS 3.0 for these demonstrations.

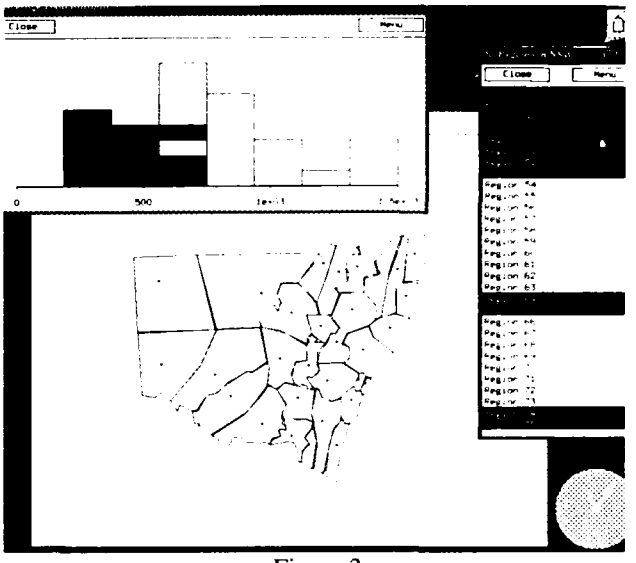


Figure 3

6.2 Locating Regions

The first demonstration has two windows, one showing a histogram of average total annual rainfalls for the regions, and the other showing a map of the regions. Then by pointing at any bar(s) of the histogram, the corresponding regions of the map are shaded in the same color as the histogram bar (Figure 4).

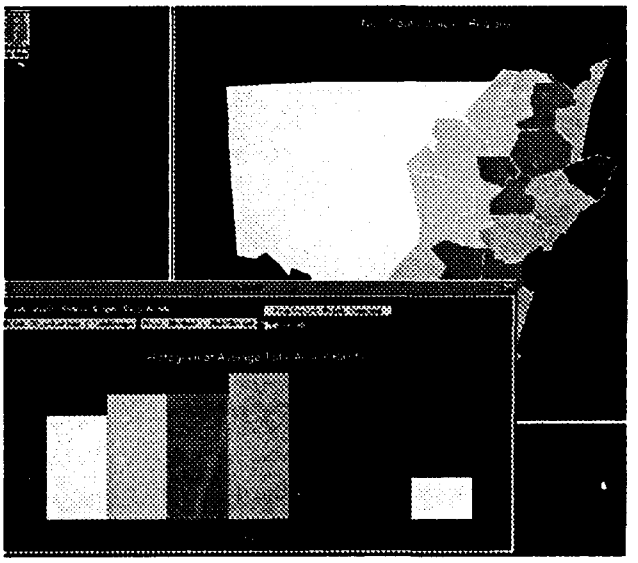


Figure 4

The skeleton of the S function to do this is:

```
demo1 <- function()
{
   .....
   X11()    # open first window
   histogram(...)
   ...
   repeat {
      loc.list <- locator(...)
      if (...) {# have selection
         if (first) {
            ...
            X11() #second window
            plot(..) #map
            ...
         }
         else ... # select 2nd
         for(...) # over selections
            for (...) # regions
               polygon(..)
         }
         else break # no selection
   }
   .... # finish up
}
```

This runs fast enough (on a Sun 4), and the main shortcoming is lack of visible feedback as the mouse is used. The user has to know that locator will be expecting input in window 1. While one can add S code to provide visual cues, this only partly solves the problem.

6.3 Pointing at Regions

The inverse operation is simple. First place a map on the screen, and then as the user points at regions, pop up a window showing some summary of the data for that region. So an argument to this function, is a function to produce a summary graph for the selected region. Obvious possibilities are the total annual rainfall for the 70 years, or the average monthly rainfall for the 12 months. Here we show the average monthly rainfalls (Figure 5).

6.4 Locating Correlations

Given that we can summarize each region by 12 monthly averages or 70 annual totals, we can look at correlations between regions. It is then useful to be able to relate given correlations to the map. In this example, the correlations are presented in a histogram, and their location on the map

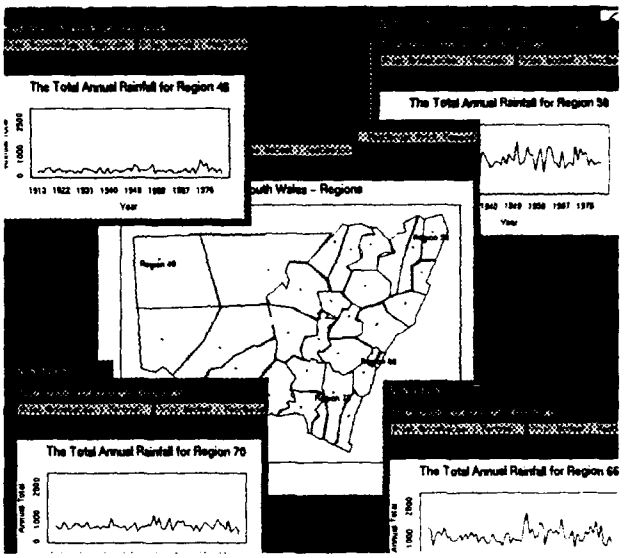


Figure 5

is shown by drawing a line between the two regions. We have used the monthly averages, and then focussed on the small number of correlations that are negative (Figure 6).

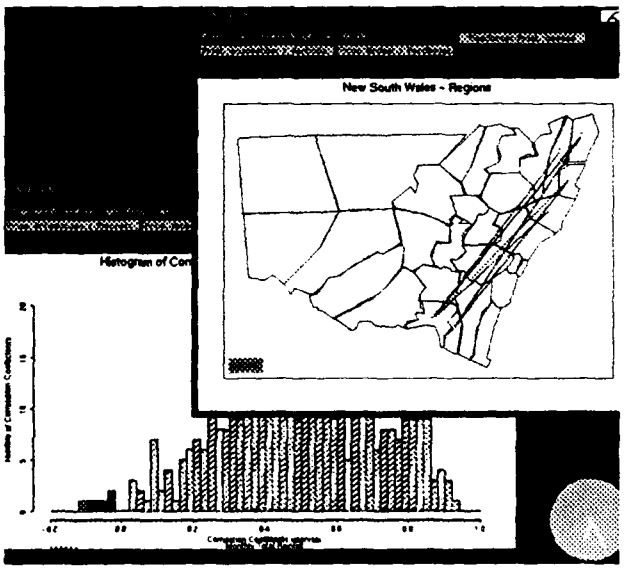


Figure 6

It turns out that all these negative correlations are between the sub-tropical north-east coast, and the Snowy Mountains. The next step is then to look at the monthly patterns for these regions, and the function provides for this.

The main point to be made from these examples, is that the big advantage of S is the flexibility in defining the graphics required, and the range of analyses that can be called on.

For a given region there are two plots of interest. The average monthly rainfall for each of the 12 months, and the pattern of wet/dry years, as shown by a plot of 70 total annual rainfalls. They are both shown here. The annual total rainfalls show a smoothed trend-line fitted using a cubic spline routine that was dynamically loaded. It would be tempting to provide a slider for the adjustment of the smoothing parameter for these trend-lines (Figure 7).

7.0 Conclusions

Having deliberately chosen a data-set that was not well served by *brushing* or *spinning* of scatterplots, it is not surprising that we did not make progress with less programmable tools like Datadesk and Xgobi.

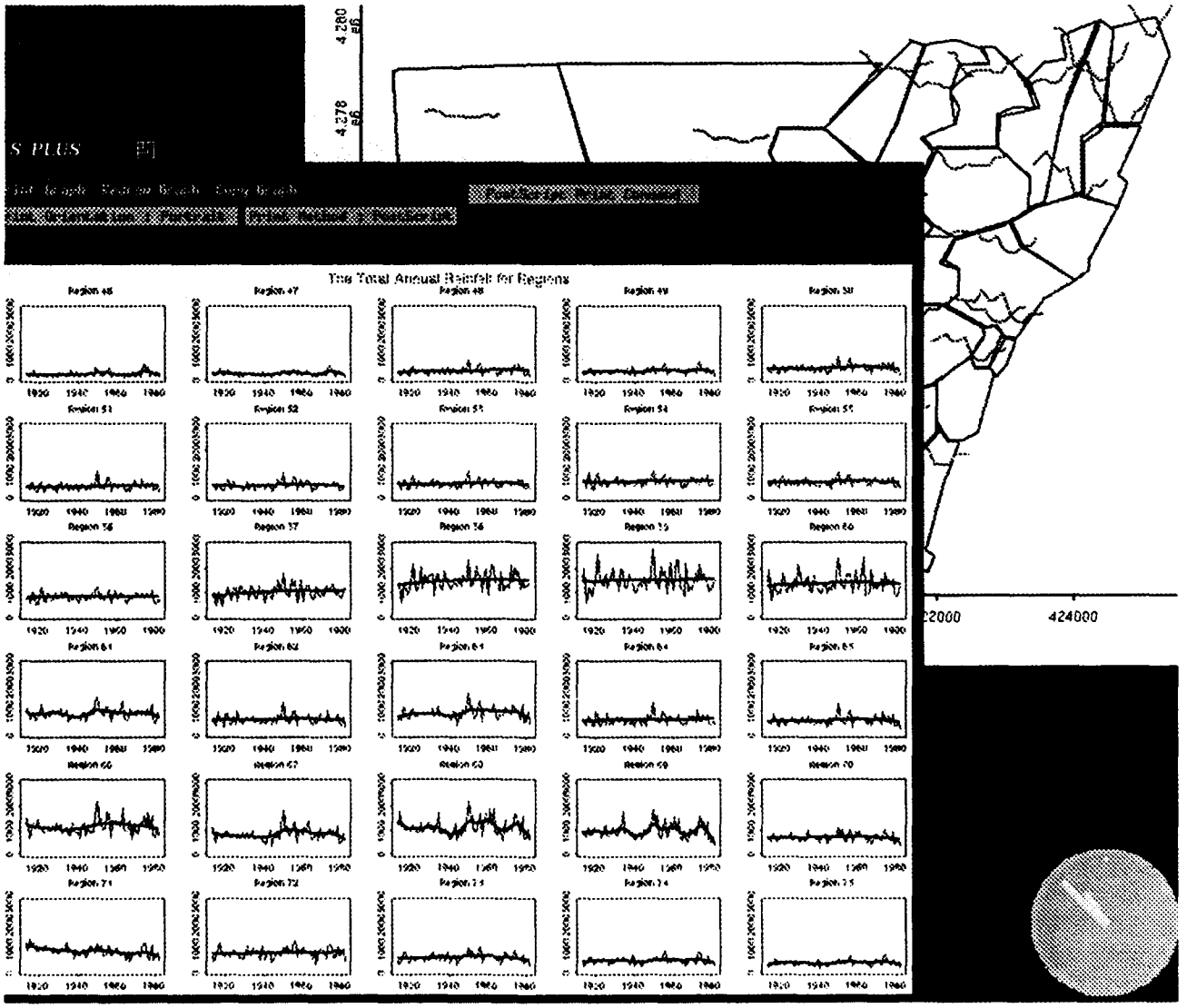


Figure 7

Lisp-Stat provides a good environment for implementing the linked views required. It is easy to program the graphs required, and the responsiveness of the system is good. However, the range of procedures available is limited, and at the next stage of the data analysis scenario, this may have become a real limitation.

S-PLUS, on the other hand has an extensive range of statistical functions, and it is likely that whatever else we would want to try could easily be done. The graphics are also flexible so that whatever style of display we require, it should be achievable. The multiple graphics windows make the multiple views possible. However the shortcoming is the degree of responsiveness to the mouse.

Since S-PLUS would be our preferred working environment, the ideal solution would be to have improved facilities for providing linked displays in this environment.

An Application of Subregion Adaptive Numerical Integration to a Bayesian Inference Problem *

Alan Genz
School of EE and CS
Washington State University
Pullman, WA 99164-2752

Robert E. Kass
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Abstract

Well-tested and available software for evaluating multidimensional integrals of moderate dimensionality may be adapted for use in Bayesian inference via elementary parameter transformations. We illustrate with an example from cognitive modeling of error rates in computer-based tasks, in which the parameter being integrated is six-dimensional and the integrand itself requires a product of twenty one-dimensional integrations for each function evaluation. This method appears competitive with, and may be superior to, alternative methods when the transformations are well chosen.

KEY WORDS: adaptive integration, hierarchical models, multiple integrals, posterior computation.

1 Introduction

Many Bayesian analysis computations require evaluation of multidimensional integrals in the form $\int h(\lambda)L(\lambda)\pi(\lambda)d\lambda$, where $L(\lambda)$ is the likelihood function, $\pi(\lambda)$ is the prior density, and λ is m-dimensional. Among integration problems generally, these are special because the likelihood function is often peaked near its maximum and thus locally of an approximately normal form. Integration strategies often try to take advantage of this special situation (e.g., Geweke, 1989; Naylor and Smith, 1982). Here we show how *subregion adaptive* numerical integration (Berntsen, Espelid, and Genz, 1991a,b) over the m-dimensional unit cube may be used following an elementary parameter transformation that accommodates the local form of the likelihood function. We apply the method to computations for a hierarchical model in which $m = 6$ but each evaluation of $L(\lambda)$ itself involves a product of twenty one-dimensional integrals. Although the results given are for only one specific example, the general approach described here should be useful in a variety of other applications.

*This work was supported in part by NSF Grant DMS-9008125.

2 Adaptive Integration

Subregion adaptive integration methods are based on the fundamental assumption that the integrand in the problem of interest can be accurately approximated locally by a low degree multivariate polynomial. The basic strategy for a subregion adaptive algorithm is to dynamically subdivide the initial integration region R into smaller and smaller subregions that are concentrated in the parts of R where the integrand is more irregular. The hope is that at some stage in this process the region R is sufficiently well partitioned that the combined integrated polynomial approximations for all of the subregions provide an accurate approximation to the initial integral. Typical input for this type of algorithm consists of (i) a description of the initial integration region R, (ii) the integrand, (iii) an error tolerance ϵ and (iv) a limit k_{max} on the total number of subregions allowed.

The adaptive algorithm itself also requires a basic integration rule (or formula) B and an associated error estimation rule E. We let B_i be the approximation to the integral in a subregion R_i obtained using the basic rule, and let E_i be the estimate for the absolute error in B_i. If at some stage in the algorithm R has been subdivided into k subregions, the relevant pieces of information are kept in a list $S = \{(R_1, B_1, E_1), (R_2, B_2, E_2), ..., (R_k, B_k, E_k)\}$. Initially we set $R_1 = R$ with $k = 1$ and compute B_1 and E_1. There are many possible adaptive strategies that may be used to dynamically refine the list S. The software (Berntsen, Espelid, and Genz, 1991b) that we have used for the tests described in Section 5 uses a globally adaptive algorithm. The main loop for this algorithm has

the general form:

while ($\sum_{i=1}^{k} E_i > \epsilon$ **and** $k < k_{max}$) **do**
a) determine j with $E_j = max_{i=1}^{k} E_i$
b) divide R_j into two pieces; $R_j = \hat{R}_j \cup R_{k+1}$
c) compute (B_j, E_j) and (B_{k+1}, E_{k+1})
d) set $k = k + 1$
end while

In step (b) the selected subregion is divided in half along the coordinate axis where the integrand is (locally) most rapidly changing (see Berntsen, Espelid, and Genz, 1991a for details). The output from the algorithm is an estimate $\sum_{i=1}^{k} B_i$ for the integral, and an error estimate $\sum_{i=1}^{k} E_i$. While software for this type of algorithm has not been widely used by statisticians, it has been available for several years. Software that uses a globally adaptive subdivision algorithm was available in the NAG library starting in 1980, and in CMLIB starting in 1985.

3 Transformations

In order to use a subregion adaptive algorithm we first need to apply transformations to the integration variables so that the region of integration becomes a hyper-rectangle. Many Bayesian analysis problems (including the problem that will be discussed in Section 4) have a prior density function that is a product of commonly-occurring one or two variable density functions, like the normal or gamma density functions. In this case an obvious choice for a prior transformation is simply to use the appropriate cumulative distribution function to transform each of the variables. For example, if one of the integration variables x has an associated factor $e^{-((x-\mu)/\sigma)^2/2}/(\sigma\sqrt{2\pi})$ in the prior with integration limits $-\infty$ and ∞, then the change of variable $x = \mu + \sigma\Phi^{-1}(z)$ with $\Phi(t) = \int_{-\infty}^{t} e^{-y^2/2} dy/\sqrt{2\pi}$ allows the removal of the associated exponential factor from the prior, and the z variable integration limits become 0 and 1. A sequence of transformations like this one can provide the hyper-rectangular domain of integration needed for a subregion adaptive algorithm.

Although this approach might seem limited to situations in which the prior is a product of distributions such as normal and gamma and for which there are available good numerical routines to evaluate the distribution function, a simple modification of this approach would be to base the transformations on elementary distributions chosen in some convenient way but without necessarily matching the marginal priors. This could require a lot of work from the user, however. Furthermore, when the algorithm is applied following this kind of transformation it may lose efficiency by spending large amounts of time finding the subregions in which the contributions to the integral are large. For the same reason, any kind of *a priori* specification of the transformation is likely to produce an integrand that is poorly suited to the adaptive algorithm. In order to obtain more efficient computation a transformation should force the algorithm to concentrate function evaluations where the integrand contributes substantially to the integral.

The general transformation method we use here is based on the assumption (Chen, 1985) that the posterior density function is approximately multivariate normal. In this case, $L(\lambda)\pi(\lambda) \doteq c \cdot e^{-(\lambda-\mu)^t \Sigma^{-1}(\lambda-\mu)/2}$ for some constant c, and optimization can be used with $log(L(\lambda)\pi(\lambda))$ to obtain the posterior mode μ and the modal covariance matrix Σ. If CC^t is the Cholesky decomposition of Σ then we may transform the original integral using $\lambda = \mu + Cy$, followed by inverse normal transformations on the individual y components to obtain

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h(\lambda)p(\lambda)d\lambda = |C| \int_{0}^{1} \cdots \int_{0}^{1} \bar{h}(\mathbf{z})g(\mathbf{y}(\mathbf{z}))\mathbf{dz},$$

where $g(\mathbf{y}) = (2\pi)^{\frac{n}{2}} e^{\mathbf{y}^t\mathbf{y}/2} p(\mu + C\mathbf{y})$, $\bar{h}(\mathbf{z}) = h(\mu + C\mathbf{y}(\mathbf{z}))$ and $\mathbf{y}(\mathbf{z}) = (\Phi^{-1}(z_1), ..., \Phi^{-1}(z_n))^t$.

This joint transformation is more straightforward to apply than one that would consist of separate transformations for each of the integration variables. In addition, as long as the stated assumption of approximate normality is correct, this method should be reasonably efficient because the transformed integrand should behave roughly as a piecewise low-order polynomial comprised of a comparatively small number of pieces.

4 An Example

An article by Carlin, Kass, Lerch and Huguenard (1990) considers two cognitive models for predicting error rates in computer-based tasks using the cognitive psychological concept of human working memory. Here we discuss in detail the computations for one of these. The more complicated model involves an integral

$$I(h) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(\lambda)L(\lambda)\pi(\lambda)d\lambda$$

with

$$\lambda = (\gamma, \mu_\theta^{(1)}, \mu_\theta^{(2)}, \sigma_\theta, \alpha, \beta)^t.$$

The prior $\pi(\lambda)$ is a product of normal densities, except for the σ_θ variable which contributes a factor $e^{-\sigma_\theta^{-2}/d}/\sigma_\theta^{2(c+1)}$. The likelihood function has the form

$$L(\lambda) = \prod_{i=1}^{2}\prod_{j=1}^{10}\int_{-\infty}^{\infty}\frac{e^{-\frac{1}{2}(\frac{\theta_j^{(i)}-\mu_\theta^{(i)}}{\sigma_\theta})^2}}{\sqrt{2\pi}\sigma_\theta}g(\theta_j^{(i)},\lambda)d\theta_j^{(i)},$$

with

$$g(\theta_j^{(i)},\lambda) = \prod_{k=1}^{3}\prod_{l=1}^{3}\frac{(e^{\theta_j^{(i)}+\gamma(k-1)+\alpha[l/2]+\beta[l/3]})^{z_{i,j,k,l}}}{(1+e^{\theta_j^{(i)}+\gamma(k-1)+\alpha[l/2]+\beta[l/3]})^{n_{i,j,k,l}}}.$$

Here, [x] is used to denote the integer part of x and the $z_{i,j,k,l}$ and $n_{i,j,k,l}$ values come from experimental data.

For prior transformations, we use appropriately chosen inverse normal transformations on all of the variables except for σ_θ. For this variable $\sigma_\theta = d'/x$, with $d' = (\frac{2}{d})^{\frac{1}{2}}$, followed by an inverse normal transformation gives (in one variable)

$$\int_0^\infty \frac{e^{-\frac{1}{d\sigma_\theta^2}}}{\sigma_\theta^{2c+2}}g(\sigma_\theta)d\sigma_\theta = \int_0^1 \Phi^{-1}(\frac{z+1}{2})^{2c}g(\frac{d'}{\Phi^{-1}(\frac{z+1}{2})})dz.$$

Using these transformations on the prior and an inverse normal transformation on each of the Likelihood inner integrals, $I(h)$ now becomes

$$I(h) = \int_0^1 \cdots \int_0^1 \bar{h}(\mathbf{z})\Phi^{-1}(\frac{z_4+1}{2})^{2c}\bar{L}(\mathbf{z})d\mathbf{z},$$

where

$$\bar{L}(\mathbf{z}) = \prod_{i=1}^{2}\prod_{j=1}^{10}\int_0^1 g(\theta_j^{(i)}(s_{i,j}),\lambda(\mathbf{z}))ds_{i,j},$$

$\bar{h}(\mathbf{z}) = h(\lambda(\mathbf{z}))$ and $\theta_j^{(i)}(s_{i,j}) = \mu_\theta^{(i)} + \sigma_\theta\Phi^{-1}(s_{i,j})$.

The modal approximation transformation can be used almost directly with the integral I. The only difficulty occurs with the variable σ_θ, which has limits 0 and ∞. We use an initial transformation $\sigma_\theta' = log(\sigma_\theta)$, and then use optimization to obtain the mode $\hat{\lambda}$ and modal covariance matrix Σ with Cholesky factor C.

The integral $I(h)$ can then be put into the form

$$I(h) = (2\pi)^3|C|\int_0^1 \cdots \int_0^1 h(\bar{\lambda}(\mathbf{z}))p(\bar{\lambda}(\mathbf{z}))d\mathbf{z},$$

with

$$p(\bar{\lambda}(\mathbf{z})) = e^{w_4+\frac{\mathbf{y}(\mathbf{z})^t\mathbf{y}(\mathbf{z})}{2}}h(\bar{\lambda}(\mathbf{z}))L(\bar{\lambda}(\mathbf{z}))\pi(\bar{\lambda}(\mathbf{z})),$$

where $\bar{\lambda}(\mathbf{z}) = (w_1, w_2, w_3, e^{w_4}, w_5, w_6)^t$ is defined using $\mathbf{w} = \hat{\lambda} + C\mathbf{y}(\mathbf{z})$ with $\mathbf{y}(\mathbf{z}) = (\Phi^{-1}(z_1), \ldots, \Phi^{-1}(z_n))^t$.

5 Computations

The integral calculations all require a six-dimensional outer integral of a function that requires a product of twenty one-dimensional inner integrals. The inner integrals were all computed with a simple subregion adaptive one-dimensional quadrature algorithm. This algorithm is similar to the algorithm used by the QUADPACK (Piessens, deDoncker-Kapenga, and Kahaner, 1983) subroutine QAG, with a 7-15 point Gauss-Kronrod pair chosen for the basic integration rule. The outer integrals were computed with a subregion adaptive m-dimensional algorithm using the SCUHRE (Berntsen, Espelid and Genz, 1991a,b) subroutine for vectors of integrals.

One problem with the computation of the inner integrals was how to set the required level of accuracy. Since the computation of all of these inner integrals is what takes most of the time in the calculations, it was necessary to choose this parameter with some care. The results in Tables 1 and 2 used a relative error tolerance for each inner integral of 10^{-4}. A significantly smaller value for the error tolerance significantly increased the computation time without significantly changing the results; a much larger value decreased the computation time but changed the results significantly.

A second problem with the inner integrals involved scaling. Some experimentation showed that these inner integrals had values that were typically about 10^{-6}. Because a product of twenty of these could cause underflow, we computed $log(L(\lambda))$ using a sum of the logs of the inner integral factors. This sum was initialized to the value 210, and the likelihood value was obtained by exponentiating the final sum. The effect of this intialization was to scale the integral value by e^{210}, but since the numbers of interest all require a division by $I(1)$, these scale factors cancel.

In Tables 1 and 2 below, the results are given for both types of transformations. The adaptive integration software computed the vector of integrals $I(h)$ for $h = 1, \mu_\theta^{(1)}, \mu_\theta^{(2)}, log(\gamma), \sigma_\theta, \alpha, \beta$, and then scaled the results by $I(1)$ to obtain the required expected values.

Since the major part of the computation is the computation of the likelihood products, the time was reduced by an approximate factor of 1/7 by the simultaneous computation of all the integrals (i.e., for all 7 choices of h). The prior transformation results required approximately four hours of single precision computation time on a DECstation 3100 (14 mips). The modal transformation results required approximately twenty minutes. In both tables, the numbers in the rows labelled "$L(\lambda)$'s" are the numbers of evaluations of $L(\lambda)\pi(\lambda)$. Several columns of re-

sults are given to illustrate the speed of convergence, and allow some estimation of the accuracy in the results.

Table 1: Prior Transformation Results

$L(\lambda)$'s	5957	11753	23989	47817
$\bar{\gamma}$	1.068	1.065	1.068	1.064
$\bar{\mu}_\theta^{(1)}$	-4.328	-4.322	-4.320	-4.314
$\bar{\mu}_\theta^{(2)}$	-4.961	-4.964	-4.946	-4.939
$\overline{log(\sigma)}$	0.178	0.183	0.184	0.181
$\bar{\alpha}$	1.397	1.391	1.396	1.395
$\bar{\beta}$	0.796	0.789	0.794	0.793

Table 2: Modal Transformation Results

$L(\lambda)$'s	483	1449	3059
$\bar{\gamma}$	1.062	1.062	1.062
$\bar{\mu}_\theta^{(1)}$	-4.311	-4.310	-4.310
$\bar{\mu}_\theta^{(2)}$	-4.930	-4.935	-4.936
$\overline{log(\sigma)}$	0.174	0.175	0.176
$\bar{\alpha}$	1.394	1.394	1.394
$\bar{\beta}$	0.793	0.793	0.793

It is clear that the modal transformation results in the first column of Table 2 are accurate to 2-3 digits, but the prior transformation results do not have this level of accuracy until the last column. In this example we can see that the modal transformation method takes about 1/100 of the time taken by the prior transformation method to achieve a comparable level of accuracy. The modal transformation results in the first column of Table 2 were actually obtained by the adaptive algorithm using only two subregions. The basic integration rule used here has degree seven, so that the transformed integrand apparently has a good local degree seven polynomial approximation. One problem that we had with the subregion adaptive software was with the error estimates. The estimates provided by the software for the relative errors in the final column Table 2 results were approximately 0.1, while the actual results apparently have much smaller errors. This problem is not uncommon with this type software, where the error estimates are usually very conservative. The usual solution to this problem is to take the approach that we have taken, and that is to estimate accuracy by looking at the level of agreement between results from finer and finer subdivisions.

6 Concluding Remarks

The reported results demonstrate the potential of subregion adaptive integration for solution of numerical integration problems in Bayesian analysis. The results also demonstrate the importance of choosing a good transformation to precondition the problem before a numerical integration method is used. The prior transformations that were used to obtain the less accurate results are transformations that might naturally be chosen by a numerical analyst, without a deeper knowledge of the expected approximate multivariate normal structure for the complete integrand. On the other hand, subregion adaptive integration methods are not widely used by statisticians. The relatively small number of function evaluations we found to be required when using the modal transformation makes us optimistic that, together with this kind of modification, subregion adaptive integration could prove to be competitive with, or superior to, available alternatives for solving similar numerical integration problems.

References

Carlin, B.P., Kass, R.E., Lerch, F.J. and Huguenard, B.R. (1990) Predicting working memory failure: a subjective Bayesian approach to model selection, Technical Report No. 503, Department of Statistics, Carnegie Mellon University.

Berntsen, J., Espelid, T.O. and Genz, A. (1991a) An adaptive algorithm for the approximate calculation of multiple integrals, *ACM Trans. Math. Soft.*, to appear.

Berntsen, J., Espelid, T.O. and Genz, A. (1991b) An adaptive multiple integration routine for a vector of integrals, *ACM Trans. Math. Soft.*, to appear.

Chen, C.F. (1985) On asymptotic normality of limiting density functions with Bayesian implications. *J. Royal Statist. Soc.*, **47**, 540-546.

Geweke, J. (1989) Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, **57**, 1317-1339.

Naylor, J.C. and Smith, A.F.M. (1982) Applications of a method for the the efficient computation of posterior distributions. *Applied Statist.*, **31**, 214-225.

Piessens, R., deDoncker-Kapenga, E., Uberhuber, C.W. and Kahaner, D.K. (1983) *QUADPACK*, Springer-Verlag, Heidelberg.

APPROXIMATIONS OF THE NORMAL-LOGISTIC CONVOLUTION INTEGRAL

John F. Monahan and Leonard A. Stefanski
Department of Statistics
North Carolina State University, Raleigh, NC 27695 - 8203

Abstract

The normal-logistic convolution arises in several statistical applications, including logistic regression models and multinomial logit models. We begin by characterizing the logistic distribution as a scale mixture of normals. We then construct least maximum approximations of the logistic distribution function using finite discrete mixtures of normal df's using the Remes algorithm. The convolution integral follows by convolving this approximation with the normal.

1. Introduction

Logistic regression has become one of the most popular methods of analyzing experiments with discrete or binary outcomes. The most common situation models the survival probability of a patient given a specified dosage, that is

$$\begin{aligned}&\Pr(\text{ patient survives } | \text{ receives dose } x \text{ })\\ &\quad = \Pr(\ Y=1 \mid X = x\ )\\ &\quad = F(\beta^T x)\end{aligned}$$

where $F(t)$ is the logistic distribution function $F(t) = (1 + e^{-t})^{-1} = e^t / (1 + e^t)$ and β is a vector of unknown regression coefficients to be estimated. The normal-logistic convolution, $G(\eta,\tau)$ defined by

$$\begin{aligned}G(\eta,\tau) &= \int F(t)\, \tau^{-1}\phi\left(\tfrac{t-\eta}{\tau}\right) dt\\ &= \Pr(\text{Logistic RV} \le t,\ t \sim \text{Normal}(\eta, \tau^2))\end{aligned}$$

arises in three related situations: measurement errors, random effects, and forecasts.

In the measurement error model, we observe Z and not X and the distribution of $X \mid Z = z$ is expressed as Normal($\mu(z)$, $\Omega(z)$). Then we have the observed outcome probability structure

$$\Pr(Y=1|Z=z) = G(\eta = \beta^T\mu(z),\ \tau = \sqrt{\beta^T\Omega(z)\ \beta}\ ).$$

In the random effects model, most commonly random litter effects in animal experiments, we have the random effect ϵ_i for litter i, and subjects j within the litter. The survival probability for the subjects in litter i (with covariates X_i) is then given by

$$\Pr(\ Y_{ij}= 1 \mid X_i,\ \epsilon_i\ ) = F(\ \beta^T X_i + \epsilon_i\ )$$

so that the expected number surviving in litter i is

$$E(\ Y_{i.} \mid X_i= x_i) = n_i\ G(\ \beta^T x_i,\ \tau\ )$$

where τ is the standard deviation of the random litter effect. In the third situation, following a Bayesian argument, one may from the logististic regression analysis of an experiment have an approximate posterior distribution for the regression coefficient vector β that is multivariate normal,

$$\beta \approx \text{Normal}(\ \hat{\beta},\ J^{-1}).$$

To construct the forecast probability of survival a patient with covariates $X = x$, one must compute the convolution integral in the form

$$\Pr(Y = 1 \mid X = x\ ) = G(\ \hat{\beta}^T x,\ \sqrt{x^T J^{-1} x}\ ).$$

In all three applications, the convolution integral $G(\eta,\tau)$ will be computed often, and fast, accurate caluculations are required.

2. Key Result and Implications

Stefanski (1990) proved that the logistic distribution can be expressed as a scale mixture of normals, that is $F(t) = \int \Phi(ts)\, q(s)\, ds$ where $\Phi(.)$ is the standard normal distribution function. The mixing distribution q(s) is, strangely enough, related to the Kolmogorov-Smirnov distribution, but that is not important here. The following calculations show the implications of this result to the computation of $G(\eta, \tau)$:

$$G(\eta, \tau) = \int F(t)\, \tau^{-1}\phi((t-\eta)/\tau)\, dt$$
$$= \int \left\{ \int \Phi(ts)\, \tau^{-1}\phi((t-\eta)/\tau)\, dt \right\} dQ(s)$$

where the expression in braces can be expressed in probabilistic terms for simplification

$$\left\{ \right\} = \Pr(Z \le ts \text{ where } Z \sim N(0, 1) \text{ and } t \sim N(\eta, \tau^2))$$
$$= \Pr(Z - ts \le 0)$$
$$= \Phi(\eta s / \sqrt{1 + s^2\tau^2}).$$

The convolution integral G can then be evaluated as

$$G(\eta, \tau) = \int \Phi(\eta s / \sqrt{1 + s^2\tau^2})\, dQ(s).$$

For computation, the integration with respect to the mixing distribution dQ(s) is approximated by a k point discrete distribution $Q_k(s)$ with masses p_j at s_j for j = 1,..., k. The approximations $G_k(\eta, \tau)$ now only require evaluations of ERF/ERFC :

$$G_k(\eta, \tau) = \sum_{j=1}^{k} p_j\, \Phi(\eta s_j / \sqrt{1 + \tau^2 s_j^2}).$$

3. Least Maximum Approximations

The expression for $G_k(\eta, \tau)$ shows only the premise of a computational method for computing the convolution integral G, since the discrete approximation Q_k in terms of $\{ p_j, s_j, j=1,...,k \}$ must be determined. While the ultimate goal is to make the error $|G - G_k|$ small, achieving this over both parameters η and τ is quite difficult. Making the error for the mixing distribution $|Q - Q_k|$ small can provide a bound for the error $|G - G_k|$, but choosing the best here cannot guarantee small error in G_k. Our approach has been to find the best $\{ p_j, s_j, j=1,...,k \}$ to minimize the maximum over t of $|F(t) - F_k(t)|$, where $F_k(t) = \sum_{j=1}^{k} p_j\, \Phi(t\, s_j)$ the approximation to the logistic df. An accurate least maximum or "Chebyshev" approximation to F will then lead to an accurate approximation to G.

Least maximum approximations are the staple of function approximations for computer evaluations (Hart, et al., 1968). However, usually the approximant is a polynomial or a rational function. Taking the polynomial case for illustration, let $H_p(x) = \sum_{j=0}^{p} a_j x^j$ be approximating the function H(x). Then, according to Chebyshev's Theorem, the least maximum approximation will leave the difference function $d(x) = H(x) - H_p(x)$ with extrema that alternate in sign and achieve equal magnitudes. With a polynomial $H_p(x)$ of degree p, with p+1 parameters, there will be p+2 extrema for the least maximum approximation. The algorithm commonly used for finding such an approximation, the Remes Algorithm, follows these steps:

Remes Algorithm

a) Find p+1 roots $\{ z_i \}$ of the difference function d(x)

b) Find p+2 extrema $\{ x_i \}$

c) Solve the linear equations in $\{ a_j, j = 0,....p \}$, D
$d(x_i) = H(x_i) - H_p(x_i) = (-1)^{i+1} D$
for i = 1, ..., p+2

d) Repeat

In our situation, the approximant $F_k(t)$ is not a polynomial, but a complicated function $F_k(t) = \sum_{j=1}^{k} p_j \Phi(ts_j)$ with 2k-1 parameters ($\sum p_j = 1$). If an analogous approach can be successful here, we would hope to find 2k-1 roots of the difference function $D_k(t) = F(t) - F_k(t)$ and k extrema alternating in sign and equal in magnitude. Some special considerations include symmetry about 0, since $D_k(-t) = - D_k(t)$, and $D(0) = 0$ because $F(0) = 1/2 = \sum p_j \Phi(0 s_j)$. Moreover, since $F(t) \to 0$ slower than $\Phi(st)$ for large t, then $D_k(t)$ will be negative for large t. Also, $D_k(t)$ will have a positive slope at the origin. These considerations lead to an even number of extrema, which luckily matches the odd number of parameters 2k-1. Our algorithm for finding the least maximum approximation differs from the Remes algorithm only in that we solve a system of nonlinear equations in $\{ p_j, s_j\}$ and D of the form $D_k(x_i) = (-1)^{i+1} D$.

In spite of these obstacles we have been able to find approximants $F_k(t)$ whose difference function has extrema that alternate in sign and have equal magnitudes, and we suppose, but have not proven, that the approximants are the least maximum approximations. For small values of k, we could obtain starting values by trial and error. But for larger values of k, we found starting values by successive nonlinear regression, minimizing over $\{ p_j, s_j\}$ the weighted sum of squares

$$\sum w_i [F(x_i) - F_k(x_i)]^2$$

By taking $w_i = [F(x_i) - F_k(x_i)]^{2r}$ we are able to minimize the 2r+2 norm, and by increasing r, approach the Chebyshev solution. We were able to achieve high accuracy for these approximations. Table 1 gives the values of $D_k^* = \sup | D_k(t) |$ for k = 1, ..., 8.

While the original problem was the approximation of $G(\eta, \tau)$ by $G_k(\eta, \tau)$, an accurate solution to the approximation of F(t) by $F_k(t)$ was hoped to lead to an accurate $G_k(\eta, \tau)$. We have found that since our approximations $G_k(\eta, \tau)$ improve with increasing τ, the error D_k^* also bounds the error $| G - G_k|$ for all η, τ. Other approximations based on Taylor-like expansions, quickly fail when τ increases. While the Crouch-Spiegelman method can achieve any level of accuracy, their approach requires many more evaluations and its use is not appropriate for the applications mentioned here.

4. References

E. Crouch and D. Spiegelman (1990) The evaluation of integrals of the form $\int f(t) \exp\{-t^2\}dt$. Application to logistic-normal models. Journal of the American Statistical Association, Volume 85, pp. 464-469.

John F. Hart, et al. (1968) Computer Approximations, Wiley, New York.

Leonard A. Stefanski (1990) A Normal Scale Representation of the Logistic Distribution, Statistics & Probability Letters, Volume 11, pp. 69-70.

Table 1
Accuracy of Approximations

k	D_k^*	k	D_k^*
1	9.5(-3)	5	6.0(-7)
2	5.1(-4)	6	8.4(-8)
3	4.4(-5)	7	1.3(-8)
4	4.7(-6)	8	2.1(-9)

AUTOMATIC DETECTION AND TREATMENT OF SINGULAR INTEGRALS

Chaiho C. Wang
Antitrust Division, U.S. Department of Justice
Washington, DC 20001

Abstract

Many numerical integration algorithms can handle singular integrals effectively if they are told the locations of the singularities. This paper gives algorithms for (1) prescreening the integrand for the locations of singularities, and (2) circumventing compiler-imposed accuracy restrictions. In application, the algorithms are used together with a core Romberg type algorithm. While transformations may be needed to handle "pathological" integrals that converge very slowly, most integrals, including nearly all probability integrals, can be evaluated quickly and accurately.

Introduction

Semi-automatic numerical algorithms are available allowing the user to perform integration by simply specify the form of the integrand, the limits of integration and the accuracy requirement. Some algorithms also allow evaluation of singular integrals by specifying, in addition, the locations of singular points of the integrand. This paper presents an algorithm for detecting the location of singularities. Once the locations are known, a semi-automatic procedure can evaluate the integrals accordingly.

A true automatic integrator, like a robot driven automobile in the streets of New York City, has not yet been perfected. Semi-automatic integrators, which require a minimal amount of human steering, are available. A number of automatic integrators were presented in Davis and Rabinowitz [3]. The three qualities required of an automatic integrator, as defined by Davis and Rabinowitz, are efficiency, reliability, and robustness. The automatic handling, of course, defines the fourth quality a non-automatic integrator does not possess: convenience. As micro-computers become increasingly powerful and computer time less costly, other than in real-time applications, the importance of efficiency is fast diminishing. Since reliability often depends on efficiency and computing power, its prominence is also declining. A little muscle-flexing of computer power would yield more accurate results. Part of the reliability requirement also goes together with robustness, since the integrator is required to handle a broad range of integrals and also be predictable. A key requirement for an automatic integrator is the ability to handle singularities. Otherwise, an integrator can be automatic in all other applications but will break down when encountering a singular integral.

The Core Integrator

The core integrator is an extended Romberg (RE) algorithm given by Wang [6]. The merits of classic Romberg are fully explored in Bauer, Rutishauser and Stiefel [1]. The method is valid for all Riemann integrals (it is therefore robust). It converges rapidly (and is therefore efficient). Finally, it is predictably accurate; (i.e. reliable). The performance of the classic Romberg method, however, depends on the asymptotic behavior of the integrand. If the integrand converges slowly somewhere on the interval of integration, such as in the neighborhood of a singular point, however, the accuracy may be unsatisfactory. The RE method treats the range of integration "dynamically," adjusting to the asymptotic behavior of the integrand. Essentially, we have an integral

$$I=\int_a^b f(x)\,dx$$

such that

$$I=I_{a_0}^{a_1}+I_{a_1}^{a_2}+\ldots+I_{a_{n-1}}^{a_n}$$

where $a_0 = a$, $a_n = b$, and

$$I_{a_j}^{a_{j+1}}=\int_{a_j}^{a_{j+1}} f(x)\,dx$$

The additive decomposition is valid provided that the conditions for the Romberg method--the integral is Riemann and bounded--are satisfied in each interval

(a_j, a_{j+1}). Mathematically, if the integral is Riemann in (a, b), it should be Reimann in (a_j, a_{j+1}) for all j. In practice, the classic Romberg method operating in (a, b) may not get close enough to the singularity points to cause problems, even though it would not yield very accurate results. But in a narrow neighborhood, such as in one of the sub-intervals (a_j, a_{j+1}), a singular point would be more likely to cause computational difficulties.

The RE algorithm has been tested successfully for evaluating internals with end-point singularities (as well as improper integrals). Evans, Hyslop and Morgan [4] gave many test integrals including 12 from Chisholm, Genz and Rowlands [2], 9 from Harris and Evens [5], and 10 of their own. The various Gaussian-type methods used and discussed in Evans, *et al.*, have achieved accuracy from 3 to 10 digits. In particular, the ϵ-Patterson procedure used by Evans, *et al.*, have consistently reached 10-digit accuracy. The RE procedure had little trouble getting 10-digit accuracy for all but two oscillatory (trigonometric) integrals. More precisely, RE obtained 7-digit accuracy for the oscillatory integrals, and 15-digit accuracy for all other integral. With the exception of a pathologically slowly converging integral

$$I_1=\int_0^1 x^{10^{-6}-1}dx=10^6,$$

no functional transformation of the integrand was required. For I_1, after an exponential transformation, RE obtained the exact value for the integral.

Detecting Singularity

In order to move closer to an automatic setting, we present a simple algorithm for detecting singularity of the integrand. The algorithm is based on a search process. The idea is to find a neighborhood of a singular point where the absolute value of the function is large, and either the function or its derivative changes sign.

1. Let $h=(b-a)/n$ for suitable n.

2. Select $a_* < a-2h$, and $b_* > b+2h$, such that neither $a-a_*$ or $b-b_*$ is a multiple of h.

3. Initialize.

Compute $y_0 = f(a_*+h)$, $Dy_0 = (f(a_*+h) - f(a_*))/h$

4. Searching.

In the interval (a_*, b_*), for $j=1,\ldots,n$,

compute $y_j = f(a+(j+1)h)$, and
$Dy_j = (f(a+(j+1)h)-f(a+jh))/h$

5. Zero in.

If $y_jy_{j-1} < 0$ or $Dy_jDy_{j-1} < 0$,

Readjust the intervals:

$a=a+jh$, $b=a+(j+1)h$, and go back to step 1.

6. Repeat the steps 1-5, until some point x_s ($y=f(x_s)$) where $|y|$ is sufficiently large to resemble infinity.

Declare x_s a singularity point.

The Under- and Over-flowing Problem

In order for the algorithm to work, we must work within and around the limit of the computation environment. The arithmetic processor sets a boundary around what it recognizes as a valid real number. For example, a given Fortran compiler may recognize a (double precision) real number as one in the interval $(-2*10^{308}, 2*10^{308})$. A zero, therefore, can retain an accuracy close to 10^{-308}. In order to identify a singularity before we are blown out by an overflowing or underflowing problem, we need to compute using those very "large" or "small" numbers. A relatively easy way to circumvent the boundary problems is to retool all arithmetic operations and intrinsic functions, including multiplication, division, exponentiation, logarithmic and trigonometric functions. For example, the division x/y can be rewritten as x div y, which does everything x/y would do except when y is "zero", (say, less than, 10^{-307}), at which point it will stop and give a message, without having to discontinue the program. The ordinary division x/y, would in this case have killed the program.

The Significant-Digit Barrier

The second limitation the arithmetic processor sets a limit on accuracy. For example, a given Fortran compiler may allow a double precision arithmetic to carry 15 to 16 significant digits before truncation or rounding takes place. Although the apparent range of valid numbers is $(-2*10^{308}, 2*10^{308})$, it is full of enormous holes. All numbers that must be represented by 17 or more digits would fall into those holes. Consider the integral

$$I=\int_0^1 f(x)\,dx$$

with singularities at both end points. The lower limit, 0, as approximated by 10^{-308} and is accurate to the 308th digit. But the upper limit, 1, if inexact, cannot be represented from below by anything better than 0.9999999999999999. As a consequence, a chunk of I, namely the integral

$$I=\int_{0.9999999999999999}^1 f(x)\,dx$$

cannot be captured. It becomes a part of the error.

For example, the value of π can be represented by the integral

$$\pi=I_2=\int_0^1 t^{-\frac{1}{2}}(1-t)^{-\frac{1}{2}}\,dt$$

treated as an integral with singularities at both ends, or by the integrals

$$I_3=2\int_0^{1/2} x^{-\frac{1}{2}}(1-x)^{-\frac{1}{2}}\,dx$$

and

$$I_4=2\int_{1/2}^1 x^{-\frac{1}{2}}(1-x)^{-\frac{1}{2}}\,dx$$

each treated as an integral with a singularity at one of the two ends. For I_3, the integrand is singular at 0; the RE procedure returned a value of 3.141592653589793, accurate to 16 digits. For I_2 or I_4, the integrand is singular at 1. In both cases, the result is 3.141589..., accurate to five digits.

Practically, there is little we can do to resolve this problem. For certain integrals, but not in general, a transformation of the integrand can shift a singularity away from a non-zero point to zero. Then things become more manageable.

References

1. Bauer, F. L., Rutishauser, H. and Stiefel, E. (1963). New Aspects in Numerical Integration, *Proceedings of Symposia in Applied Mathematics*, Vol. XV, Am. Math. Soc.

2. Chisholm, J. S. R., Genz, A. and Rowlands, G. E. (1973). Accelerated Convergence of Sequences of Quadrature Approximations, *J. Comp. Physics*, 10, 284-307.

3. Davis, P. J. and Rabinowitz, P. (1984). *Numerical Integration*, Blaisdell, Waltham, Massachusetts.

4. Evan, G. A., Haslop J., and Morgan, A. P. G. (1983). An Extrapolation Procedure for the Evaluation of Singular Integrals, *Intern. J. Computer Maths.* 12, 251-265.

5. Harris, C. G. and Evans, W. A. B. (1977). Extension of Numerical Quadrature Formulae to Cater for End Point Singular Behavior over Finite Intervals, *J. Computer Math.* 6, 219-227.

6. Wang, C. C. (1990). A Bread-and-Butter Algorithm for Probability Integrals, *Proceedings of the 22nd Symposium on the Interface of Computing Science and Statistics*, in press.

Note

The views expressed in this paper do not necessarily reflect those of the U.S. Department of Justice.

Computation of the Multinomial Distribution Function

Trong Wu
Department of Computer Science
Southern Illinois University at Edwardsville
Edwardsville, Illinois 62026

Abstract

The computation of the *multinomial distribution function* is of interest to many researchers and practicers who are working in the areas of engineering and in the related disciplines of computing sciences. The accurate computation of probabilities is very important in some applied areas. A direct computation is not only difficult, due to the limitation of the computer systems, but also inaccurate, as a result of many redundant computations. This research is to develop an effective method to compute the weight probabilities for the multinomial distribution function *accurately* and *efficiently*.

1. Introduction

Accurate probabilities of the multinomial distribution function are very important in many theoretical and applied areas related to engineering and computing sciences. However, the direct computation is still considered difficult due to the computation of factorials and decimal numbers; the limitations of computer systems such as overflow, underflow, and the maximum accuracy; the programming techniques such as writing a reliable and efficient program; and the time consumed by computing.

As a result, currently, there are no software packages available for the computation of the multinomial distribution function, such as IMSL Library [5], or an effective algorithm for dealing with computations. This paper presents an effective method to compute the probabilities for the multinomial distribution function accurately and efficiently. Section 2 diccusses some important applications of the multinomial distribution function. Section 3 presents the new method and its mathematical foundations. Finally, some computational examples are given in Section 4 and conclusions in Section 5.

2. Applications of Multinomial Distribution Function

Let E_i (i=1, 2, ... , k) be k mutually independent events, and the probability of occurrence of the event E_i is equal to q_i. Then the joint distribution of the random variables $\mathbf{n}_i$ (i = 1, 2, ... , k) representing the numbers of occurrences of the events E_i (1, 2, ... , k) respectively, in N trials (with $n_1 + n_2 + \ldots + n_k = N$) is called *multinomial distribution function* and defined by

$$\mathbf{P}(n_1, n_2, \ldots, n_k) = N! \prod_{i=1}^{k} (q_i^{n_i}/n_i!).$$

The multinomial distribution function is employed in many diverse fields of statistical analysis. In general, it is used in the same circumstances as those in which a binomial distribution might be used, when there are multiple categories of events instead of a simple dichotomy. For example:

In computer science, a program requires I/O, input or output services from device i with probability q_i at the end of a CPU, *central processing unit*, with $q_1 + q_2 + \ldots + q_k = 1$. This situation gave rise to a

multinomial distribution problem [6].

If one observes n CPU burst terminations, then the probability that n_i of these will be directed to I/O device i (for i = 1, 2, ... , k) is given by the a multinomial probability mass function. One may also replace the "I/O device" by a word "file" for the application in the database management system. Another application of multinomial distribution function occurs in a operating system when we consider a paging system and we model a program using the independent reference model [2]. In this model, we assume that successive page references are independent and the probability of referencing page i is q_i.

Another important field of application is in the kinetic theory of classical physics [4]. Particles are considered to a cell in a six-dimensional space, three for position and the other three for velocity. Each allocation of N particles among the k cells available constitutes a microstate. The thermodynamic probability of a macrostate is proportional to the multinomial distribution function.

3. The Method and Its Mathematical Foundations

This research will apply prime number factorization to factorials and rewrite probabilities q_i (i=1, 2, ... , k) in the simplest fraction form with denominator as a product of prime numbers. Then the cancellation of numerator and denominator is applied to reduce the computational complexities to the minimum and to achieve maximum accuracy. Also the Ada programming language's special features [1, 7], "exception handling" and "tasking" are used to handle the difficulties of computation such as: overflow and underflow problems, redundant multiplications and divisions of the same numbers, and time comsuming computation. These Ada special features will make the computation effective, efficient, and accurate. The mathematical foundation for this method is given as follows:

In order to reduce the computational complexity of $\mathbf{P}(n_1, n_2, \ldots, n_k)$, we need theorems from the theory of numbers [3] which is stated and proved as below:

Theorem 1. Let p be a prime. Then the exact exponents of p that divides n! is

$$\left[\frac{n}{p}\right] + \left[\frac{n}{p^2}\right] + \left[\frac{n}{p^3}\right] + \cdots,$$

where [x] is the largest integer less than x.

Proof: For

$$\begin{aligned} n! = {} & 1 \cdot 2 \cdot 3 \cdots (p-1) \\ & \cdot p \cdot (p+1) \cdot (p+2) \cdots 2p \cdots (p-1)\, p \\ & \cdot p^2 \cdot (p^2+1) \cdot (p^2+2) \cdots \\ & \cdot p^3 \cdot (p^3+1) \cdot (p^3+2) \cdots \\ & \cdots \cdots (n-1) \cdot n. \end{aligned}$$

We see that the number of p's factors is [n/p] the number of p^2's factor is $[n/p^2]$, the number of p^3's factors is $[n/p^3]$, and so forth. Then the Theorem follows.

From Theorem 1, we are able to factor the n!, for all $n \geq 1$, as a product of prime numbers. The result is given in Theorem 2 below:

Theorem 2. For any positive integer $n \geq 2$, the n! can be written as a product of prime numbers.

$$n! = p_1^{r_1} \cdot p_2^{r_2} \cdot p_3^{r_3} \cdots p_k^{r_k},$$

for some positive integer k.

Example : Consider the 20!, we have the following exponents of prime numbers:

$$\text{The exponent of } 2 \text{ is } \left[\frac{20}{2}\right] + \left[\frac{20}{2^2}\right] + \left[\frac{20}{2^3}\right] + \left[\frac{20}{2^4}\right]$$

$$= 10 + 5 + 2 + 1 = 18.$$

The exponent of 3 is $\left[\frac{20}{3}\right] + \left[\frac{20}{3^2}\right] = 6 + 2 = 8.$

The exponent of 5 is $\left[\frac{20}{5}\right] = 4.$

The exponent of 7 is $\left[\frac{20}{7}\right] = 2.$

The exponents of 11, 13, 17, and 19 are all equal to 1.

Hence, the $20! = 2^{18} \cdot 3^8 \cdot 5^4 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19.$

4. Some Computational Results

The following sample results were obtained from an output of an Ada program; this program was running on a MicroVaxII machine. The program can compute the probabilities of multinomial distribution up to 200 categories of events and some results and their computation time are given as follows:

(1) $n_1 = 10, \quad q_1 = 0.200$
$n_2 = 15, \quad q_2 = 0.300$
$n_3 = 5, \quad q_3 = 0.100$
$n_4 = 12, \quad q_4 = 0.200$
$n_5 = 8, \quad q_5 = 0.200$

The probability,

P(10, 15, 5, 12, 8)
=4.26081493102356544070815822094335O E − 04

The time used for the computing is
1.899414062500000 E − 01

(2) $n_1 = 4, \quad q_1 = 0.050$
$n_2 = 3, \quad q_2 = 0.050$
$n_3 = 2, \quad q_3 = 0.200$
$n_4 = 3, \quad q_4 = 0.300$
$n_5 = 5, \quad q_5 = 0.200$
$n_6 = 4, \quad q_6 = 0.050$
$n_7 = 3, \quad q_7 = 0.050$
$n_8 = 2, \quad q_8 = 0.200$
$n_9 = 3, \quad q_9 = 0.300$
$n_{10} = 5, \quad q_{10} = 0.200$

The probability,

P(4, 3, 2, 3, 5, 4, 3, 2, 3, 5)
=2.99145322229358904938097171835734O E − 11

The time used for the computing is
1.299438476562500 E − 01

(3) $n_1 = 4, \quad q_1 = 0.010$
$n_2 = 7, \quad q_2 = 0.090$
$n_3 = 10, \quad q_3 = 0.020$
$n_4 = 3, \quad q_4 = 0.010$
$n_5 = 12, \quad q_5 = 0.050$
$n_6 = 13, \quad q_6 = 0.120$
$n_7 = 5, \quad q_7 = 0.130$
$n_8 = 7, \quad q_8 = 0.020$
$n_9 = 21, \quad q_9 = 0.050$
$n_{10} = 3, \quad q_{10} = 0.250$
$n_{11} = 9, \quad q_{11} = 0.010$
$n_{12} = 11, \quad q_{12} = 0.020$
$n_{13} = 6, \quad q_{13} = 0.010$
$n_{14} = 3, \quad q_{14} = 0.030$
$n_{15} = 7, \quad q_{15} = 0.180$

The probability,

P(4, 7, 10, 3, 12, 13, 5, 7, 21, 3, 9, 11, 6, 3, 7)
=2.70257344796675865790915932356903O E − 47

The time used for the computing is
3.699951171875000 E − 01

5. Conclusions

The computation of the *multinomial distribution function* is in general a critical problem due to the limitation of the computer systems and programming techniques. The goal of a computation is accuracy; the time consumed for the computation must also remain reasonably small. This research has developed a method based on theorems from the theory of numbers and implemented them in the Ada programming language, the former is to break the limitations of a computer system and the later is to solve the technical difficulty in the programming. Since the computation of the *multinomial distribution function* is a number theory problem in the nature. The problem can only be solved in

number theory.The results given in Section 4 have reached the predefined goal for this computation.

References

1. Barnes, J.G.P. (1989) *Programming in Ada,* Third Edition, Addison-Wesley Pulishing, Reading, Massachusetts.

2. Coffman, E.G. and Denning, P.J. (1973) *Operating System Theory,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

3. Hua, L.K. (1982) *Introduction to Number Theory* (English Translation), Springer-Verlay, New York.

4. Huang, K. (1965) *Statistical Mechanics,* John Wiley & Son, Inc., New York.

5. IMSL Library (1984) *FORTRAN Subroutines for Mathematics and Statistics, User's Manual,* IMSL, Inc., Houston, Texas.

6. Trivedi, K.S (1982), *Probability & Statistics with Reliability, Queuing, and Computer Science Applications,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

7. Vax Ada (1985) *Language Reference manual,* Digital Equipment Corporation, Maynard, Massachusetts.

FGP: Using Statistics to Drive an Expert Database

Scott Fertig
Department of Computer Science
Box 2158, Yale Station
Yale University
New Haven, Connecticut 06520-2158

ABSTRACT

We describe the FPG machine, which uses similarity-based retrieval and "simulated speculation" to convert pools of data directly into quasi-expert advice. The central operation is the retrieval of a small set of records *similar* to a partially-instantiated new record. The system uses two statistical techniques to improve on the standard Euclidean measure for calculating distance between two records represented as a vector of features. One is a facility to automatically weight the importance of features which will add or subtract to those features' contribution to the overall distance score. The other is a means for separating the most relevant records from the rest by finding a natural break in the ordering of the records by distance from the input. We explain the role these techniques play in the overall operation of the system in the next section; the algorithms used for the calculations are described in the appendices.

1 INTRODUCTION

The program we've built is named the FGP machine, after its basic operations—fetch, generalize and project. We imagine the FGP machine's database as a collection of regions in space (*cf.* the standard vector space text-retrieval model). Each element of the database corresponds to some region. Nearby regions correspond to nearby cases. When presented with an inquiry, the machine's basic task is to add to the database a new region corresponding to the inquiry. Stationing itself on top of this new region (so to speak), the machine then looks around and reports the identities of the nearby regions—these will correspond to elements of the database that are nearby to, in other words closely related to, the subject of the inquiry. We can then inspect this list of nearby regions and "generalize"—determine which attributes tend to be shared in common by all or by most of them. We can guess that these common attributes are likely to hold true for the case being described in the inquiry as well.

Having reached whatever conclusions seem reasonable, the machine may now indulge in a bit of simulated speculation. Temporarily turning aside from the inquiry in hand, it focusses on any "evocative possiblities" that may have suggested themselves during the examination of nearby regions. An "evocative possibility" is a datum that *might* be true, and that would be significant if it were. The calculation of evocativeness is discussed in appendix A. The machine's interaction with the user represents a combination of fairly safe conclusions, speculation experiments and the subsequent investigation of resulting guesses. An example transcript is shown in figure 1.

The system can operate interactively, but here it is working in "commentary" mode: the user presents an entire case; the system scans it element-by-element, offering comments. This case initially seems malignant (note the early mention of related cases with diagnoses of infiltrating ductal carcinoma); the fact that the mass has not changed in density and has no comet (contradicting the system's guesses, which in the nature of guesses will often be wrong) points in the other direction ("cyst" and "fcd" refer to benign diagnoses); but further data, particularly the absence of a halo, tips the balance, and the system guesses that this is a malignant mass. This guess is correct, and the diagnosis was in fact infiltrating ductal carcinoma. This transcript is driven by a small collection of 67 cases, which is the only domain knowledge provided.

2 THE MODEL

An FGP machine is defined in terms of a single kind of data-object and three primitive operators. These define a virtual machine in terms of which the system is programmed. We summarize the essential points in the remainder of this section; see [1] for further discussion.

Data-objects & databases.

FGP machines run off a database of a single type of data-object, a feature tuple to which we refer generically as a τ. A τ consists basically of a list of attribute-value pairs; we give examples below. An FGP database consists of an unordered collection of τ's. A new case for inclusion in the database is presented as a τ, and a query is an incomplete τ—a partial list of attribute-value pairs, with a request that the system fill in certain missing ones. We use $\mathcal{M}$ to represent an unordered collection of τ's; an FGP database of stored cases and paradigms is an $\mathcal{M}$. $\mathcal{L}$ is a list of τ's ordered on their "closeness" to some other τ: we explai: below.

Primitive operators.

The three basic FGP operations are *fetch*, *generalize* and *project*. They work as follows.

fetch maps a τ and an $\mathcal{M}$ to an $\mathcal{L}$: given a feature-tuple and a database of feature-tuples, it produces an ordered list of those feature-tuples in the database that are "closest to" the τ mentioned in the query. It is this operation that makes use of the statistical techniques that are detailed in the appendices.

fetch uses a two-step procedure. First it calculates a "distance" from the new point to every point in the database[1];all

[1]This calculation needn't require that every τ in $\mathcal{M}$ be examined; we can use a hash scheme to direct attention to τ's that occupy com-

(AGE 58)	
(MASS_DENSITY ISO_DENSE)	
(MASS_BORDER_COMPLETE? NO)	
(MASS_TYPE_BORDER IRREGULAR)	
(MASS_BORDER_DEFINED? NO)	*Speculating: MASS_DENSITY_CHANGED?...* *Guessing INCREASED - e.g.* *case ((id 14) (age 46) (diagnosis CA_INF_DUCTAL))* *case ((id 50) (age 70) (diagnosis CA_INF_DUCTAL))* *Speculating: MASS_COMET?...* *Guessing YES - e.g.* *case 14* *case ((id 40) (age 69) (diagnosis CA_INF_DUCTAL))*
(MASS_LOCATION UIL)	
(MASS_SIZE_CHANGED? YES)	
(MASS_DENSITY_CHANGED? NO)	
(MASS_COMET? NO)	*Speculating: BACKGROUND_DENSITY...* *Guessing DENSE - e.g.* *case ((id 21) (age 61) (diagnosis cyst))* *case ((id 47) (age 45) (diagnosis fcd))*
(MASS_HALO? NO)	
(BACKGROUND_DENSITY MODERATE)	*Concluding* *(ARCHITECTURAL_DISTORTION? NO)* *Speculating: MALIGNANT?...* *Guessing YES - e.g. cases (2 6 8)* *Speculating: SKIN_CHANGES...* *Guessing RETRACTION - e.g. cases (2 8 28)*
...	
(SKIN_CHANGES NO)	
(NIPPLE_INVERSION? NO)	
(ADENOPATHY? NO)	
(FAMILY_HISTORY_CANCER SISTER)	
(PERSONAL_HISTORY_CANCER NO)	*Closest known cases:* *(19) (YES) (CA_INF_DUCTAL)* *(33) (YES) (CA_INF_DUCTAL)* *(26) (YES) (CA_INF_DUCTAL)* *(28) (YES) (CA_INF_DUCTAL)* *(18) (YES) (CA)* *YES has been concluded or guessed for MALIGNANT?* *Speculating: DIAGNOSIS...* *CA?* *CA_INF_DUCTAL?*

Figure 1: Transcript of an FGP machine operating in the domain of mammography. The user's case description is in the left column, the system's commentary on the right.

cases further away than some parametrized threshold are removed from further consideration. *fetch*'s calculation not only takes into consideration the number of shared attributes and their types, but also, in the context of a request to fill in values for missing attributes, the "evocativeness" of each with respect to the current goal—a more evocative feature is one that recalls a group of cases with a more highly focussed set of values for the goal. The evocativeness of an attribute-value pair with respect to a goal attribute is inversely proportional to the entropy (disorder) of the distribution of values for the goal represented in the group of cases returned by *fetch*. See appendix A for details on how this value is calculated. Next *fetch* checks to see if there exists a well-defined group of "close" points among those remaining by performing a crude cluster analysis. Appendix B describes the clustering algorithm. An ordered list of these close points is returned as *fetch*'s value.

generalize maps an $\mathcal{L}$ to a τ: it takes an ordered list of feature-tuples and compresses them into a single new feature-tuple. The weighter a τ and the closer it is to the top of the list, the larger the contribution its attribute-value pairs make to the combined τ returned by *generalize*. Suppose we query on the τ (name apple), and suppose that $\mathcal{M}$ holds one hundred individual apples, half red and half yellow; a *generalize* operation over a list consisting of one hundred apples, half red and half yellow, yields a single τ that might look like ((name apple 100) (type fruit 100) (color (red 50) (yellow 50)) ...).

project maps a τ to a τ: given a feature-tuple it returns a new tuple constructed from a subset of the features in the original. While *project* is a purely syntactic operation, it is used by higher level operations (see the discussion of *refocus* in [1]) to change contexts; the system focusses on those attributes and values that are evocative, temporarily ignoring other information on hand.

2.1 THE BASIC CYCLE

Given this three-instruction virtual machine, how does the system operate? The basic cycle is two phase: (1) extend the current τ; (2) choose a new current τ, and repeat. Step one is implemented by an *extend*(τ) function that is defined in terms of *fetch* and *generalize*. Step two is implemented by *refocus* which is defined in terms of all three.

To extend a τ — to discover new implications given our database of cases and paradigms — we begin by executing the operation *generalize*(*fetch*($\mathcal{M}$, τ)), where $\mathcal{M}$ is the database. If τ, for example, describes a particular patient, *fetch*($\mathcal{M}$, τ) will return a list of remembered τ's that are close to (similar to or reminiscent of) this particular patient; executing *generalize* over this list will produce an amalgam of all these remembered cases. Any highly-focussed and sufficiently-weighty values can be classified as conclusions: if the memories examined by *generalize* mainly have a value of "blonde" for attribute "hair-color", say, the system will conclude that (hair-color blonde) is likely to characterize this case as well. It reports (hair-color blonde) to the user as a conclusion and augments the current τ with this new attribute-value pair. The system attempts to conclude any value turned up by the *fetch-generalize* combination which hasn't yet been seen in the context of the current query. Values which contradict[2] are withdrawn; the user's input always takes precedence over system guesses. The *extend* operation is complete when all values that can be concluded have been and all contradictions removed.

SIMULATED SPECULATION

Refocus is then invoked over the extended τ. Its role is to examine a τ and refocus attention from this entire τ to one (possibly small and conceivably unrepresentative) part of it. This element considered in isolation may serve as a seed for a new set of inferences. We call this process "simulated speculation." *refocus* may choose no, one or many data points; each chosen data point becomes the current τ in turn. The more evocative a data point with respect to the goal—the more sharply-defined the cases nearby a τ consisting only of that data point with respect to the goal attribute, in other words—the likelier target for *refocus*. The more sharply a data point stands out from the pack—by assumption it won't stand out clearly enough to qualify as a conclusion, but there are many intermediate shadings here—the likelier a *refocus* target.

Typically, the system will examine each of a small set of values associated with a particular attribute whose value, if known, would focus the search space considerably. The system performs the basic *fetch-generalize* cycle on each of these seed-tuples and is left with a set of regions in vector-space. One may be much closer to the original query than the others and may therefore be mergeable with it. The reader can see the system's behavior during several *refocus* experiments by examining the transcript shown in figure 1. *refocus* first announces the attribute *projected* to, followed by any values tentatively guessed as the result of the speculation experiment. It then gives pointers to specific cases that both have this value and also resemble the rest of the user's input.

3 PERFORMANCE

A version of the FGP machine was implemented in the T-dialect of Scheme. There are approximately 5000 lines of code spread among 10 modules.

We are encouraged by our initial tests of the system. Experiments were conducted on case databases in three domains, one of which we discuss here. This test involved a small database of patient records, specifically descriptions of mammograms. There were originally 88 records in the database; 20 cases were reserved for testing and 67 were used to seed the system spanning 13 possible diagnoses (one of these 13 possibilities was the diagnosis *normal* meaning no disease present)[3]. The system was presented with the 20 test cases and asked to judge if a malignant lesion was indicated and if so determine a specific diagnosis. As discussed above, the system would present a short list ($\leq$4) of possible diagnoses if unable to decide on one with certainty.

The domain expert was the radiologist who had compiled the database [4]. Working from the descriptions of the mammograms alone, he accurately judged the malignancy of the testcases at the 68% level. The system performed slightly worse at 63%. However the system outperformed the domain expert in producing a differential diagnosis, with the right answer being stated outright or appearing in a short list of possibilities (<= 3) 70% of the time to the clinician's 60% correct performance

parable subspaces.

[2] As expected, two distinct values of a boolean-typed attribute always contradict. System-concluded values of other types of attributes contradict only if this information is specified in the attribute's distance metric. See [1] for more details.

[3] One record was thrown out because no diagnostic information was included.

[4] Dr. Paul Fisher of the Department of Diagnostic Radiology, Yale University School of Medicine. Dr. Fisher's clinical specialty is mammography.

4 CONCLUSION

We have presented the main features of a methodology for extracting expertise from case databases automatically. The FGP systems's domain independent similarity-based weighting and clustering algorithms support retrieval of noisy and incomplete data, drive an intelligent interface, and provide a mechanism for incremental learning of concepts. Experiments with a portion of the National Cancer Institute's SEER tumor registry are beginning and should tell us how well the architecture scales in the face of truly large databases.

A CALCULATION OF EVOCATIVENESS

The calculation of evocativeness attempts to determine how strongly a presenting case r brings to mind a value for the current goal. What we would really like to measure is how much *information* about the goal we gain from r. Does r strongly suggest only one goal value, or does it bring to mind ten goal values which are equally likely? One way to measure this information content I is to relate it to the entropy D of a probability distribution.

Let the total number of goal values found in the top-cluster $\mathcal{Q}$ be T. Assuming that the probability p_i of goal value i being correct is proportional to the number of times n_i it occurs in the top-cluster, we can calculate the entropy (disorder) of the distribution:

$$\begin{aligned} D &= -\sum_i p_i \ln p_i \\ &= -\sum_i \frac{n_i}{T} \ln \frac{n_i}{T} \\ &= -\frac{1}{T}\sum_i n_i \ln n_i + \frac{\ln T}{T}\sum_i n_i \\ &= -\frac{1}{T}\sum_i n_i \ln n_i + \ln T \end{aligned}$$

The entropy function D ranges from a value of 0, occurring when only one goal value is represented, to $\ln N$, where N represents the total number of possible values for the goal in the database $\mathcal{M}$. We can scale the entropy to range from 0 to 1 simply by dividing by $\ln N$:

$$S = \frac{1}{\ln N}\left(-\frac{\sum_i n_i \ln n_i}{T} + \ln T\right)$$

We can further adjust the scale so that a scaled-entropy of 0 corresponds to the maximum evocativeness allowed by the system, while a scaled-entropy of 1 corresponds to the minimum evocativeness allowed by the system. By setting the endpoints of the scale far enough apart, we can coerce a particular evocativeness value to an integer without meaningfully reducing precision, and therefore avoid the cost of storing and calculating with floating point numbers. This integer is the final evocativeness number E used by the FGP machine.

$$E = S(*\texttt{min-evoc}*) + (1-S)(*\texttt{max-evoc}*)$$

B CLUSTERING ALGORITHM

fetch maps a r and an $\mathcal{M}$ to an $\mathcal{L}$: given a feature-tuple and a database of feature-tuples, it produces an ordered list of those feature-tuples in the database that are "closest to" the r mentioned in the query. To do this, *fetch* needs an algorithm to cluster the values returned by the distance calculation. We use an algorithm developed by Mitchell Sklar that is efficient and experience has shown performs reasonably well. The description of the algorithm that follows is taken from Sklar's medical school thesis [2].

(We can use a routine) *CLUSTER* to find a natural break point in a list of cases, dividing that list into a "close" group $\mathcal{C}$ and a "distant" group $\mathcal{D}$. Referring to a list of numerical distances, the algorithm attempts to partition the list into two groups x_i and y_j such that the sum of the squared deviations *within* the groups is locally minimized. That is, *CLUSTER* attempts to find a local minimum for

$$\sum_{i=1}^{m}(x_i-\bar{x})^2 + \sum_{j=1}^{n}(y_j-\bar{y})^2.$$

Computationally, however, this calculation is inefficient. Instead, we can note that if μ represents the mean value of all the distances x_i and y_j combined, then

$$\begin{aligned} \sum_{i=1}^{m}(x_i-\mu)^2 &= \sum_{i=1}^{m}\left(x_i - \frac{m\bar{x}+n\bar{y}}{m+n}\right)^2 \\ &= \sum_{i=1}^{m}\left(x_i - \bar{x} + \frac{n}{m+n}(\bar{x}-\bar{y})\right)^2 \\ &= \sum_{i=1}^{m}(x_i-\bar{x})^2 + \frac{mn^2}{(m+n)^2}(\bar{x}-\bar{y})^2 \end{aligned}$$

Similarly,

$$\sum_{j=1}^{n}(y_j-\mu)^2 = \sum_{j=1}^{n}(y_j-\bar{y})^2 + \frac{m^2n}{(m+n)^2}(\bar{y}-\bar{x})^2$$

Combining these results gives

$$\begin{aligned} &\sum_{i=1}^{m}(x_i-\mu)^2 + \sum_{j=1}^{n}(y_j-\mu)^2 = \\ &\sum_{i=1}^{m}(x_i-\bar{x})^2 + \sum_{j=1}^{n}(y_j-\bar{y})^2 + \frac{mn}{(m+n)}(\bar{x}-\bar{y})^2 \end{aligned}$$

Since the quantity on the left is constant for the given list of distances no matter what the partition, we can minimize the total intra-group summed squared deviations simply by choosing our partition to find the first maximum for the last quantity,

$$\frac{mn}{(m+n)}(\bar{x}-\bar{y})^2.$$

This will give us the first clean break in the distance list.

REFERENCES

[1] Scott J. Fertig. The design, implementation, and performance of a database-driven expert system. Technical Report 851, Yale University Department of Computer Science, Aug 1990.

[2] Mitchell J. Sklar. Mu: A domain-independent case-based expert system, 1988. Thesis submitted to the School of Medicine.

Databasing Longitudinal Data: Approaches in S

V. Carey, Y. He, A. Muñoz
Johns Hopkins School of Hygiene and Public Health
Baltimore, MD 21205

May 28, 1991

1 Introduction

Statements of statistical modeling objectives are often sufficiently formal as to provide a basis for construction of precise database queries. Specific examples may be found in section **3** below; a form of query commonly encountered in the analysis of longitudinal data is: "obtain all ordered p-tuples of repeated measurements subsequent to (or prior to) the occurrence of a certain event." For certain kinds of modeling, there may be a further proviso to the effect that the measurements selected should have been obtained at certain regular intervals. Depending on the nature of the study under consideration, resolution of such queries may entail problems of

- *spacing verification:* determining that the components of a long candidate vector are approximately equally spaced in time, and taking proper action in the presence of irregularities
- *synchronization:* although the data in general may be obtained in a very regular fashion, measurements of interest to the analyst may not be synchronized from subject to subject; instead, the "origin" of the time-course in a variable of interest may be subject-specific (depending, for example, on the time of an exposure event)
- *attribute linkage:* the analyst's interest in the value of a given attribute at time t may depend on the fact that t was the first time at which some other attribute attained a certain value.

Such an enumeration of problems arising in data manipulations preparatory to longitudinal analysis is hardly exhaustive, but may serve to convince the reader that no direct means to address these problems exists in standard facilities for data manipulation connected with statistical analysis environments (the SAS data step, S matrix manipulation facilities, relational databases). Consequently, users of these environments who wish to analyze actual longitudinal data are often engaged in detailed programming tasks to extract and format data with the required longitudinal properties. There is no question that the environments are adequate to support such programming, but the programming itself is expensive, prone to error, and is often thrown away.

Our objective in this report is to consider how to reduce programming burdens encountered in manipulating data for longitudinal analyses. Clearly, part of the burden will depend on the form of the permanent data store, and we propose a "longitudinal relation" as a format for permanent representation of observations obtained in longitudinal studies. Our major concern, however, is the formulation of a programmable query idiom which is in close correspondence to the statement of the modeling objective. This formulation has not been achieved, but we will discuss a working function on longitudinal relations which solves some problems of interest. A satisfactory interface to this function may require language concepts and tools not accessible to the statistical user of S.

All of the programming related to this essentially conceptual investigation is carried out in S. The ultimate realization of the objectives explored here would likely be implemented in some other language or database function; our purpose here is in establishing broader features of the problem and possible solutions.

2 Longitudinal Data

The basic data structure we are concerned with is derivable from "panel studies", "follow-up cohort studies", "repeated-measures studies", though some of these terms may suggest aspects of regularity or data balance which we do not assume. There are I subjects presenting for observations repeated in time. For the moment, confine attention to the case where the observation is scalar-valued, on the variable X. Subject i provides n_i measurements on X; n_i may differ from subject to subject. T_i is the n_i-vector of unique times of observation, measured in some convenient scale from some common origin; $\{T_i\}$ is the set of elements of T_i. For $i \in 1, \ldots I$, $t \in \{T_i\}$, we denote by X_{it} the value of measurement X obtained on subject i at time t.

Let $\mathcal{I} = \{1, \ldots, I\}$, $\mathcal{T} = \bigcup_i \{T_i\}$. Then $\mathcal{D} = \mathcal{I} \times \mathcal{T}$ is a natural index set for the longitudinal data X_{it}. The suitability of this index set to actual use for organization of the data will depend on the distribution of inter-visit gaps and on the variation of n_i with i. In the case of "equidistant" ($|T_{ij} - T_{ik}| \equiv c$, all $j \neq k$), "balanced" ($n_i \equiv n$, all i), "complete" data (no missing observations), every point in $\mathcal{D}$ corresponds to a unique data point. If only the "equidistant" condition is dropped, $\mathcal{I} \times \{1, \ldots, n\}$ may be used to index both X and the set of observation times.

In practice, X is vector-valued, and the set of components of X (and of course the values of these components) may differ from time to time (within individual) and from individual to individual. Furthermore, equidistance and balance are rarely achieved in practice. Therefore the simple indexing schemes just discussed will be useful only if considerable sparseness is tolerated.

3 Analysis of longitudinal data

We mention a few analytical activities relevant in data analysis of such studies. We assume throughout that time is measured in units from an origin common to all subjects.

3.1 Conditional autoregression

Let time be measured in units. Adopt the model $Y_{it} = \beta_0 + \sum_{k=1}^{p} \beta_k Y_{i,t-k} + X_i\gamma + \epsilon_{it}$; X_i (and γ) may be vector-valued, and some components of X_i may be time-dependent. We refer to this model as autoregressive with order p, $AR(p)$. To fit the model, the ordered $p+1$-tuple of equidistant measurements on Y_i must be obtained and collated with the appropriate elements of X_i to establish the contribution(s) of subject i to the outcome vector and predictor matrix to be submitted to a regression procedure. If the subject presents more than $p+1$ measurements, it is possible that several $p+1$-tuples may be suitable for the analysis, and all must be obtained. See Muñoz et al., (1988), for example and further references.

3.2 Proportional hazards regression

Writing $x(t)$ for the vector of time-dependent covariates at time t, the model for the hazard of an event is $h(t|x(t)) = h(t|x(t) = 0)exp(x(t)\beta)$. We consider the implementation (agreg) supplied by Therneau (1991) to STATLIB for use in S. Partition $[0, T_{i,n_i})$ into disjoint intervals on each of which $x_i(t)$ is constant, and denote the j^{th} such interval by $[s_{ij}, u_{ij})$. We may assume that there are $n_i - 1$ such intervals without loss of generality. The required data structure for the j^{th} contribution from individual i is $\langle s_{ij}, u_{ij}, \delta(u_{ij}), x_i(t_{ij}) \rangle$, where $\delta(u)$ is the indicator of "event occurs at time u", $s_{ij} \leq t_{ij} < u_{ij}$, and the vector $x_i(t)$ is constant on $[s_{ij}, u_{ij})$.

3.3 Discussion

These examples are emblematic, and the data manipulation activities entailed by these particular problems arise in other settings. We identify a few of the broad features of the "required" data.

- The "time-gap" separating observations is a crucial datum; sequences of gaps may play an important role in identifying analyzable contributions.

- Observations on different attributes (e.g., outcome and predictor variables) may need to be "linked" (for extraction purposes) with regard to their time of measurement. The nature of linkage may be complicated, not limited to e.g., "simultaneity".
- The structure of the contribution need not be a function of elapsed time only, but may depend on the values taken by time-dependent va ables; the time at which such variables take on certain critical values may constitute a subject-specific "origin".

4 Longitudinal relation

The application of relational database techniques to the management of longitudinal data may take various forms. We define a longitudinal relation to be a relation comprising observations as described in section **2** above, with the ordered pair $\langle i, t \rangle$ as the compound key for the relation. As an example, we provide an extract from a hypothetical cohort study of HIV infection; data on age, markers of infection, and infection status are recorded.

id	date	age	cd4	cd8	HIV
70328	9597	36.8	338	1222	+
70328	9772	37.3	542	617	+
70328	10163	38.3	312	1656	+
70328	10346	38.8	270	1645	+
70319	8861	38.3	1021	688	-
70319	9049	38.8	785	447	+
70319	9238	39.3	915	826	+
70319	9412	39.8	848	915	+

The longitudinal relation is not the inevitable form of organization for longitudinal data. Often, "flat files" are constructed at certain stages in the study, with the file comprehending all observations falling in a certain interval. These files are then subject to frequent merging and subsetting.

A unified longitudinal relation may be awkward for the combination of attributes varying smoothly in time and discrete attributes. For example, the **HIV** attribute above has values in each of the n_i rows for subject i, but these n_i data items record the single piece of information: "first date at which infection was observed."

5 Processing longitudinal relations

Our approach to the use of longitudinal relations for extracting data for statistical analysis will be illustrated for the case of the $AR(1)$ model (see section **3.1**.) The longitudinal relation is readily implemented in an S matrix bearing attribute-names as column-names. For a very simple $AR(1)$ model for change in **cd4**, we require equidistant pairs of observations on this attribute, lagged at approximately six month intervals. As a covariate, we employ **cd8** measured at the lagged time. A possible solution is the following longitudinal relation:

id	date	cd4	cd8	date+	cd4+	cd8+
70328	9597	338	1222	9772	514	617
70328	10163	312	1656	10346	270	1645
70319	8861	1021	688	9049	785	447
70319	9049	785	447	9238	915	826
70319	9238	915	826	9412	848	915

We have adopted the convention that **var+** is the value of **var** at the "next" time as required by the spacing scheme. Such suffixing is naturally iterative. Having obtained such a longitudinal relation in an S matrix, say **ar1ld**, the S command

```
lsfit( ar1ld[,c("cd4","cd8")],
       ar1ld[,"cd4+"])
```

is one way to estimate the parameters of the model of interest.

We have implemented an S function, **pairgen**, to carry out this process. The user must supply an "admission function", which operates on criterial variables (typically the "time" component of the key is a criterial variable) to indicate which rows of the input longitudinal relation should be combined to produce an output longitudinal relation whose rows possess data elements satisfying certain time-dependent conditions. In the present example, the

admission function specified that a pair of observations is to be admitted to the output relation only if the times of the observations are separated by more than 160 and fewer than 200 days.

It may be worth noting that, at least for subject 70319, the output relation attributes (`date, date+`) are a partition of that subject's observation time-line as would be needed for the `agreg` analysis mentioned in section **3.2**. It is straightforward to generate such partitions from longitudinal relations using trivial admission functions.

Because the `pairgen` function considers a vector of criterial variables, it addresses the problems of synchronization and attribute linkage mentioned in the introduction: the condition for admission of an observation to the output relation may be specified in terms of arbitrarily many attributes in the input relation.

6 Discussion

The problem of *effecting transformations of* data from permanent storage ("archives" or "databases") into the structures required by particular analytical procedures is often addressed by programs in high-level languages, in isolation from actual statistical procedure-invocation. Programmed statistical procedures are used essentially as targets: a procedure is selected in accordance with modeling objectives, the input requirements of the procedure are ascertained, and then data in the permanent store are extracted and transformed in accordance with the input requirements.

This extraction and transformation process is highly error-prone and inspires too much "throw-away" programming effort. We propose that classes of modeling objectives be identified, that the data structures needed in pursuit of these objectives be identified, and that data management systems be equipped with high-level functions delivering these data structures. We have focused attention on modeling objectives related to longitudinal data analysis, and have identified some data structure features which must often be obtained in performing such analyses. While it is obviously feasible for analysts to develop *ad hoc* extraction and formatting procedures to facilitate longitudinal analysis, we have investigated the possibility of implementing a systematic approach and our results suggest that further effort may be profitable. The chief difficulty we face in making this functionality widely usable is in obtaining an interface with a natural syntax. The appendix presents the current interface to the `pairgen` function. This function generates regularly spaced longitudinal pairs and partitions observation time-lines in a useful fashion. A similar function might be developed in the SAS data step, based on the LAGn functions, or in the programming system of a RDMBS.

7 Appendix

The `pairgen` function takes three arguments, a longitudinal relation, a list of criterial variables, and an admission function. The list of criterial variables is a subvector of the "attribute-names" vector of the longitudinal relation. The admission function must be written in terms of elements of a vector of criterial variables extended to embrace the naming convention described in section **5**. To extract from the first relation in section **4** the pairs representing lags approximately six months in length and ending with `cd8` values lower than 800, the criterial variables are identified in the vector `c("date","cd8")`, and the following admission function might be used:

```
function(x)
  {
  x["date+"] - x["date"] > 160 &
  x["date+"] - x["date"] < 200 &
  x["cd8+"] < 800
  }
```

8 References

- Muñoz A, et al., "Predictors of Decline in CD4 Lymphocytes in a Cohort of Homosexual Men Infected with Human Immunodeficiency Virus", *Journal of Acquired Immune Deficiency Syndromes*, 1988(1), 396-404.
- Therneau, T., `agreg`, in the S archive at `statlib@lib.stat.cmu.edu`.

Covariance Structure Analysis Under a Simple Kurtosis Model

by

P. M. Bentler*
University of California, Los Angeles

Maia Berkane
Leiden University

Yutaka Kano
University of Osaka Prefecture

Abstract

A model for the relation between multivariate fourth-order central moments of a set of variables and the marginal kurtoses and covariances among these variables is used to produce an estimator for covariance structure analysis that is asymptotically efficient and yields an asymptotic χ^2 goodness of fit test of the covariance structure while substantially reducing the computations. When the kurtoses of the variables are equal, the method reduces to one based on multivariate elliptical distribution theory, and, when there is no excess kurtosis, to one based on multivariate normal distribution theory.

Introduction

In covariance structure analysis, the p x p population positive definite covariance matrix Σ is hypothesized to be a function of a q x 1 vector $\theta \in \Theta$ of more basic parameters $\Sigma = \Sigma(\theta)$. An asymptotically efficient distribution-free estimator $\hat{\theta}$ of θ can be obtained by minimizing the quadratic discrepancy between

$$F = (s - \sigma(\theta))'\hat{\Gamma}^{-1}(s - \sigma(\theta)) \qquad (1)$$

where $\sigma(\theta) = \text{vecs}(\Sigma(\sigma))$, $s = \text{vecs}(S)$, and $\hat{\Gamma}$ is a $(p^* \times p^*)$ weight matrix converging in probability to a positive definite matrix Γ, the asymptotic covariance matrix of s. Here, S is the usual sample covariance matrix based on a sample of size $n + 1$, $p^* = p(p+1)/2$, and vecs is the p^* x 1 column vector formed from the nonduplicated elements of S. Under the null hypothesis, at the minimum of (1)

$$n\hat{F} = n(s - \sigma(\hat{\theta}))'\hat{\Gamma}^{-1}(s - \sigma(\hat{\theta})) \sim \chi^2_{(p^* - q)} \qquad (2)$$

is asymptotically distributed as a central χ^2 variate with $(p^* - q)$ degrees of freedom, and the asymptotic covariance matrix of the estimator is given by

$$\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{\mathcal{L}} \mathcal{N}[0, (\Delta'\Gamma^{-1}\Delta)^{-1}], \qquad (3)$$

where $\Delta = \partial\sigma(\theta)/\partial\theta'$, evaluated at the true value $\theta = \theta_0$. Appropriate regularity conditions for (2) and (3) to hold are given by Satorra (1989) and others.

As shown by Browne (1982), the elements of Γ are given in the distribution-free case by

$$\Gamma_{ij,kl} = \sigma_{ijkl} - \sigma_{ij}\sigma_{kl}, \qquad (4)$$

where $\sigma_{ij} = E(X_i - \mu_i)(X_j - \mu_j)$ and $\sigma_{ijkl} = E(X_i - \mu_i)(X_j - \mu_j)(X_k - \mu_k)(X_l - \mu_l)$ for random variables $X = X_i,\ldots,X_l$ having means $\mu_i,\ldots\mu_l$. Estimating the mixed fourth-order moments σ_{ijkl} requires a lot of computer time and storage, and the moment estimator $\hat{\sigma}_{ijkl}$ tends to be unstable in small samples. Hence there has been a search for alternatives to (1) – (4) that are more practical, yet retain asymptotic optimality. The two major approaches seek to substitute a computationally simpler and more stable estimator for σ_{ijkl} in (4).

Supported in part by USPHS grants DA0017 and DA01070. This manuscript is based on a paper presented at Interface '91 (Seattle, April 1991). Address reprint requests to P. M. Bentler, Department of Psychology, UCLA, Los Angeles, CA 90024-1563.

Simple Efficient Models

One general alternative approach has been to determine conditions under which the matrix $\Gamma_{\mathcal{N}}$ with elements

$$\Gamma_{\mathcal{N}ij,kl} = \sigma_{ik}\sigma_{jl} + \sigma_{il}\sigma_{jk} \quad (5)$$

can substitute for (4) with no loss of efficiency. This is the form that (4) would take if

$$\sigma_{ijkl} = \sigma_{ij}\sigma_{kl} + \sigma_{ik}\sigma_{jl} + \sigma_{il}\sigma_{jk}\,, \quad (6)$$

which holds when the variables X are multivariate normally distributed, or have no "excess" kurtosis. However, (5) can be used without loss of efficiency with nonnormal data, provided that some conditions on the model and parameters are met. The relevant asymptotic robustness theory has been the object of intensive recent research (e.g., Amemiya & Anderson, 1990; Browne & Shapiro, 1988; Satorra & Bentler, 1990). Although conditions for asymptotic robustness have been developed, in general they are difficult to verify and apply in practice.

Another approach has been to determine conditions under which relatively simple extensions of (6) would hold. As noted by Browne (1982) and Bentler (1983), under multivariate elliptical distributions (e.g., Fang & Anderson, 1990; Shapiro & Browne, 1987)

$$\sigma_{ijkl} = \eta(\sigma_{ij}\sigma_{kl} + \sigma_{ik}\sigma_{jl} + \sigma_{il}\sigma_{jk})\,, \quad (7)$$

where $\eta = \sigma_{iiii}/3\sigma_{ii}^2$ represents the common marginal kurtosis of the $i = 1,\ldots$ p variables. Hence, (1) – (3) can be applied optimally if a consistent estimator $\hat{\eta}$ of η is used in (7) and hence (4). Such estimators are readily available and thus elliptical theory is implemented in standard computer programs (e.g., Bentler, 1989). When $\eta = 1$, the multivariate normal form $\Gamma_{\mathcal{N}}$ of Γ in (5) applies, and computations are simpler still.

In practice, the assumption of homogeneous kurtosis for all p variables as made under both normal and elliptical theories is excessively strong. Thus, Kano, Berkane, and Bentler (1990) proposed the structure

$$\sigma_{ijkl} = (a_{ij}a_{kl})\sigma_{ij}\sigma_{kl} + (a_{ik}a_{jl})\sigma_{ik}\sigma_{jl} + (a_{il}a_{jk})\sigma_{il}\sigma_{jk} \quad (8)$$

where $a_{ij} = a_{ji}$ are parameters arbitrarily selected to assure that Γ with elements given in (4) is positive definite. They proved that use of consistent estimators $\hat{a}_{ij}$ with (8), and in minimizing F in (1), yielded the asymptotic χ^2 goodness of fit test in (2) and the minimum variance estimator with covariance matrix given in (3). In practice, they suggested using the structure

$$a_{ij} = \tfrac{1}{2}(\eta_i + \eta_j), \quad (9)$$

where $\eta_i^2 = \sigma_{iiii}/3\sigma_{ii}^2$. Thus, estimates of p marginal kurtoses, along with covariances, are needed to implement (8). If the marginal kurtoses η_i^2 are equal for all variables, (9) with (8) reduces to (7). Thus this methodology generalizes the approach based on elliptical theory, while requiring no heavier computations.

A Simple Kurtosis Structure

A limitation of (8) was noted by Kano, Berkane, and Bentler (1990). Letting $C = A*\Sigma$ (i.e., $c_{ij} = a_{ij}\sigma_{ij}$), Kano et al. proved that a necessary condition for Γ to be positive definite is that C is positive definite and the η_i are all positive. While the latter condition is not restrictive, if the η_i are highly variable, the structure (9) might not be consistent with a positive definite Γ. Hence (8) would be an inappropriate kurtosis model. For example, with $p = 2$, if $\eta_1 = 1$, $\eta_2 = 10$, and Σ is the 2 x 2 correlation matrix with $\sigma_{12} = .6$, under (9) C would not be positive definite. Here we give an alternative structure for (8) that is more widely applicable than that based on (9).

Let $\eta_i^2 = \sigma_{iiii}/3\sigma_{ii}^2$ as before. In addition, let

$$a_{ij} = \sqrt{\eta_i}\sqrt{\eta_j}\,. \quad (10)$$

Then we have the following result: Under (10), the matrix $C = A*\Sigma$ is positive definite. Clearly, $c_{ij} = \sqrt{\eta_i}\sqrt{\eta_j}\,\sigma_{ij}$, i.e., $C = D\Sigma D$ where D is a diagonal matrix with $d_{ii} = \sqrt{\eta_i}$. Thus, since Σ is positive definite by assumption, so is C for any marginal kurtoses of the variables.

The structure (10) thus extends the applicability of the Kano et al. (1990) theory to a wider range of nonnormal distributions. In particular, (2) and (3) hold, provided that (8) holds with (10). The counterexample based on (9), given above, would not be a problem when based on (10). The structure (10) also represents a generalization of the elliptical structure (7). That is, if $\eta_i^2 = \eta_j^2$ for all variables, substitution of (10) into (8) yields the structure (7). In turn, the normal theory relation (6) is also a special case.

Function Simplification

Kano, Berkane, and Bentler (1990, eq. 11) showed how the function (1) to be optimized can be simplified into a computationally more efficient form under the kurtosis structure (8). Under this structure, the function specializes into a form that avoids computation of the large ($p* \times p*$) weight matrix in (1). Because the proposed structure (10) is used under (8), the same function simplification as described by Kano et. al applies. These authors also showed how a yet further simplification is possible when the model $\sigma(\theta)$ meets a condition of full scale invariance. A different type of simplification is possible under the newly proposed structure (10), if the model meets the ICSF assumption:

The covariance structure $\sigma(\theta)$ is said to be invariant under a constant scaling factor (ICSF) if for any positive number α and $\theta \in \Theta$, there exists a $\theta^* \in \Theta$ such that $\alpha\sigma(\theta) = \sigma(\theta^*)$.

Under the ICSF assumption, Satorra and Bentler (1986) and Shapiro and Browne (1987) showed that $\sigma = \Delta d$, for some vector d. Then, under the kurtosis model (8) with the relation (10), the general matrix Γ defined in (4) can be written in the form

$$\Gamma = 2K_p'(C \otimes C)K_p + cc' - \Delta dd'\Delta' \qquad (11)$$

where K_p' is a known matrix such that $\sigma = K_p'\text{vec}(\Sigma)$, and where $c = K_p'\text{vec}(C)$. Shapiro (1986) obtained the result that if Γ could be expressed in the form $\Gamma = W + \Delta G\Delta'$ for some symmetric matrix G, then under some regularity conditions, at the minimum

$$n(s - \sigma(\hat{\theta}))'W^{-1}(s - \sigma(\hat{\theta})) \sim \chi^2_{(p* - q)} \qquad (12)$$

and the estimator $\hat{\theta}$ is asymptotically efficient. In practice, one uses a consistent estimator $\hat{W}$ of W in (12).

It is apparent that (11) is of the form required for (12) to be applicable, with

$$W = 2K_p'(C \otimes C)K_p + cc'. \qquad (13)$$

Some algebra can verify that the function (12) to be minimized under (13) can be written as

$$F = \tfrac{1}{2}\text{tr}\{[S - \Sigma(\theta)]C^{-1}\}^2 - \delta\{\text{tr}[S - \Sigma(\theta)]C^{-1}\}^2 \qquad (14)$$

where $\delta = (2p + 4)^{-1}$. The advantage of minimizing (14) rather that (1) is that matrices of much smaller order are involved. Also, since (14) is a variant of the form given by Bentler (1983, eq. 3.13) for estimation under elliptical distributions, only minor modifications to standard programs are needed to implement estimation by minimizing (14).

References

Amemiya, Y., & Anderson, T. W. (1990). Asymptotic chi-square tests for a large class of factor analysis models. *Annals of Statistics*, **18**, 1453-1463.

Bentler, P. M. (1983). Some contributions to efficient statistics in structural models: Specification and estimation of moment structures. *Psychometrika*, **48**, 493-517.

Bentler, P. M. (1989). *EQS structural equations program manual.* Los Angeles: BMDP Statistical Software.

Browne, M. W. (1982). Covariance structures. In D. M. Hawkins (Ed.), *Topics in applied multivariate analysis* (pp. 72-141). Cambridge: Cambridge University Press.

Browne, M. W., & Shapiro, A. (1988). Robustness of normal theory methods in the analysis of linear latent variate models. *British Journal of Mathematical and Statistical Psychology*, **41**, 193-208.

Fang, K. -T., & Anderson, T. W. (Eds.) (1990). *Statistical inference in elliptically contoured and related distributions.* New York: Allerton.

Kano, Y., Berkane, M., & Bentler, P. M. (1990). Covariance structure analysis with heterogeneous kurtosis parameters. *Biometrika*, **77**, 575-585.

Satorra, A. (1989). Alternative test criteria in covariance structure analysis: A unified approach. *Psychometrika*, **54**, 131-151.

Satorra, A., & Bentler, P. M. (1986). Some robustness properties of goodness of fit statistics in covariance structure analysis. *Proceedings, Bus. Econ. Stat. Sect., American Statistical Association*, pp. 549-554.

Satorra, A., & Bentler, P. M. (1990). Model conditions for asymptotic robustness in the analysis of linear relations. *Computational Statistics & Data Analysis*, **10**, 235-249.

Shapiro, A. (1986). Asymptotic theory of overparameterized structural models. *Journal of the American Statistical Association*, **81**, 142-149.

Shapiro, A., & Browne, M. W. (1987). Analysis of covariance structures under elliptical distributions. *Journal of the American Statistical Association*, **82**, 1092-1097.

A Method for Controlling Multivariate Kurtosis in the Simulation of Elliptically-Contoured Distributions

Ronald Horswell
Department of Management Science
Ball State University
Muncie, IN 47306

Stephen Looney
Department of Quantitative Business Analysis
Louisiana State University
Baton Rouge, LA 70803

Abstract

In Monte Carlo simulation of multivariate distributions, it is often helpful to use a general class of distributions which share certain defining characteristics but which allow controlled variation of other characteristics. We show how multivariate kurtosis, as measured by Mardia's coefficient, $\beta_{2,p}$, can be controlled across the class of elliptically-contoured distributions. This allows convenient assessment of the effects of kurtosis on test power, robustness, or whatever the Monte Carlo subject of interest. We illustrate the method's utility by showing that common tests for skewness are also very sensitive to kurtosis even in non-skewed distributions.

1 Introduction

Elliptically-contoured multivariate distributions are those whose equal-density countours are ellipses (bivariate case) or hyper-ellipses (for the p>2 case.) Multivariate normal distributions are special cases, and elliptically-contoured distributions provide one approach for organizing departures from multivariate normality. See Chmielewski (1981) for a summary and review of elliptically-contoured distributions and their contributions to robustness studies. Johnson (1987, chapter 6) describes an easily implemented approach for generating elliptically-contoured distributions.

We are here concerned with controlling multivariate kurtosis, as defined by Mardia (1970). For a p-variate distribution, Mardia's multivariate kurtosis coefficient is

$$\beta_{2,p}=E[(\underline{X}-\underline{\mu})^T\Sigma^{-1}(\underline{X}-\underline{\mu})]^2,$$

where $E(\underline{X})=\underline{\mu}$ and $Cov(\underline{X})=\Sigma$. The sample analog is

$$b_{2,p} = (n^{-1})\Sigma_{i=1,n}[(\underline{x}_i-\hat{\underline{\mu}})^T\Sigma^{-1}(\underline{x}_i-\hat{\underline{\mu}})]^2, \text{ with } \Sigma \text{ to order } n^{-1}.$$

For the univariate case, $\beta_{2,1}$ and $b_{2,1}$ become the usual univariate population and sample kurtosis coefficients.

Mardia (1970) also defines population and sample multivariate skewness coefficients, $\beta_{1,p}$ and $b_{1,p}$. These reduce to $(\sqrt{\beta_1})^2$ and $(\sqrt{b_1})^2$ for p=1. Elliptically-contoured distributions are not skewed by any reasonable skewness criterion; i.e., $\beta_{1,p}=0$.

Largely following Johnson (1987, chapter 6), an elliptically-contoured random vector $\underline{Y}$ can be generated via:

$$\underline{Y}_{(p\times 1)} = \mathbf{R}\mathbf{B}_{(p\times p)}\underline{U}_{(p\times 1)} + \underline{\mu}_{(p\times 1)}$$

where

R is a non-negative random variable with finite variance;

$\underline{U}$ is a point uniformly distributed on the unit p hypersphere;

R and $\underline{U}$ are independent.

B is a factorization of $\mathbf{M}=\mathbf{B}\mathbf{B}^T$, with **M** being proportional to Σ.

In this scheme:

$$E(\underline{Y}) = \underline{\mu}, \text{ and}$$

$$Cov(\underline{Y}) = \Sigma = (p^{-1})E(R^2)\mathbf{B}\mathbf{B}^T.$$

The special case of spherically-contoured distributions arises when $\mathbf{B}=a\mathbf{I}_{(p\times p)}$. Thus, a generation scheme for spherically-contoured distributions with $\underline{\mu} = \underline{0}$ and $\Sigma=p^{-1}E(R^2)\mathbf{I}$ is

$$\underline{X}_{(p\times 1)} = R\underline{U}.$$

As the above generation schemes imply, any elliptically-contoured random vector can be obtained by an affine transformation of a spherically-contoured random vector.

2 Controlling Kurtosis

Theorem 1: For spherically-contoured distributions generated as $\underline{X}=R\underline{U}$, if $E[R^2]$ and $E[R^4]$ exist, then $\beta_{2,p} = p^2E(R^4)/[E(R^2)]^2$.

Proof:

We first note that $E(\underline{U}) = \underline{0}$ and $Cov(\underline{U}) = p^{-1}\mathbf{I}$. [See Mardia, Kent, and Bibby, 1979, p. 429, for both results.] It follows that:

$$E(\underline{X}) = \underline{\mu} = \underline{0}, \text{ and}$$

$$Cov(\underline{X}) = \mathbf{\Sigma} = E[R\underline{U}(R\underline{U})^T] = p^{-1}E(R^2)\mathbf{I}.$$

$$\beta_{2,p} = E[(\underline{X}-\underline{\mu})^T\mathbf{\Sigma}^{-1}(\underline{X}-\underline{\mu})]^2 = E\{(R\underline{U}^T[p^{-1}E(R^2)\mathbf{I}]^{-1}(R\underline{U})\}^2$$

$$\beta_{2,p} = p^2E\{[R^2\underline{U}^T\underline{U}]^2\}/[E(R^2)]^2$$

Since $\underline{U}^T\underline{U}=1$ by definition of a point on a unit p-hypersphere,

$$\beta_{2,p} = p^2E(R^4)/[E(R^2)]^2.$$

#

Theorem 2: For elliptically-contoured distributions generated as $\underline{Y}=R\mathbf{B}\underline{U} + \underline{\mu}$, if $E[R^2]$ and $E[R^4]$ exist, then $\beta_{2,p} = p^2E(R^4)/[E(R^2)]^2$.

Proof:

Mardia (1970) shows that if $\underline{Y}$ is an affine transformation of $\underline{X}$, then $\beta_{2,p}(\underline{Y}) = \beta_{2,p}(\underline{X})$. Thus, Theorem 1 also establishes Theorem 2.

#

In application, to generate an elliptically-contoured $\underline{Y}$ with a target covariance matrix, $\mathbf{\Sigma}$, establish kurtosis via selection of the distribution on R. Set $\mathbf{M}=p\mathbf{\Sigma}/E(R^2)$, and obtain $\mathbf{B}$ via a Cholesky decomposition of $\mathbf{M}$.

Choice of the distribution on R, often called the "radius" random variable, controls $\beta_{2,p}$ and also controls all higher-order even moments of the multivariate distribution. Since for establishing kurtosis, the pertinent moments are $E(R^2)$ and $E(R^4)$, it is often more convenient to differentiate among elliptically-contoured distributions in terms of differing distributions placed on R^2.

$R^2 \sim [\Gamma(p/2,\beta)]$ leads to multivariate normality. Distributions which depart from normality in kurtosis but still have unbounded support can be obtained by using other gamma distributions for R^2. Distributions with bounded support and known kurtosis, can be obtained by using a bounded-support univariate distribution on R^2, such as a beta distribution. $\beta_{2,p}$ can be easily determined so long as $E(R_2)$ and $E(R^4)$ are known.

3 Example of Application

Skewness and kurtosis coefficients are commonly used as tests for both univariate and multivariate normality. For instance, Mardia (1970) describes the use of $b_{1,p}$ and $b_{2,p}$ as tests for multivariate normality. However, for both the univariate and multivariate cases, there is also a widespread presumption that these tests are "diagnostic" in the sense that they indicate the nature of departure from normality. Put another way, this amounts to a belief that the skewness tests used to test the normality hypothesis possess an additional valid interpretation as tests of a non-skewness hypothesis. For instance, Mardia (1970) states: "To test $\beta_{1,p}=0$ for large samples, we calculate A [a test statistic based on $b_{1,p}$] and reject the hypothesis for large values of A" (p. 523).

Empirical results, however, do not support the presumption that skewness tests are diagnostic. For instance, here we present Monte Carlo results showing skewness test powers against spherically-contoured distributions, all of which are non-skewed. We use six spherically-contoured distributions generated as $\underline{X}=R\underline{U}$. These are, defined by the univariate distributions placed on R:

SC1: $R \sim [\Gamma(8p,1/8)]^{1/2}$, yielding $\beta_{2,p}=p(p+1/8) < \beta_{2,p}(MVN)$;

SC2: $R \sim [\Gamma(4p,1/4)]^{1/2}$, yielding $\beta_{2,p}=p(p+1/4) < \beta_{2,p}(MVN)$;

SC3: $R \sim [\Gamma(2p,1/2)]^{1/2}$, yielding $\beta_{2,p}=p(p+1/2) < \beta_{2,p}(MVN)$;

SC4: $R \sim [\Gamma(p/2,2)]^{1/2}$, yielding $\beta_{2,p}=p(p/2) = \beta_{2,p}(MVN)$;

SC5: $R \sim [\Gamma(p/4,4)]^{1/2}$, yielding $\beta_{2,p}=p(p+4) > \beta_{2,p}(MVN)$;

SC6: $R \sim [\Gamma(p/8,8)]^{1/2}$, yielding $\beta_{2,p}=p(p+8) > \beta_{2,p}(MVN)$.

Note that distribution SC4 is multivariate normal.

In this study, we used levels of

$n=25, 50, 100$;

$p=2, 5, 10$; and

$\alpha=0.05, 0.10$.

As an example of results, Table 1 reports results only for p=5, α=0.10. Results for other values of p and α=.05 are not qualitatively different. Our power results are all based on 1,000 replications.

The table shows results for four skewness-based tests. The rows referenced as $b_{1,p}(a)$ are powers of Mardia's $b_{1,p}$, with critical values obtained from an (asymptotic) approximate null distribution suggested by Mardia (1970). The $b_{1,p}(e)$ rows are results for $b_{1,p}$ with critical values derived empirically from 10,000 multivariate normal distributions. Q_1 is a skewness test suggested by Small (1980); while b_{1p} is a skewness test suggested by Srivastava (1984). Critical values for Q_1 and b_{1p} were obtained from the asymptotic distributions suggested by those authors. Conceptually, Q_1 tests for skewness in any of the p marginal distributions, while b_{1p} tests for skewness in any of the p principal components. These are, therefore, both more specific and less comprehensive skewness tests than are tests based on $b_{1,p}$.

Results in Table 1 suggest the following conclusions:

1) For distributions with less than normal kurtosis, the skewness tests' detection levels are deflated well below test size.
2) For distributions with greater than normal kurtosis, the skewness tests' detection levels are inflated well above test size. This implies that a "skewness" test has a strong probability of misdiagnosing as "skewed" a nonskewed distribution with high kurtosis.
3) These effects, at least the inflation of detection levels, grow more pronounced as sample size increases. Thus, they do not appear to be small sample properties.

This effect is not isolated or unique to our study. Although the effect has often been overlooked, to our knowledge, empirical (Monte Carlo) studies have been unanimous in demonstrating that typical "skewness" tests have detection levels strongly inflated (deflated) by greater than (less than) normal kurtosis. Furthermore, other studies also suggest this is not a small sample property, but rather an effect that grows more pronounced with increasing sample size. [See Horswell and Looney (1991) for a review of relevant Monte Carlo studies and additional Monte Carlo results of ours along these lines.]

4 Discussion

The poor "diagnostic" properties of skewness-based tests stem from the fact that the tests use normality-based null distributions. Normal distributions are not skewed. However, the sampling distributions of skewness coefficients (or skewness test statistics) differ greatly over non-skewed distributions. For instance, the sampling distributions of $b_{1,p}$ differ greatly across the six spherically-contoured distributions used here. If, for a given n and p, the sampling distribution of, say, $b_{1,p}$ was approximately or asymptotically the same across non-skewed distributions, then a normality-based null distribution of $b_{1,p}$ might be generally useful to test hypotheses of non-skewness. However, this is not the case. A more extensive discussion of this problem appears in Horswell and Looney (1991).

Table 1: POWERS OF SKEWNESS TESTS AGAINST SPHERICALLY-CONTOURED DISTRIBUTIONS p=5

Nominal test size = 0.10
Table entries are per cent of distributions rejected

	n=25	n=50	n=100
SC1			
$b_{1,p}(a)$	0	0	0
$b_{1,p}(e)$	0	0	0
Q_1	0	0	0
b_{1p}	1	0	0
SC2			
$b_{1,p}(a)$	0	0	0
$b_{1,p}(e)$	0	0	0
Q_1	0	0	0
b_{1p}	0	0	0
SC3			
$b_{1,p}(a)$	0	0	0
$b_{1,p}(e)$	0	0	0
Q_1	1	1	0
b_{1p}	1	1	0
SC4=multivariate normal			
$b_{1,p}(a)$	4	7	8
$b_{1,p}(e)$	11	10	10
Q_1	10	12	9
b_{1p}	6	8	10
SC5			
$b_{1,p}(a)$	37	63	81
$b_{1,p}(e)$	58	69	83
Q_1	32	42	48
b_{1p}	22	35	42
SC6			
$b_{1,p}(a)$	88	99	100
$b_{1,p}(e)$	94	100	100
Q_1	72	81	86
b_{1p}	56	71	86

References

Chmielewski, M. A. (1981), "Elliptically Symmetric Distributions: A Review and Bibliography," *International Statistical Review,* 49, 67-74.

Horswell, R. and Looney, S. (1991), "Diagnostic Limitation of Tests for Multivariate Normality Based on Skewness Coefficients," working paper, Louisiana State University.

Johnson, M. E. (1987), *Multivariate Statistical Simulation,* New York: John Wiley.

Mardia, K. V. (1970), "Measures of Multivariate Skewness and Kurtosis with Applications," *Biometrika,* 57, 519-530.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979), *Multivariate Analysis,* New York: Academic Press.

Small, N. J. H. (1980), "Marginal Skewness and Kurtosis in Testing Multivariate Normality," *Applied Statistics,* 29, 85-87.

Srivastava, M. S. (1984), "A Measure of Skewness and Kurtosis and a Graphical Method for Assessing Multivariate Normality," *Statistics & Probability Letters,* 2, 263-267.

Estimation in Highly Skewed Data

Shane P. Pederson

Los Alamos National Laboratory
Los Alamos, NM 87545

Abstract

The problem of inference for the mean of a highly asymmetric distribution is considered. Even with large sample sizes, usual asymptotics (i.e., normal theory) give poor answers, and standard modifications, such as higher moment correction factors, provide little help. We attempt to develop diagnostics to indicate when inferences are likely to be valid, and we examine the performance of several modifications to the standard procedure. The problem is illustrated with data from particle physics.

1. Introduction

When interested in measures of central tendency, robust estimates of location such as the median or the trimmed mean are commonly recommended if data are asymmetric. Occasionally, however, the estimates of the population mean or total are needed. We consider an example from particle physics, in particular from the use of Monte Carlo to simulate neutron transport. As the resulting distributions of these complex processes are rarely known, simulation is used to determine the values of physical quantities, such as average flux passing through a region. Thus the mean value of such a distribution is truly of interest. In this paper, we examine performance of the standard normal-theory based estimator in developing confidence intervals for the mean. We will consider modifications of the standard procedure, as well as compute theoretical moments for the appropriate reference distributions.

2. Nonparametric Confidence Intervals

Efron (1988) has characterized the problem of nonparametric confidence intervals for the mean as follows: "In one sense this problem is impossible, since modifying F with a tiny probability of X being enormous ... can totally change μ without ever showing up in most samples ... On the other hand, the problem is 'solved' every day by using the standard Student t-intervals ..." We will examine modifications to these procedures in the "naive" case in which we assume nothing about the data other than that it is non-negative and independent and identically distributed (i.i.d.).

To generate positively skewed data, we use the absolute value of a Cauchy random variable. It has the density

$$f(x) = \frac{2}{\pi}\frac{1}{1+x^2}, \quad x \geq 0.$$

If we censor at a threshold value T, the resulting random variable will have all moments finite but arbitrarily large.

The standard nonparametric approach to confidence intervals gives intervals with nominal coverage rate $1 - \alpha$ of the form

$$\bar{x} \pm t_{n-1,(1-\alpha/2)}\frac{s}{\sqrt{n}},$$

where $\bar{x}$ and s are the sample mean and sample standard deviation, n is the sample size, and where $t_{n-1,(1-\alpha/2)}\frac{s}{\sqrt{n}}$ is the appropriate percentage point from the t-distribution with n-1 degrees of freedom. When the approximation to the t-distribution is poor the performance of these intervals is degraded. In the case where the underlying random variables are positively skewed, the estimates of mean and variance will be biased low and correlated. This results in intervals that often miss the true mean on the low side. Figures 1 and 2 illustrate this. Figure 1 is a plot of 1000 pairs of $(\bar{x}, s)$, each pair from a sample of size 1000 from a Gaussian random variable with mean 6.5. The envelope formed by the two diagonal lines contains the pairs which result in confidence intervals which cover 6.5; about 95% of the points lie within this envelope, as expected. Figure 2 shows 1000 pairs of the same statistics for samples of size 1000 generated from the absolute value Cauchy distribution, censored at T = 10000 (with mean 6.5). Note first that each axis in in log scale, resulting in curved envelope lines. The distribution of $\bar{x}$ and s is no longer elliptical, and significant biases and correlations are present. In fact, only about 60% of the points result in intervals which cover 6.5. In most cases 1000 is considered a large sample size but here it is clear that the t-approximation is a bad one, and that n is too small.

Modifications to the standard procedure often try to better characterize the distribution of $t = \sqrt{n}\,(\bar{x}-\mu)/s$ (e.g., Johnson 1978). Following Hall (1983), an Edgeworth expansion of the distribution of t can be inverted to obtain a modified confidence interval. The modifications involve 3rd- and higher central moments of the underlying distribution; in the examples considered here, only the first modification (using 3rd-moment terms) improved coverage rates. That modification gives

$$\bar{x} + \frac{1}{6n}\frac{\hat{\mu}_3}{s^2}(1 + 2t^2_{n-1,(1-\alpha/2)}) \pm t_{n-1,(1-\alpha/2)}\frac{s}{\sqrt{n}},$$

where $\hat{\mu}_3$ is the sample 3rd central moment. This interval is the same length as before but is now asymmetric about $\bar{x}$, biased in the direction suggested by the sample skewness.

Table 1, reprinted from Pederson (1991), contains observed coverage rates for the mean of an absolute value Cauchy random variable, censored at 10000. For each of the several sample sizes considered, 800 independent replications were generated. Both the standard (denoted by std) and 3rd-moment-corrected (denoted by 3rd) intervals were computed.

Table 1
Obsevered Coverage Rates
(nominal level = 0.95)

n	Standard	3rd-moment
1000	0.59	0.66
5000	0.74	0.79
10000	0.82	0.84
20000	0.87	0.92
50000	0.93	0.94
100000	0.93	0.94

The standard intervals cover at below the nominal rate for sample sizes less than 50000, and the 3rd-moment corrections are modest at best. The intervals that miss are almost always too low, as expected. These results suggest that standard confidence interval procedures are inappropriate in this problem for samples under 20000 in size, as there is considerble undersampling of the tail regions.

3. Diagnostics

If standard modifications are limited by the sample size and skewness of the problem, one would like to know when such a situation exists. Geary (1947) computed the first four semi-invariants of t, from which moments can be obtained, expressed in terms of the first four moments of the underlying distribution. Pederson (1991) found that the critical quantity in determining the convergence of t to a standard Gaussian random variable is γ, the squared coefficient of variation of s^2. To first order, the variance of t is $1 + \frac{7}{4}\rho^2\gamma$, where ρ is the correlation between $\bar{x}$ and s^2 and is usually near 1 in the cases considered here. Thus when γ is small relative to 1, the variance of t will be near that of a standard normal. For the simulations from Table 1, γ for $n = 20000$ is 0.26, and for $n = 50000$ is 0.11; by the time γ has reached 0.1, coverages are near nominal levels. Unfortunately, an estimate of γ based on sample moments does not appear to be a useful diagnostic, because it is biased low. Work on developing useful diagnostics continues.

References

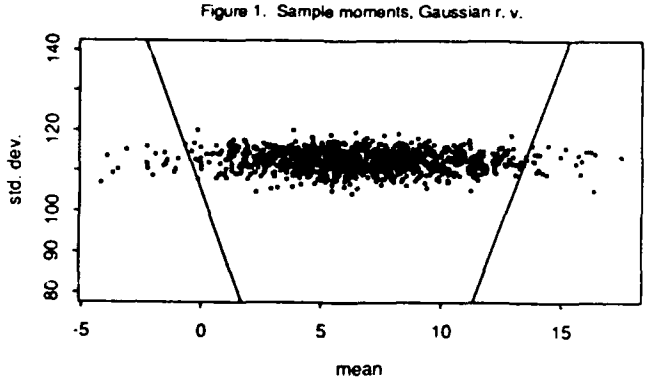
Efron, B. (1981). Nonparametric standard errors and confidence intervals. *Canad. J. Statist.*, 9, 139-172.

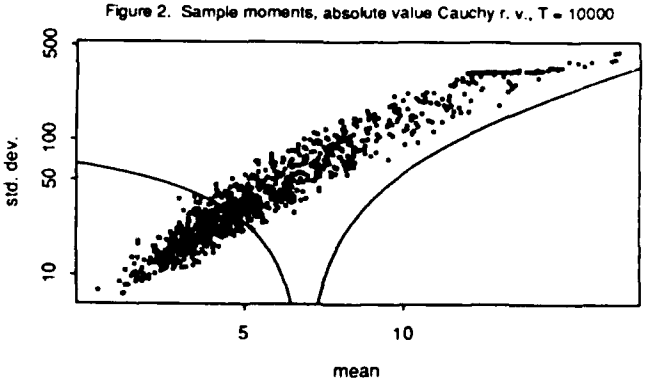
Geary, R. C. (1947). Testing for normality. *Biometrika* 34, 209-242.

Hall, P. (1985). Inverting an Edgeworth expansion. *Ann. Statist.*, 11, 569-576.

Johnson, N. J. (1978). Modified *t* tests and confidence intervals for asymmetrical populations. *J. Amer. Statist. Assoc.*, 73, 536-544.

Pederson, S. P. (1991). Mean estimation in highly skewed samples. *Los Alamos Report*, 91- .





ESTIMATION OF THE MEAN OF POSITIVELY SKEWED DISTRIBUTIONS
with applications to estimation of exposure
to contaminated soils

Ling Chen
Department of Statistics
Florida International University

Robert W. Jernigan
Department of Mathematics and Statistics
The American University

ABSTRACT

We consider estimating the mean of a positively skewed distribution. It has been noted that in random samples the sample mean has a large probability of falling below the mean of the distribution, because of such skewness. Various ad hoc procedures have been proposed to correct this low coverage of the mean in order to estimate conservatively long-term exposure to contaminated soils at toxic waste sites. We propose a direct estimate of the mean based on a penalized empirical loss function. This loss function is made up of a squared error loss plus a penalty for each observation that falls above the estimate. The resulting minimum risk estimate, called the penalized mean, is derived iteratively and shown to be biased in favor of greater coverage.

We show that, asymptotically, a one-step iterate of the penalized mean is unbiased, converges almost surely to the true mean, and with mild assumptions on the form of the penalty, is normally distributed. Based on a penalized loss, we show that this new estimator is uniformly better than thesample mean when sample size is large. The simulation results show that if we choose the penalty constant properly, the new estimator has the same coverage as an upper confidence limit estimator that has been proposed but with less variance and bias.

1. INTRODUCTION

The normal distribution has long been the standard model for the development of statistical theory. Its structure, properties, and centrality in asymptotic theory allows for the construction of an elegant and concise theory of estimation. But experiments and data collection often result in measurements that are inconsistent with an assumption of normality. Statisticians often encounter samples for which a few very large or outlying measurements are included. Many ad hoc procedures have been developed for the practical handling of such outliers, and a debate has often ensued on "whether, and on what basis, we should discard observations from a set of data on the grounds that they are 'unrepresentative', 'spurious' or 'mavericks', or 'rogues', " Barnett and Lewis (1978).

More modern views and analysis have taken a different approach. Rather than modifying the data to fit the prescribed assumptions of the normal theory, much work in the past twenty years has gone into the development of estimation procedures and statistical tests that are either resistant to the effects of these outliers or robust, that is relatively unaffected, by the lack of adherence to underlying assumptions. These techniques are "becoming a core component of statistical practice," Hoaglin, et al.(1985).

Tukey(1962) proposed that outliers could be explained through the use of "longer-tailed" distributions as underlying models. Numerous robust estimators of location in symmetric distributions have been proposed based on unequal weighting. In Huber(1981) and Hampel, et al.(1986) trimmed means, M-estimators, L-estimators, and reweighted estimators have been developed using theoretical and empirical approaches.

Fuller(1970, 1991) has investigated simple estimators for the mean of a skewed population using a technique suggested by Charles Winsor and studied in Tukey and McLaughlin(1963) and Dixon and Tukey(1968). In this technique the largest k observations are replaced by the (k+1)st largest observation and similarly for the smallest observations. The mean of the resulting sample was called by Tukey a "Winsorized" mean. Fuller studied this estimator assuming that the right tail of the distribution function could be well approximated by the tail of a Weibull distribution.

A problem arises when using a sample mean to estimate the mean of long-term exposure to contaminated soils at toxic waste sites under the EPA's Superfund program. Since the underlying distribution is positively skewed, the sample mean has a large probability of falling below the population mean. This results in a consistent under-estimation of the population mean. The EPA Office of Emergency and Remedial Response (OERR) convened a Workshop discussion on February 23, 1990, to examine methods for solving this under-estimation problem. In the workshop, many approaches were proposed such as stratifying the data or interpolating the data using kriging, polygon methods or triangle methods. Advantages and disadvantages of these methods were widely

discussed in the Workshop. But no final conclusion was made. The under-estimation problem was left open. An upper confidence limit estimator based on normal theory, that is, $UCL = \bar{X} + 1.96sdv(\bar{X})$, was temporarily put into use.

In order to correct this under-estimation problem, in this paper, we propose a direct estimate of the mean based on a squared error loss plus a penalty for each observation that falls above the estimate. In attempting to minimize the average of such penalized losses we derive a new estimator of the mean of a positively skewed population with adequate coverage, where the coverage of an estimator defined as the probability of the estimator greater than the estimated parameter, that is, $P(\hat{\theta} > \theta)$.

2. A New Criterion ——— Penalized Loss

We define a new criterion, that is, penalized loss, as follows:

$$L(d, \theta) = (d - \theta)^2 + \lambda_n I_{(d<\theta)} \qquad (2.1)$$

where $\lambda_n > 0$, $\lambda_n = o(1)$ and θ denotes the true mean in the population.

The first term of the loss function is the square error loss. We define the second term as a lack of coverage loss. We define 1 to be the penalty constant. We penalize the estimate if it is less than θ.

Let $T(X)$ be the estimator of θ. Then the risk of this estimation, i.e. the average loss, is

$$R(\theta) = E(T(X) - \theta)^2 + \lambda_n P(T(X) < \theta). \qquad (2.2)$$

This risk function consists of two terms. One is the mean square error term and the other is the penalty term. Minimizing the mean square error pulls the estimator towards θ. Minimizing the penalty term pulls the estimator above θ. Hence, this risk is a kind of balance of the variance, bias and coverage. Minimizing this risk is difficult for the nonparametric problem considered here. We do not know the form of the underlying distribution. We only know that the underlying distribution is positively skewed.

In order to find an estimate with small risk under penalized loss, we define a penalized empirical loss function based directly on the sample.

3. A Penalized Empirical Loss Function

The risk defined in equation (2.2) includes a term for a square error loss as well as a penalty term for lack of coverage. A sample based approach to measuring similar ideas can be developed from an empirical loss, like that seen in maximum likelihood or M-estimation.

Define an empirical loss function based on sample as follows:

$$L^*(x_i, t) = (x_i - t)^2 + 2\lambda_n P(t < X < x_i). \qquad (3.1)$$

Here, the penalty we impose is proportional to the probability that X falls between the observation and the estimate, t. The loss defined in (2.1) imposes a penalty if the estimate falls below the parameter θ. A sample based approach to this penalty would impose the penalty whenever an observation falls above the estimate, t.

Based on the form of the penalty term in (3.1), the more extreme an observation, the greater the penalty it imposes on our estimate. To minimize the penalty, the estimator will tend to be larger, and thus have a small probability of falling below θ. With this sample based loss, the penalty is based on the underlying distribution of the population and not on the distribution of the unknown estimator. The constant 2 used in the penalty term is only for the convenience of calculation.

The empirical average loss, i.e. the empirical risk, is

$$R(t) = 1/n \sum L^*(x_i, t)^2$$
$$= 1/n \sum (x_i - t)^2 + 2\lambda_n \, 1/n \sum P(t < X < x_i). \qquad (3.2)$$

We hope to find an estimator of θ based on minimizing the empirical risk (3.2) such that it has small risk under the penalized loss (2.1) but has adequate coverage.

We call the empirical risk in (3.2) a penalized empirical risk.

4. Penalized Mean and Its Large Sample Properties

In order to find the minimum empirical risk estimator of the mean, we have proved several properties of this penalized empirical risk as follows:

(1) $R(t)$ is a continuous function of t.

(2) $R(t)$ is piecewise differentiable. The derivative does not exist at $t = x_i$, but $R'(x_i)$ and $R'(x_i^+)$ have the same sign, $i = 1,...,n$.

(3) If $|f'(t)| \le M$, then the minimum value of $R(t)$ exists and is unique.

Based on these results we have the following theorem.

Theorem 4.1. Suppose $|f'(t)| < M$, then the empirical risk $R(t)$ is minimized by the solution to the equation

$$t = \bar{x} + \lambda_n f(t)(1 - F_n(t)), \qquad (4.1)$$

if it exists.

The solution of the equation (4.1) is not the minimum risk estimator of θ, since the equation (4.1) includes unknown pdf of the population. In order to find the estimator we are looking for, we substitute a density estimator $\hat{f}(t)$ into equation (4.1), and define a new estimator of the mean as follows.

Definition 4.1. If X 's are i.i.d. from a positively skewed distribution, then

$$\hat{\theta} = \bar{X} + \lambda_n \hat{f}(\hat{\theta})(1 - F_n(\hat{\theta})) \qquad (4.2)$$

is called penalized mean, where $\hat{f}(\cdot)$ is a density estimator of X, $F_n(\cdot)$ is the empirical distribution function of X and λ_n is the penalty constant.

Since equation (4.2) defining $\hat{\theta}$ includes an estimate of the underlying density function, it may appear that this density estimate would provide adequate information to estimate θ, but this is not so. We need an estimate of the density function

only at one point which we approximate by $\hat{f}(\hat{\theta})$.

The penalized mean is defined recursively. So it can be obtained iteratively. The recursive algorithm is the following.

Given an initial estimate θ_0,

$$\hat{\theta}_{k+1} = \bar{X} + \lambda_n \hat{f}(\hat{\theta}_k)(1 - F(\hat{\theta}_k)), \quad k \geq 0. \tag{4.3}$$

It is not difficult to prove the following theorem.

Theorem 4.2. Let $\hat{f}(x)$ be a density estimator of f(x). Assume that

(i) $\| \hat{f}_n - f \| \longrightarrow 0$, as $n \longrightarrow \infty$;

(ii) $\lambda_n \longrightarrow 0$, as $n \longrightarrow \infty$

and $\hat{\theta}_n$ be the solution of (4.2). Then we have

(1) $\hat{\theta}_n$ is an asymptotically unbiased estimator of θ;

(2) Assume $\lambda_n = O(n^{-1/2})$, we have $E(\hat{\theta}_n - \theta)^2 \leq O(n^{-1})$;

(3) $\hat{\theta}_n \xrightarrow{p} \theta$;

(4) $\hat{\theta}_n \xrightarrow{a.s.} \theta$;

(5) Assume $\lambda = o(n^{-1/2})$, then $\sqrt{n}(\hat{\theta}_n - \theta) \xrightarrow{L} N(0, \sigma^2)$.

5. One-Step Iterate of the Penalized Mean

Recall the definition of the penalized mean. It defines the estimator in terms of itself. Therefore, the penalized mean forms the basis for a recursive algorithm. If we choose $\bar{x}$ as a starting point, the simulation results show that most of improvement of the estimate is after one step. We define the one-step interate of penalized mean as

$$\hat{\theta} = \bar{X} + \lambda_n \hat{f}(\bar{X})(1 - F_n(\bar{X})). \tag{5.1}$$

Under the same assumptions given in Theorem 4.2 and by using the same approach as we used for the penalized mean, we have

(1) $\hat{\theta}$ is an asymptotically unbiased estimator of θ;

(2) $\hat{\theta}$ converges to θ almost surely;

(3) if we assume $\lambda_n = o(n^{-1/2})$, $\hat{\theta}$ is asymptotically normally distributed with mean θ, and variance σ^2;

(4) $\hat{\theta}$ has better coverage than $\bar{X}$.

And comparing with this estimator, we have proved $\bar{X}$ is inadmissible in a class of nonnegatively skewed distributions under the penalized loss in (2.1).

6. Simulation Results.

We have found a point estimator of the population mean for positively skewed distributions, the one-step iterate of penalized mean, under the penalized loss. But choosing the penalty constant is still a knotty problem. In point estimation problems, one of the traditional ways to evaluate the goodness of a point estimator is to check unbiasedness of the estimator. Now the problem we face is how to evaluate the goodness of a biased estimator with adequate coverage. Instead of constructing a frequency distribution of the estimates obtained : repeated sampling and noting how closely the distribution centers about the population mean, we suggest checking the proportion of the estimates obtained in repeated sampling that fall in a given error range. We evaluate the goodness of our estimators based on the following four different coverages.

$$p0 = P\left(\frac{\sqrt{n}(\hat{\theta} - \theta)}{\sigma} > 0\right). \tag{6.1}$$

$$p1 = P\left(-1/n < \frac{\sqrt{n}(\hat{\theta} - \theta)}{\sigma} < 1\right). \tag{6.2}$$

$$p2 = P\left(-1/n < \frac{\sqrt{n}(\hat{\theta} - \theta)}{\sigma} < 2\right). \tag{6.3}$$

$$p3 = P\left(-1/n < \frac{\sqrt{n}(\hat{\theta} - \theta)}{\sigma} < 3\right). \tag{6.4}$$

Note p0 is the same as the coverage defined in introduction, and pi, i = 1, 2, 3, are the probabilities of the estimator falling i standard deviations of X above the true mean with an error of the order of 1/n on the left side of the mean.

In order to slow down the convergence of the penalty term, we choose

$$\lambda_n = \frac{c}{\sqrt{n}\log\log(n)} \tag{6.5}$$

in (5.1), and choose c such that our estimator has the same P0 as UCL.

Here we only show the simulation results for lognormal(0,1) case. The similar results hold in other common positively skewed distributions.

Table 6.1 shows that the pmean has better coverages than $\bar{X}$, and has almost the same p0 coverage as UCL. The p1 and p2 coverages of pmean are much greater than those of UCL. Pmean also has less mean square error than UCL. Note that coverages p0 and p3 of pmean in all cases are almost the same. This means there is no long tail on the right side of the distribution of pmean. Obviously, the pmean estimator is superior to the UCL estimator. But the optimal penalty constant c is still under investigation.

Table 6.1
Lognormal (0,1) Case
(100 repetitions)

n = 25 est.	p.c.	MSE	P0	P1	P2	P3
$\bar{X}$		.1845	.4000	.3200	.4200	.4300
UCL		1.099	.8600	.2800	.5700	.6900
pmean	20	.2424	.8600	.5300	.8000	.8700
n = 35 est.	**p.c.**	**MSE**	**P0**	**P1**	**P2**	**P3**
$\bar{X}$		.1123	.4100	.3100	.4100	.4200
UCL		.5235	.8300	.2500	.5200	.6700
pmean	25	.1601	.8300	.4800	.7800	.8400
n = 50 est.	**p.c.**	**MSE**	**P0**	**P1**	**P2**	**P3**
$\bar{X}$		.0930	.4300	.3100	.4000	.4200
UCL		.5446	.9200	.3100	.5500	.8200
pmean	36	.1497	.9100	.4900	.8600	.9200

NOTE: (1) p.c. is the penalty constant c in (6.5);
(2) pmean is the one-step iterate of penalized mean defined in (5.1);
(3) UCL = $\bar{X}$ + 1.96sdv ($\bar{X}$)

References

Barnett, V. and Lewis, T. (1978). "Outliers in Statistical Data", 2nd edition, John Wiley, New York.

David C. Hoaglin, Frederick Mosteller & John W. Tukey (1985). "Exploring Data Tables, Trends, and Shapes", John Wiley, New York.

Dixon, W.J. & Tukey, J.W. (1986). "Approximate Behavior of the distribution of Winsorized t', Technometrics 10, p. 83-98.

Fuller, W.A. (1970). "Simple Estimators for the Mean of Skewed Populations", Report to the U.S. Bureau of the Census, Iowa State University, Ames, Iowa.

Fuller, W.A. (1970). "Simple Estimators for the Mean of Skewed Populations", Statistica Sinica, 1, p.137 - 158.

Huber, P.J. (1981). "Robust Statistics", John Wiley, New York.

Hampel, F.R., Ronchetto, E.M.., Rousseeuw, P.J. and Stahel, W.A. (1986). "Robust Statistics: The Approach Based on Influence Functions", John Wiley, New York.

Tukey, J.W. (1962). "The Future of Data Analysis", Ann. Math. Statist. 33, p.1-67.

Tukey, J.W. and McLaughlin, D.H. (1963). Less Vulnerable Confidence and significance Procedures for Location Based on a Single Sample", Sankha Ser. A 25, p.331-352.

DIAGNOSTIC CLASSIFICATION OF IMAGES

Michael L. Goris
Division of Nuclear Medicine
Stanford University
Stanford, CA 94305

Abstract

Diagnostic classifications based on the estimation of norm deviations are based on homomorphic features between images and pathology.
The measurement is the three dimensional mapping of a tracer distribution, which reflects regional myocardial blood flow. This measured distribution or image is compared to a normal distribution, derived from measurements in subjects from a population in which the myocardial perfusion is assumed to be normal. In addition to the normal (average) distribution, a measure of natural variation is made.
The comparison between a test case and the normal population distribution leads a measure of deviation from the norm. The degree of deviation is quantitatively diagnostic only if larger distribution indicate either more advanced disease or a higher probability of disease.
KEY WORDS : Quantitative classification; homomorphism;

1 INTRODUCTION

In the classical approaches, medical images are interpreted visually. Abnormalities are detected by comparing the findings with a virtual image of normal cases. This comparison, to be effective, must take normal variation into account.
In great part, normal variation is due to biological variation in size and shape of normal subjects.
Formal quantification in Nuclear Medicine has usually been restricted to dynamic parameters extracted from dynamic images. In this approach, the image is used merely to define sampling regions, mostly on the basis of pattern recognition (e.g. the generation of a time activity curve from a region of interest drawn over the renal region). On the other hand, quantification of spatial distributions has been hampered by the inability to spatially match untransformed images.

2. MATERIALS AND METHODS

The data are tomographic images obtained after the intravenous injection of thallium chloride at the end of a stress test. Since thallium is a diffusible intracellular tracer (a potassium analog), for a short, but appreciable time following the injection, its distribution in tissues is proportional to the relative distribution of cardiac output in those tissues (1).
The image is a three-dimensional mapping of the spatial distribution of the tracer in the myocardium. The sampled value is the maximum pixel value found across the myocardial region.
To minimize biological variation due to size and shape, and, thus to be able to define corresponding points, the images undergo a polar transform as follows: The origin of the polar transform is located in the cavity of the left ventricle. The latitude in the volume image is represented by the angle P which has its origin at six o'clock (or south

pole or apex in the reoriented image), and is mapped in the vector as r = SQRT[(x-32)**2 + (y-32)**2], where P = r*64/135.

The longitude is represented by the angle **Th**, which has its origin at 9 o'clock in the volume image, or east (septal) , and maps in the vector as itself, again with the origin at 9 o'clock.

The sample value found in the volume image along the radius (P,Th) is located in the vector at x = r*cos(Th) + 32, y = r*sin(Th) + 32.

Radii are sampled for 0<P<135, and 0<Th<360. This sampling follows a registration of the image such that the most proximal part of the basis of the heart lies along the radius at P = 135 for any value of Th., and such that the radius with angle P = 0. goes through the apex, and represents the long axis of the heart.
If one wanted to map each radial sampling value one-to-one in the vector, one would be allowed only 32 radii for P (over 135 degrees), and a variable amount of discrete values of Th, with the maximum being 256 at Th = 135, and the minimum of 4 at P = 135/64. Since this would lead to undersampling in the volume image, the mapping has to be many to one.
This requires a special strategy. Classically, the sampling value is the maximum value along the radius. Many-to-one mapping needs special accommodation. The initial resident value at (x,y) is -1. If the vector (P,Th) which maps in (x,y) has a sampling value of A < the resident value at (x,y) AND >= 0, the A is substituted for the resident value at (x,y).
The vector A(x,y) allows inter patient comparison, since it contains only directionality, but no information on size, and minimal information on shape.
It follows that one can construct an average vector Av(x,y) and a standard deviation S(x,y) from patients in the control group (2).
Comparison of a test case with the normative vectors consist in computing
D(x,y) = (Av(x,y)-A(x,y))/S(x,y)
for each point in the vector A, and a sum (SS) of D(x,y) which represents the global deviation from the normal in the test case.
The transformation and sampling result in considerable information loss. One still needs to prove that SS is a quantitative diagnostic measure.
One approach is to adapt Bayes' theorem. In general if Se is the sensitivity, Sp the specificity, P the prevalence and PP the positive predictive value, then PP=Se.P/(Se.P + (1.-P)(1.-Sp)).
But, this formulation presumes that the sign or symptom is either present or absent, and not quantifiable.
However, if one defines the values Sp and Se for increasing values of SS (see above), then the measure is diagnostically quantitative if the PP does increase with increasing values of SS. Alternatively the sensitivity must decrease or stay constant, and the specificity must increase (if the latter) or increase or remain constant (if the former).
The test population consists of a cohort of serial patients stratified for risk of coronary artery disease. The stratification, as described in the work of Diamond and Forrester (3) , is based on the patient's age and sex, the nature of the symptoms, and the degree of ST segment depression. Furthermore,

the patients are classified as well-tested, if the end-point of the stress test was reached (either 85% of maximum predicted heart rate, ST segment depression, blood pressure drop or significant arrhythmia.
The patients are grouped according to this classification in 9 groups, according to the disease prevalences.
The prevalence of the symptoms [P(S)] in any population can be expressed as the weighted sum of the prevalence in those who have, and those who do not have the disease:
P(S) = Se.P + (1.-P).(1.-Sp)
This equation can be rearranged to read:
P(S) = P.[Se-(1.-Sp)] + (1.-Sp)
If one sets P(S) = y; P = x ; [Se+(1.-Sp)] = a ; and (1.-Sp) = b, the we obtain the regression equation:

$$y = a.x + b.$$

For each of the 9 groups one defines y as the frequency of positive outcomes.
The sensitivity is given by (a + b) and the non-specificity by b, the coefficients from the regression analysis of the paired observations x and y.

3. RESULTS AND DISCUSSION

The results on a cohort of 135 cases confirms the hypothesis. With SS = 0., Se = 0.92, and Sp = 0.60. When SS = 10., Se = 0.74 and Sp = 0.93. The linear relationship measured in the regression analysis is maintained until Sp = 1.00.
At that time one would expect that higher values would detect more advanced disease, but since the analysis is based on relative distributions, local abnormalities tend to be less visible in the presence of global disease. Homomorphy is present for the probability of disease only. Unless algorithms can be devised to match images by elastic deformation of the coordinate systems, one needs to achieve quantitative image analysis following a transformation which eliminates size and shape information.
More importantly, however, numerical extraction becomes quantitative only if higher deviations from the norm have higher positive predictive values.

1. Sapirstein LA (1958): Regional blood flow by fractional distribution of indicators. Am J Physiol 193:161-168.
2. Goris ML, Boudier S, Briandet PA (1987): Two-dimensional mapping of three-dimensional SPECT data: A preliminary step to the quantification of thallium myocardial perfusion single photon tomography. Am J Physiol Imaging 2:176-180.
3. Diamond GA, Forrester S(1979): Analysis of probability as an aid in the clinical diagnosis of coronary artery disease. N Engl J Med 309:518-522.
4. Goris ML, Bretille J, Askienazy S et al (1989): Validation of diagnostic procedures on stratified populations: Application on the quantification of thallium myocardial perfusion scintigraphy. Am J Physiol Imaging. 4:11-15.

PARALLEL COMPUTING: A TUTORIAL FOR STATISTICIANS *

William F. Eddy and Mark J. Schervish
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213

The purpose of this article is to provide statisticians a brief introduction to parallel computing. We begin by discussing a few basic notions which are fundamental to parallel processing. Next some important aspects of hardware for parallel computers are reviewed. We then provide a brief analysis of system performance including a statistical approach to the performance of one kind of distributed computing system. Next there is a discussion of a particular form of parallel iteration which we have found generally useful followed by a discussion of several statistical applications. We conclude with a review of some of the difficulties of programming parallel systems and mention one programming system we have used which helps overcome these problems. We recommend Bertsekas and Tsitsiklis (1989) for the reader who is interested in further details on many of these topics.

1 Fundamentals

There are a small number of key issues that are necessary for the understanding of parallel computing:

1. Sequential processes,
2. Synchronization of parallel processes, and
3. Interprocess communication.

1.1 Processes

A fundamental notion required for the understanding of parallel processing is the notion of a sequential process. A sequential process is the actual execution of a sequential program. A sequential program specifies the sequential execution of a list of program statements. Note that here we are using the terms process and program in a slightly different way than is customary. For our purposes here a program (and the process executing it) might only consist of a small number of or even a single instruction.

A parallel program specifies the execution of one or more sequential programs that can be executed as parallel processes. Parallel processes can be implemented in one of three ways.

- Multiprogramming: The processes can execute on a single processor from a single memory.
- Multiprocessing: The processes can execute on distinct processors but share a common memory.
- Distributed processing: The processes can execute on distinct processors each with its own memory.

Multiprogramming is merely the traditional time-shared version of "parallel" processing; each user seems to have a private machine but is, in fact, sharing a single machine with many others.

The execution path of any program, parallel or not, can be represented by an acyclic directed graph called the process flow graph. Each node in the graph represents a process and each directed arc represents a dependency in the calculation. An arc from node i to node j means that the process at node j requires the result of the process at node i.

1.2 Synchronization

Synchronization of parallel processes is required in order to properly execute the process flow graph of a computation. This controls the cooperation/interference of one process with another.

Consider, as an example, the standard Jacobi iteration for the solution x of a homogeneous linear system

$$x^{(i+1)} = Ax^{(i)}$$

where A is a square matrix. If there is one processor for each row of A then all processors must complete the calculation of the dot product $a_j x^{(i)}$ (where a_j is the jth row of A) before

*This work was partially supported by ONR Contract N00014-91-J-1024 and NSF Grant DMS88-05676

any processor can start the next iteration. The processors must be synchronized at the end of each iteration.

There are several standard programming techniques for implementing process synchronization:

1. Shared variables,
2. Semaphores, and
3. Data flow.

A more detailed discussion of synchronization with programs implementing some of the methods is given in Eddy (1986).

1.3 Communication

Communication among the processes in a parallel program is handled either by the use of common memory (in a shared memory system) or by the use of a communication network (in a distributed memory system). Typically, if the system uses shared memory then communication is handled through *shared variables* stored in the common memory. Those variables are usually accessed with hardware instructions such as "load and lock" and "store and unlock." In a distributed memory system communication is handled by means of *message passing*.

One important aspect of communication is whether or not the communications are synchronized. Synchronization of interprocess communications in a shared memory environment is handled through the use of the memory locking mechanism indicated above. Synchronization in a distributed memory environment is typically handled through the use of I/O instructions which "wait until completion."

A further complication is that synchronization can be different at each end of the communication channel and at different times. Each time an actual read or write is issued, it can include an implied "wait until completion" or not. In the case of asynchronous writes, one problem is that a large number of writes may be issued without a corresponding number of reads. Consequently, the receiving process must have available a nearly unlimited amount of buffer space to store these messages until the receiving process is prepared to read them.

2 Hardware

2.1 Flynn's Taxonomy

Flynn (1966) introduced terminology for models of computation which has become standard, although it is unfortunately imprecise. Flynn's scheme is cross-classification of hardware on the basis of two attributes:

1. whether the machine can process more than one instruction simultaneously;
2. whether the machine can process more than one data item simultaneously.

The four resulting types of machines are:

	DATA	
INSTRUCTION	Single	Multiple
Single	**SISD**	**SIMD**
Multiple	**MISD**	**MIMD**

The SISD machines are traditional sequential computers, often called von Neumann machines after their designer. The SIMD machines occur in a number of varieties, the most important being the vector processors such as the Cray and the systolic machines. There are really no practical MISD machines. The MIMD machines occur in two varieties, both of which are very important:

1. the distributed memory machines such as the hypercubes and other networks of processors with local memory, and
2. the shared memory machines such as the multiprocessor VAXes and the Cray XMP and YMP machines.

2.2 Memory Hierarchy

An important part of the design of parallel computers is controlling the flow of data to and from the various components of storage. We naturally think of a hierarchy relating speed of access and capacity of these storage elements. Similarly, a critical problem in the development of parallel algorithms for a particular hardware environment is the placement of data items at various levels in the memory hierarchy. While the precise elements of the hierarchy can vary substantially from machine to machine, a fairly general list of the available levels includes

1. CPU registers,
2. cache memory,
3. local memory,
4. distributed memory,
5. disk memory, and
6. off-line storage.

As one proceeds down the hierarchy, data items take ever and ever greater amount of time for access but are simultaneously available in greater quantity. Thus, for example, the most rapidly accessible items are those stored in the CPU registers but there is a very limited number of them. On the other hand, data stored off-line is only accessible after a considerable wait but there is an essentially unlimited amount of such storage available.

2.3 Examples

In the actual presentation at the meeting a variety of different hardware systems were described. These included

- bus-connected systems,
- cross-bar switch based systems,
- mesh-connected systems,
- shuffle-exchange networks,
- hypercube connections, etc.

Space limitations preclude that discussion here. The interested reader might consult Dongarra et al. (1991) for similar examples or Duncan (1990) for a more technical survey.

3 Performance

3.1 Amdahl's Law

Amdahl's law concerns the basic fact that not all parts of a calculation are equally amenable to processing in parallel. Assume that the execution of a program requires M total work and that a certain fraction f of this work can be done at a speed of S^{-1} (sequential) and the remainder can be done at a speed of P^{-1} (parallel). The total time T required to complete the program is

$$T = f \cdot M \cdot S + (1 - f) \cdot M \cdot P.$$

Consequently, the effective speed R with which the program is executed is given by

$$R = M/T = \frac{1}{f \cdot S + (1 - f) \cdot P};$$

this is Amdahl's law. The critical implication of Amdahl's law is that the time required for the execution of any parallel program is bounded below by the time required to execute the sequential portion of that program even if the parallel portion is executed infinitely fast. That is,

$$T > f \cdot M \cdot S.$$

3.2 Load Balancing

There are two possible views of performance in a parallel system. One view (that of an individual user) desires to complete a single job in the shortest possible time (minimizing the makespan). The other view (that of a system manager) desires to keep all the processors of the system as busy as possible.

4 Statistical Issues

There are some statistical issues that arise in the study of distributed system performance. These relate to the processors and communication channels often being in use by other applications besides the one of interest as well as other unpredictable features of the hardware and software. There are two natural ways to study the stochastic performance of distributed systems. One is to create statistical models for the performance of the system and the other is to collect data on the actual performance of a system.

4.1 Statistical Models

Suppose that we wish to minimize the expected time to completion of a task (the *makespan*). We need to divide the work among the processors in some optimal fashion.

4.1.1 A No-Cost Model

Consider the following simple model to start. There is a task of "size" 1 unit and there are p processors among which we can divide the task. Suppose that the task can be divided into smaller subtasks whose sizes add to 1 in such a way that the time it takes to complete a subtask of size β is an exponential random variable with mean β. Suppose that communication is fast enought so that there is no time lost between subtasks. Suppose the task is divided into $n > p$ equal subtasks of size $\beta = 1/n$, and one subtask is assigned to each processor. Each time a processor completes a subtask, another one is assigned until the computation completes. Under these assumptions, the makespan can be calculated as follows:

$$\frac{1}{p} + \frac{1}{n}\sum_{k=1}^{p}\frac{1}{k} \doteq \frac{1}{p} + \frac{\ln p}{n} \qquad (1)$$

This is minimized if n is chosen as large as possible. This is clearly nonsense in any real application.

Consider next, the case where the number of subtasks n is fixed and suppose we are interested in choosing which n parts of the task to make into the n subtasks. For example, if we have two processors and we can only divide the task into three subtasks, we can still choose the sizes of the subtasks. Retain, for now, the zero cost assumption. If we make all three subtasks the same size, then the makespan is 2/3 from (1). If instead, we make the sizes of the three subtasks $1/\alpha_i, i = 1, 2, 3$, we can still calculate the makespan. The completion times for the messages are exponential random variables with natural parameters α_i. It follows that the makespan is

$$\frac{1 + \left[\frac{\alpha_1}{\alpha_2} - \frac{\alpha_1}{\alpha_2+\alpha_3}\right] + \left[\frac{\alpha_2}{\alpha_1} - \frac{\alpha_2}{\alpha_1+\alpha_3}\right]}{\alpha_1 + \alpha_2} + \frac{1}{\alpha_3}. \qquad (2)$$

For each value of α_3, (2) is minimized at $\alpha_1 = \alpha_2 = \alpha$, say. In this case, the makespan is $(2\alpha^2 - 5\alpha + 5)/(2\alpha^2 - 2\alpha)$, which, in turn, is minimized at

$$\alpha = \frac{5 + \sqrt{10}}{3} \doteq 2.721.$$

The minimum makespan is then $\sqrt{10} - 5/2 \doteq 0.6623$, which is not a great improvement over the equal subtask solution ($\doteq .6667$).

In general, the optimal division into n subtasks will have makespan less than the division into n equal subtasks as in (1). With exponential distributions, as $n \to \infty$, the makespan converges to $1/p$, the minimum possible value. Consequently, for these cases we believe that division into optimal size subtasks will not substantially improve upon the simpler division into equal subtasks.

4.1.2 A Random Cost Model

We could assume that each subtask carries an overhead c such that the time to complete each subtask of size β is an exponential random variable with mean $c + \beta$. For simplicity, we will assume that all subtasks are the same size, so that, if there are n subtasks, $\beta = 1/n$. Suppose that there are p processors. The first subtask to finish takes time equal to the minimum of p exponential random variables, so its distribution is $exp(p/[c + \beta])$. The memoryless property of exponential distributions gives that the time between when the $i-1$st and ith subtasks finish also has $exp(p/[c + \beta])$ distribution for $i = 2, \ldots, n-p+1$. For $i = 1, \ldots, p-1$, the time after subtask $n-i$ finishes until subtask $n-i+1$ finishes has $exp(i/[c + \beta])$ distribution. The makespan is then

$$(c + \beta)\left[\frac{n-p+1}{p} + \sum_{i=1}^{p-1} \frac{1}{i}\right]$$

$$= \left(c + \frac{1}{n}\right)\left[\frac{1}{2} + \cdots + \frac{1}{p} + \frac{n}{p}\right].$$

This is minimized at $n = \sqrt{pA(p)/c}$, where $A(p) = \sum_{i=2}^{p} 1/i$.

Of course, the memoryless property of the exponential distribution makes it an implausible model for running times of fixed subtasks. Other factors, such as network traffic and predictable patterns of usage in a distributed system also make the assumptions of this model seem unrealistic.

4.2 Empirical Study

If one is concerned about the performance of a distributed system, but is not confident in any of the simple statistical models which one can construct for their performance, one can collect data on the performance of the system by giving it various problems to solve which are similar to those for which one will want to use the system in the future. One can vary the size and nature of the problem, the number and types of processors, and the sizes of subtasks. As an example, Eddy and Schervish (1986) report on a case for which the application could be made arbitrarily large and for which the subtasks could be made very small. Two different configurations of distributed system were used, one containing eight processors and the other 15 processors. (The systems were heterogeneous in that there were three different kinds of CPU represented among the 15 nodes.) Figure 1 is a plot of the times to completion of many runs using these two systems vs. the natural logarithm of the number of subtasks (messages). We see that the time to completion is relatively insensitive to the number of subtasks within a certain range, but when the number of subtasks gets very large, communication bottlenecks cause inefficiencies. When the number of subtasks gets too small, excessive time is lost waiting for the last subtask to complete.

5 Asynchronous Iteration

Consider an iterative method in which each iteration is a substantial computation. A typical example is the solution of a fixed point problem by successive substitution, where each evaluation of the function is time consuming. If the evaluation of the function can be broken into subtasks, each iteration can run on a distributed system. Since the subtasks are performed asynchronously, these methods are called *asynchronous iteration*. A theoretical problem arises concerning the convergence of such an iterative method. Each "iteration" of such an asynchronous algorithm is not the same as an iteration of the corresponding synchronous algorithm. For example, consider the following iterative method for finding the largest eigenvalue and corresponding eigenvector of a large square matrix A. Let x_0 be a non-zero starting vector, and let c_0 be the absolute value of the largest coordinate of x_0. For $n = 0, 1, 2, \ldots$, define

$$x_{n+1} = A x_n \frac{1}{c_n} \tag{3}$$

If $x_{n+1} = x_n$, then that vector is an eigenvector of A corresponding to the largest eigenvalue, c_{n+1}. Each iteration of the usual synchronous algorithm for successive substitution consists of multiplying the result of the previous iteration by the matrix A and rescaling as in (3). If A is large, it might make sense to have several processors doing different parts of the multiplication at the same time. For example, suppose A is $m \times m$ and $m = m_1 + m_2$. We might let one processor calculate $A_1 x_n$ and let another calculate $A_2 x_n$ where A_1 is the first

m_1 rows of A and A_2 is the last m_2 rows. Similarly, we might split A into $m_1, \ldots, m_p$ disjoint sets of rows and use p processors. Alternatively, we can split A into $m_1, \ldots, m_k$ disjoint (or even overlapping) sets of rows and use p processors with $p < k$. Now the question naturally arises as to whether we should wait until all k partial iterations are complete before forming x_{n+1} or should we form a new x_{n+1} every time that we learn some of the new coordinates. The first scheme produces what are known as *Jacobi* iterations, while the second produces *Gauss-Siedel* iterations. With Jacobi iterations, the x_{n+1} which results after all k partial iterations are complete is the same as what would be produced if the entire multiplication were done at once. In the case of Gauss-Siedel iterations, precisely which vector x gets multiplied by some subset of the rows of A at a particular partial iteration depends on which coordinates have been updated by the time that partial iteration begins.

As a simple illustration, suppose we split A into $k = 3$ disjoint sets of rows A_1, A_2, A_3 and we have $p = 2$ processors. (For convenience, suppose that we know that the largest eigenvalue is 1 so that we don't have to divide by the c_n values.) Let $x_0^{(i)}$ denote the coordinates of the starting vector x_0 which correspond to the rows in A_i. Suppose processor i is assigned the subtask of multiplying $A_i x_0$ for $i = 1, 2$. Now suppose processor 2 finishes its multiplication first. Let $x_1^{(2)}$ stand for the m_2 coordinates returned. For a Jacobi iteration scheme, we would now assign processor 2 the subtask of multiplying $A_3 x_0$. For Gauss-Siedel iterations, we construct x_1 by combining $x_0^{(1)}$, $x_1^{(2)}$, and $x_0^{(3)}$. Processor 2 is then assigned the subtask of multiplying $A_3 x_1$. These two subtasks will not produce the same output. For Gauss-Siedel iterations, we can assign subtasks in a simple cyclic fashion 1, 2, 3, 1, 2, 3, ... until some convergence criterion is met. Each time a new subtask is assigned, the vector x_n consists of the most recently updated values for all coordinates. When processors have widely differing speeds, one needs to be careful to keep track of how old a subtask is before updating coordinates. For example, suppose that processor 2 finishes its second subtask before processor 1 finishes its first subtask. Let $A_3 x_1 = x_2^{(3)}$. Using the cyclic assignment scheme, we would construct x_2 out of $x_0^{(1)}$, $x_1^{(2)}$, and $x_2^{(3)}$ and then assign processor 2 the subtask of multiplying $A_1 x_2$. Then if processor 2 finishes its third subtask before processor 1 finishes its first, we would have a value $x_3^{(1)} = A_1 x_2$ which would supercede the result of processor 1, namely $A_1 x_0$, when processor 1 finally completes.

There are conditions (see Baudet 1975, for example) under which Gauss-Siedel iterations of the asynchronous type described above converge. For example, let $F = (F_1, \ldots, F_n)$ be an $n-$dimensional function of n variables, and suppose that we are seeking a fixed point of F. For each j, let $x^j = (x_1^j, \ldots, x_n^j)$ be an $n-$dimensional vector representing the j^{th} iterate of an asynchronous algorithm. Let L_j represent the set of subscripts i (elements of $\{1, \ldots, n\}$) such that F_i will be calculated during the j^{th} iteration. To calculate each F_i, we need to choose an $n-$dimensional vector x as its argument. Let s_ℓ^j be the iterate from which the ℓ^{th} coordinate of x will be drawn to be used as the argument of each F_i in the j^{th} iteration. To summarize, the iterates are calculated as

$$x_i^j = \begin{cases} F_i(x_1^{s_1^j}, \ldots, x_n^{s_n^j}) & \text{if } i \in L_j \\ x_i^{j-1} & \text{if } i \neq L_j. \end{cases} \tag{4}$$

We need the following conditions:

1. $s_\ell^j \leq j - 1$ for all j and ℓ in order to guarantee that the scheme does not require future calculations to be done before past calculations.

2. $\lim_{j \to \infty} s_i^j = \infty$ for all i in order to guarantee that coordinates of the arguments to the F_i functions get chosen from newer iterations as time goes on.

3. for every i, $i \in L_j$ for infinitely many j in order to guarantee that every coordinate is updated infinitely often.

The theorem proven by Baudet (1975) is

Theorem 1 *Let $F : \mathbf{R}^n \to \mathbf{R}^n$ satisfy $|F(x) - F(y)| \leq A|x - y|$ for a $n \times n$ matrix A with spectral radius less than 1, where absolute values are to be understood coordinatewise. Then, under the three conditions described above, the asynchronous iteration scheme in (4) converges to the unique fixed point of F.*

Chazan and Miranker (1969) prove a similar theorem for linear systems.

Theorem 2 *If $F(x) = Ax + b$ with b non-zero, and the three conditions above hold, then the scheme in (4) converges if and only if the spectral radius of A is less than 1.*

For more general discussion of parallel matrix computations the interested reader should consult Gallivan et al. (1990) or Schendel (1984).

6 Statistical Applications

Several statistical applications which made use of the system of Eddy and Schervish (1986) were described by Schervish (1988). One application involved a calculation of the sum of a very large number of terms, each of which required only a small amount of computation. This is the application whose running times are displayed in Figure 1. There were 38,266,040 terms in the sum, and it took a single MicroVAX II computer 40,100 seconds to do the sum. The fifteen node

system took 4,303 seconds. This system consisted of eight MicroVAX IIs, six MicroVAX Is and one VAX 11/750. This system was estimated to have the computing power of ten MicroVAX IIs, hence reduction in running time by a factor of .107 is quite good. Another benefit of the distributed computation, compared to the serial computation, was numerical accuracy. The computation was divided into 6129 subtasks of 6237 terms each. Single precision distributed computation agreed with double precision serial computation to five decimal places, whereas single and double precision serial computation differed by as much as 5%.

Kim and Schervish (1988) analyzed survey response of 9566 inmates in order to try to model criminal careers. Due to the fact that the inmates were in jail at the time of the survey, the sample had serious recognizable bias. The likelihood function was complicated by the need to correct the bias. Also, a hierarchical model was fit, which required performing a numerical integration for each inmate. Each evaluation of the likelihood function took 57.5 minutes to compute on a VAXstation II. The application was distributed by dividing the inmates into subtasks of size 100 each. Every time a value of the likelihood function was needed, the computation was distributed. Each evaluation of the likelihood took 7.1 minutes on ten VAXstation IIs.

Not all applications benefit so dramatically from the distributed computation. Schervish and Tsay (1988) developed multiprocess models for time series which allowed for abrupt changes in level as well as outliers at every time period. As time goes on, more and more combinations of possible outliers and level changes needed to be considered. After each time period, the probabilities of the 60 combinations which seemed most likely were calculated and parameters were estimated for such combination. The 60 combinations were treated as 60 subtasks and distributed after each of 20 time periods. It took a single MicroVAX II 1391 seconds, and it took a system of six MicroVAX IIs 360 seconds. In this application, there is a significant amount of work which is not divided amongst the subtasks and this takes as much time for a distributed system as for a single processor. Amdahl's law strikes again!

7 Programming

7.1 Difficulties

There is considerable difficulty attendant to writing parallel programs. The most formidable obstacle is the lack of familiarity; programmers have been programming sequential machines for decades and various sequential programming paradigms are well-known. The need for the programmer to understand the issues related to synchronization and interprocess communication make parallel programming inherently more complex than sequential programming. There is the further difficulty that there are no standardized languages akin to Fortran, Lisp, Cobol, C, etc. for programming parallel machines.

A significant complicating factor is that unlike a program written for a sequential computer, a program written for a parallel computer cannot be easily "ported" to a different kind of parallel computer.

7.2 Linda

We have recently begun to use Linda for our parallel programming. Linda is an extension to existing languages which is based on computational model assuming a shared memory machine. The shared memory is addressed by an associative scheme. The particular model is both simple and easily implemented on a variety of real architectures and real programming languages. Consequently, programs written in Linda are portable without change across hardware environments. They are not necessarily efficient in the various environments.

The actual implementations of Linda are handled as simple extensions to existing languages such as Fortran and C. The version that we have used is C-Linda for a distributed network of processors. There are four extra functions added to the usual C language for implementing the shared memory.

1. *in*: remove data from the shared memory;
2. *out*: add data to the shared memory;
3. *rd*: copy data from the shared memory; and
4. *eval*: evaluate and add data to the shared memory.

The key to the parallel execution of the program is the function *eval*. If one of its arguments is itself another function then that function is actually executed in a separate process on a distinct processor.

To demonstrate the simplicity of C-Linda programming, below we give a "parallel" version of the classic "Hello world" program. The only features that deserve further explanation are

1. the name of the highest level routine
2. the operator "?"
3. the method by which entries in the content addressable shared memory are accessed

```
real_main()
#define NUMBER 30
{int i, hello();
  out(''number'',0);
  for(i=1;i<NUMBER;i++)eval(hello(i));
  in(''number'',NUMBER);
```

```
}
hello(i)
int i;
{int j;
  printf(''Hello world; %d.\n'',i);
  in(''number'',?j);
  out(''number'',j+1);
}
```

The name of the highest level routine must be

```
real_main().
```

The operator "?" is used for selecting any item from the shared memory using an associative scheme. For an item to match the argument of an *in* function it is necessary that it match both in type and in content if it is specified without the "?" operator. Thus the *in* in the main program is not matched by an element in the shared memory until the subroutine `hello` has been executed NUMBER times. Also the *in* in the subroutine `hello` is matched by any element in shared memory which has a variable of type *int* for its second entry.

8 References

Baudet, G.M. (1975). Asynchronous iterative methods for multiprocessors. *J. Assoc. Comput. Mach.*, **25**, 226-244.

Bertsekas, D.P. and Tsitsiklis, J.N. (1989). *Parallel and Distributed Computation - Numerical Methods.* Prentice-Hall, Englewood Cliffs.

Chazan, D. and Miranker, W. (1969). Chaotic relaxation. *Linear Algebra and Its Applications*, **2**, 199-222.

Dongarra, J.J., Duff, I.S., Sorensen, D.C., and van der Vorst, H.A. (1991). *Solving Linear Systems on Vector and Shared Memory Computers.* SIAM, Philadelphia.

Duncan, R. (1990). A survey of parallel computer architectures. *IEEE Computer*, **23**, 2, 5-16.

Eddy, W.F. (1986). Parallel architecture: A tutorial for statisticians. *Computer Science and Statistics: Proceedings of the 18th Symposium on the Interface*, 23-29.

Eddy, W.F. and Schervish, M.J. (1986). Discrete-finite inference on a network of VAXes. *Computer Science and Statistics: Proceedings of the 18th Symposium on the Interface*, 30-36.

Flynn, M.J. (1966). Very high speed computing systems. *Proceedings of the IEEE*, **14**, 1901-1909.

Gallivan, K.A., Heath, M.T., Ng, E., Ortega, J.M., Peyton, B.W., Plemmons, R.J., Romine, C.H., Sameh, A.H., Voigt, R.G. (1990). *Parallel Algorithms for Matrix Computations.* SIAM, Philadelphia.

Kim, C.E. and Schervish, M.J. (1988). Stochastic Models of Criminal Careers. in *Bayesian Statistics 3*, eds. J. Bernardo, M. DeGroot, D. Lindley, and A. Smith, Valencia, Spain: University of Valencia.

Schendel, U. (1984). *Introduction to Numerical Methods for Parallel Computers.* Ellis Horwood, Chichester.

Schervish, M.J. (1988). Applications of Parallel Computation to Statistical Inference. *J. Amer. Statist. Assoc.*, **83**, 976-983.

Schervish, M.J. and Tsay, R.S. (1988). Bayesian Modeling and Forecasting in Large Scale Time Series. In *Bayesian Analysis of Time Series and Dynamic Models*, ed. J.C. Spall, New York: Marcel Dekker.

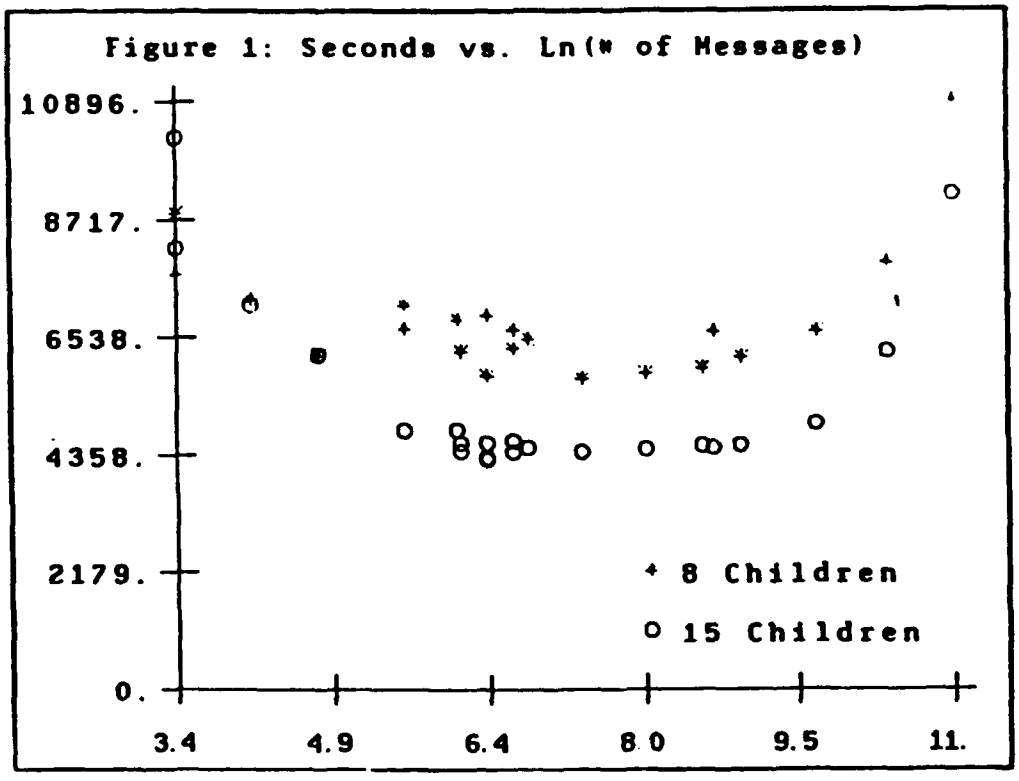


Figure 1: Empirical Study of System Performance

Some Results in the Simulation and Analysis of the Shape of Spread of Epidemics on a Grid

Michael Lloyd

Department of Actuarial Mathematics and Statistics
Heriot-Watt University
Edinburgh EH14 4AS
Scotland

Abstract

Models concerned with the spatial features of epidemic spread are often defined in terms of a nearest-neighbour grid network (Mollison & Kuulasmaa 1985). It is strongly conjectured, and can be proved in certain cases (eg Cox & Durrett 1988), that the infected area has (asymptotically) a well-defined shape.

The present work concerns computer analysis of the shape of spread of a discrete-time single-parameter infection process on an eight neighbour lattice. Data from such simulations can be fitted with a particular group of three parameters which reveal features of the shape of the expanding epidemic. These three parameters are discussed in relation to the effect of the basic model form upon them, with a view to a framework for making a priori statements about the various members of a more general model class. Such a framework would allow choices between various grid epidemic models to be made, where in the past such choices have tended to be decided arbitrarily by other (convenience) factors.

Introduction

This paper considers a class of models for stochastic diffusion. Historically, this type of model has had applications in the modelling of epidemics and forest fires; it can be conjectured that it can cover other spatial modelling situations, such as rusting or tumour growth. The class is sufficiently broad to allow many interpretations and many choices between assumptions. Currently, little is known about the effects of the specific choice of model within the class; this paper attempts to establish a quantification of the behaviour of one model which can be generalized to others of the class, facilitating comparison.

General Model Form

Following Richardson (1973), define the broad class as follows:
Starting with n-dimensional Euclidean space S, impose a (partially-ordered) cell division T. Each cell can have one of two states, referred to as White and Black. The cell which contains the origin is coloured black at time 0 and all other cells are white.

Next we impose G, a stochastic growth process, which tends to change white cells with black neighbours into black cells. We consider $C(t)$, the black shape at time t.

The most basic decisions in model selection from this class are (i) continuous versus discrete time and (ii) the neighbour structure. For example, we could have a continuous time model where each cell emits germs as a Poisson process and these germs land on neighbouring cells by some rule; alternately, we could consider discrete time where each black-white neighbour pair becomes a black-black pair at time step t with probability p. Choices of neighbour structure include 4 or 8 neighbours for cells drawn around locations with integral coordinates (the so-called Rook's Case and Queen's Case models, the former being more common in the literature). These structures are quite easy to simulate by computer, where an array readily represents the model space, but more complex tesselations can also be envisaged, such as 6-neighbour hexagonal grids.

Known Results

Some authours consider more complex model forms, especially extra states for a cell: typically removal (Cox & Durrett 1988), so that we have black, grey and white cells with white→grey transitions by neighbours and grey→black transitions purely by time. We can even consider regrowth (black→white by neighbours, Mollison & Kuulasmaa 1985) or recovery (without immunity). For some such models, asymptotic existence results have been proven; as a typical example, consider the Cox & Durrett Shape Theorem:

Recode black, grey, white to healthy, infected and immune. Each infected site emits 'germs' by a Poisson process, rate α, uniformly distributed to the 4 nearest neighbours. Infected sites survive for a length of time following some specified distribution. Define:

C_0 as the set of sites that will ever become infected from the initial infective at the origin
ζ_t as the set of immune sites at time t
ξ_t as the set of infected sites at time t

For a sufficiently well-behaved (finite second moment) infected→immune process, and for α sufficiently large,

$\exists$ **D**, a convex set, s.t. $\forall\ \varepsilon>0$,
(1) $P[\ C_0 \cap t(1-\varepsilon)\mathbf{D} \subset \zeta_t \subset t(1+\varepsilon)\mathbf{D}\ \forall$ suff large $t] = 1$
(2) $P[\ \xi_t \subset t(1+\varepsilon)\mathbf{D} - t(1-\varepsilon)\mathbf{D}\ \forall$ suff large $t] = 1$

Note that the "well-behaved" assumption is only necessary for (2). The process considered below does not have this property, but (1) still applies.

This result is typical of those in the literature, in the sense that only existence is proven, and only in an asymptotic form ($t\to\infty$). The shape of **D** is not known beyond convexity (and obvious symmetry), and the form of approach to this "equilibrium" shape **D** is unknown. Specifically, it can be concluded from Durrett & Liggett's (1981) Flat Edge Result that, against expectation, **D** cannot be circular for many cases.

The Process Under Consideration

To pursue finite-time results for such models, we turn to simulation of a specific process as follows:
Time is to be discrete; start with one "ill" individual at the origin and all others "well" (as usual) on the lattice of integer-valued coordinates in $\mathbf{R}^2$. Use an 8-neighbour (Queen's Case) connection lattice. At each timestep, every ill individual attempts to infect (separately) each well neighbour with fixed probability p of success. Iterate this procedure for T timesteps.

This process generates observations of C(T) (which must lie inside [-T,T]x[-T,T]). Re-code each element of C(T) as 1 for ill and 0 for well, and take the mean of 3000 such observations. This provides a collection of estimates

$$S(i,j;p,T) = P[(i,j) \text{ infected by time } T]$$

(note: 3000 comes from the fact that a single proportion estimate has SE $\sqrt{pq/n}$, which is maximized at p=0.5 and has a value close to 0.01 for p=0.5,n=3000, so we obtain <1% pointwise error)

Investigating the Probability Surfaces

It currently takes roughly 24 hours of processing time to generate a complete set of surfaces S(i,j;p,T) for p=0.1(0.05)0.9 and T=5(5)60 - a total of 204 simulations. This is small enough to make simulation an attractive tool for investigation of the behaviour of this type of model

Shown in Figures 1 and 2 are 3D plots of S(i,j;0.2,30) and S(i,j;0.6,30). These surfaces exhibit (surprisingly?) large areas for which $S\approx 1$, and then a rapid drop to $S\approx 0$. The shape of this area can be shown to be non-circular; its nature is most readily investigated with the aid of interactive graphical software, such as Data Desk (Velleman 1990). The usefulness of EDA software in probabilistic modelling has, in the opinion of the author, yet to be fully appreciated. This subject will be approached more comprehensively in future work.

A Functional Form for S

It is conjectured that the form of S(i,j;p,T) for this model is well approximated by a combination of a logistic curve and a measure of the distortion introduced by the grid; explicitly, the suggested form is as follows:

$$S(i,j) = \frac{1}{\exp\{b\,(r_\alpha(i,j) - a)\}+1}$$

$$\text{where } r_\alpha(x,y) = \sqrt[\alpha]{|x|^\alpha + |y|^\alpha}$$

Note that this form involves three parameters from a model which initially has only two; any redundancy is not currently clear. The parameters α, a and b are described below.

α measures the distortion from the circular: $\alpha=2$ gives a circular S surface because r_α becomes Euclidean r, $\alpha=\infty$ gives a square S, and as α increases from 2 distortion increases monotonically. This distortion is akin to the cross-section of a balloon being inflated inside a box - initially circular and eventually square.

a can be readily interpreted as the radius of the 0.5 probability contour of S in the metric generated by r_α, since when $r_\alpha = a$ we have $S(i,j) = 1/(\exp(0)+1) = 0.5$

b is a measure of the sharpness of the change from $S\approx 1$ to $S\approx 0$; as b increases S becomes more like a step function from 1 to 0 at $r_\alpha = a$.

Results from Fitting

Simulation sets have been collected as described (p=0.1(0.05)0.9 and T=5(5)60) and a fitting procedure used

to find α, a and b for each data set. Currently, the method used is to compute the sum of squared residuals between the simulation output and the fitted surface at each of the grid points, and a numerical minimization routine is applied to the surface in $\mathbf{R}^4_+$. It is by no means clear that this is the ideal criterion for picking the α, a and b of best fit.

A section of a scatterplot matrix of fitted values for α, a and b against p and T is given in figure 3. Alongside are shown evaluations of some suggested closed-form approximations for α, a and b which are proposed primarily on exploratory grounds. The forms shown are computed as follows:

$$\alpha = 2 + 1.58 \ln(T) \frac{p^3}{(1-p)^{0.5}}$$

$$\frac{a}{T} = 1 + 1.41 (p-p_c) T^{\wedge} [(1-p_c) \frac{e^{p-1}}{e-1} + p_c]$$
$$- (p<p_c) (.07(p_c-p) + 1.7(p_c-p)^2 + 3.35(p_c-p)^3)$$

$$b = e^{-.28-4.0p+15.5p^2-9.1p^3} + e^{-0.04T}$$

or more succinctly

$$\alpha = 2 + c_1 \ln(T) p^{c_2} q^{c_3}$$

$$a = T\{1 + c(p-p_c)T^{-\text{linear}(\exp(p))} - (p<p_c)\text{cubic}(p)\}$$

$$b = e^{\text{cubic}(p)} + e^{-cT}$$

from which we observe the following:

α behaves like 1/q so that we get the correct limit of ∞ as $p \to 1$; it is not clear what the behaviour should be as $p \to 0$ (it is unclear whether a single point - the origin - is circular or square!), and it is hard to reliably compute α for very small p as a very large number of simulations are required. Note that there is a very small upturn in the simulation output values at very small p which is not reflected in the suggested approximation. Also, α appears to behave as log(T), which confounds an earlier hypothesis in Lloyd (1991) that alpha would be invariant with T.

a has a quite complex form, although essentially each term is just a deviation from the first, which indicates that $a \approx T$. Certainly for p and T both sufficiently large, $a \approx T$ is a very good approximation indeed ("large" here means roughly $p \gg 0.5$ and $T \gg 30$). As T decreases, there is an additional term of the form 1/T - the complicated exponent for T is just a form that runs from p_c to 1 as exp(p). Finally, for p below a certain value, a increases linearly with T, but with a slope less than 1. The final term accounts for this: p_c is that value of p for which (it is hypothesized that) a comes out as exactly T for all T. Note that it may prove interesting to relate this critical probability (currently estimated at roughly 0.51) to the many other critical probabilities to be found in the literature for this type of problem. Above p_c, a must asymptotically converge to T, but below it the limit is f(p)T. The cubic form is a crude expression of f(p), and it is very much hoped that a form with more meaning can soon be found.

b splits quite simply into independant functions of p and T, both exponential in form. Again, a cubic for p fits well but can hopefully be replaced with a more justifiable function with further research.

Finally it is stressed once again that these closed forms are exploratory in nature. Once similar functions are available for other model assumptions (other neighbour and time structures) they will facilitate direct model result comparison. Specifically, it is hoped that the α parameter will provide a basis for discussion of the distortion introduced into the model by the assumption of a grid; the model form with lowest values for α would be the most desirable as it would have the most isotropic behaviour.

References

JT Cox and R Durrett (1988), Limit Theorems for the Spread of Epidemics and Forest Fires, Stoch Proc Appl, 30:171-191

R Durrett and Liggett (1981), The Shape of the Limit Set in Richardson's Growth Model, Ann Probab, 9(2):186-193

M Lloyd (1991), Computer Analysis of the Shape of Spread of Epidemics on a Grid, *to appear in* Math Biosci

D Mollison and K Kuulasmaa (1985), Spatial Epidemic Models: Theory and Simulations, *in Population Dynamics of Rabies in Wildlife*, PJ Bacon ed, Academic Press, London, pp 291-309

Richardson D (1973), Random Growth in a Tessellation, Proc Camb Phil Soc, 74:515-528

P Velleman (1989), Data Desk, Odesta Corp, Illinois

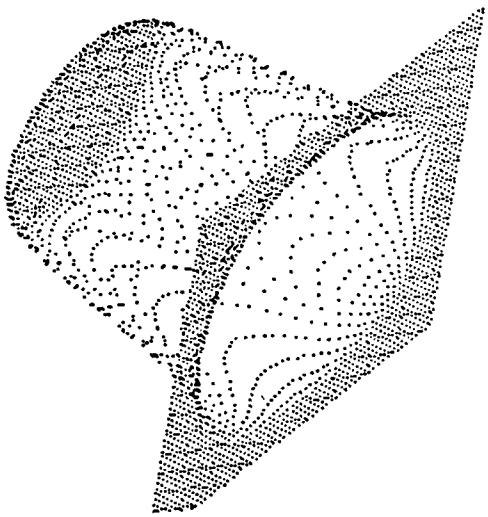


Figure 1: S(i,j;0.2,30)

Figure 2: S(i,j;0.6,30)

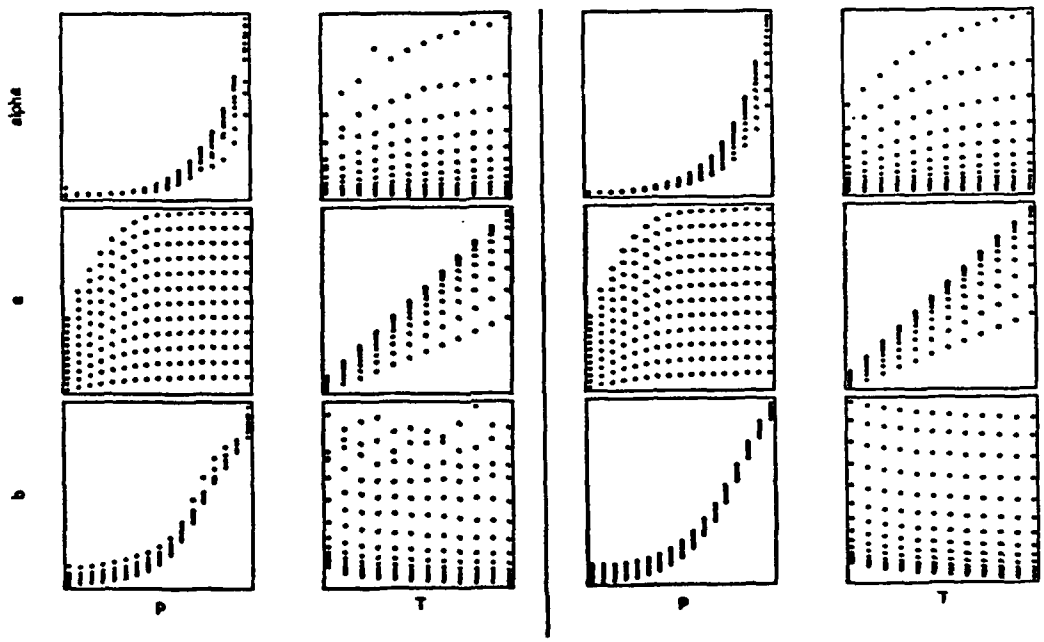


Figure 3: Observed (L) and Fitted (R) α, a and b

Spatial Patterns of Trees Attacked by Beetles: Pseudolikelihood Estimation and Iterative Simulations

Haiganoush K. Preisler
USDA Forest Service
Pacific Southwest Experiment Station
1960 Addison St., Berkeley, Cal. 94704

1 INTRODUCTION

In this preliminary study a generalized linear model is used to describe the conditional probability of a tree being attacked by mountain pine beetles in a given year, given the characteristics of the tree (e.g., size) and the location of other attacked trees in the stand. The model is used to analyze mountain pine beetle attack data in two lodgepole pine stands in Oregon over a period of 10 years (see Fig. 1 and 2). The data may be viewed as a realization of a spatial point process with the probability of a tree being attacked dependent on the status of other trees in the stand. Although full maximum likelihood estimation is apparently not feasible, maximum pseudolikelihood estimates of the parameters can be readily calculated with standard statistical packages such as GLIM. The pseudolikelihood function that is maximized is the product, over all trees, of the conditional probabilities. This method of estimation was first proposed by Besag (1975). See also Strauss and Ikeda (1990).

2 AN AUTO-LOGISTIC MODEL

Let Y_{ik} equal 1 if tree i was attacked in year k and 0 otherwise, where $i = 1, \cdots, n_k$; $k = 1, \cdots, K$ and n_k= number of trees that have not been attacked in any of the previous years $1, \cdots, k-1$. The probability, p_{ik}, of tree i being attacked in year k conditional on the status of all other trees in the stand, will be modeled by

$$\begin{aligned} p_{ik} &= \Pr[\, Y_{ik} = 1 \mid y_{jk} \,;\, j \neq i \,] \\ &= \Pr[\, Y_{ik} = 1 \mid \theta, dbh_i, vig_i, D_{ik}] \\ &= \frac{e^{\eta_{ik}}}{1 + e^{\eta_{ik}}}, \end{aligned} \tag{2.1}$$

with

$$\eta_{ik} = \alpha_k + \beta_1 D_{ik} + \beta_2 \log(dbh_i) + \beta_3 vig_i \tag{2.2}$$

$$D_{ik} = \sum_{j \neq i} d_{ij}^{-2} y_{jk} \tag{2.3}$$

Fig. 1. Stand A after 1980 attack. + =trees attacked that year. o = trees with dbh>23cm.

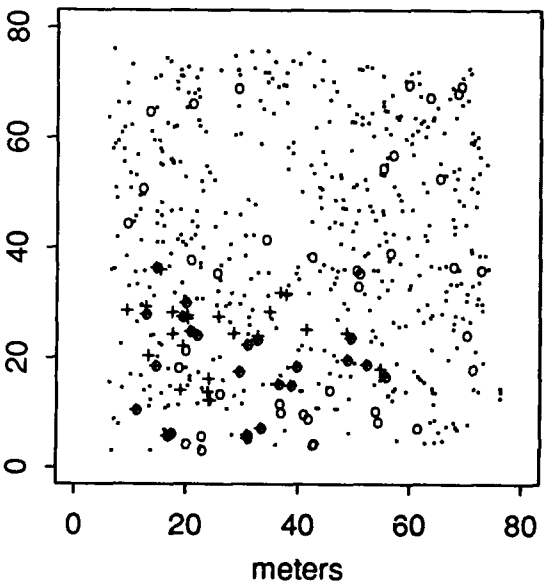
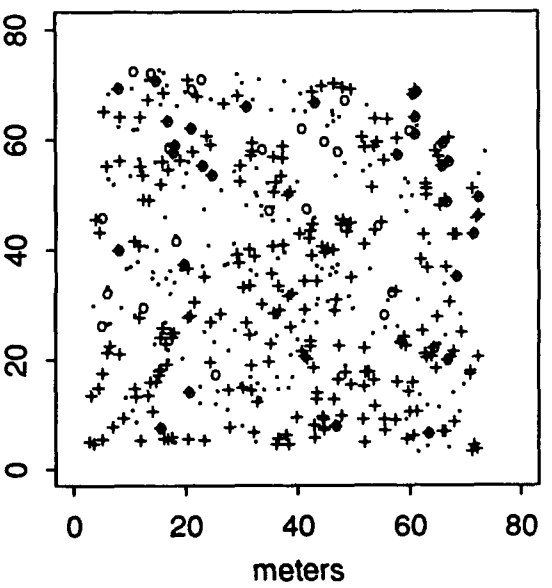


Fig. 2. Stand B after 1984 attack. + =trees attacked that year. o = trees with dbh>23cm.



where dbh_i=diameter at breast height, vig_i=vigor of tree as measured by the amount of stemwood produced per square meter of crown leaf area per year (Waring *et al.* 1980), d_{ij} = distance between trees i and j and $\theta = \{\alpha_k, \beta_1, \beta_2, \beta_3; k = 1, \cdots, K\}$ is a set of unknown parameters. The variate D_{ik} could be viewed as a measure of the density of attacked trees surrounding the i^{th} tree. D_{ik} is large if tree i is near other attacked trees.

The conditional probability model in (2.1)-(2.3) is an auto-logistic model (Besag 1974) with the logit line

$$\eta(\mathbf{y}) = \alpha_i + \sum_{j \neq i} \beta_{ij} y_j$$

where $\beta_{ij} = \beta_{ji} = \beta_1 d_{ij}^{-2}$ and $\alpha_i = \alpha + \beta_2 log(dbh_i) + \beta_3 vig_i$. Auto-logistic models satisfy the constraints in the Hammersley-Clifford theorem (Besag 1974) which guarantees that the conditional probabilities defined above are consistent with a joint probability distribution.

3 POINT ESTIMATION

Given the above generalized linear model, a maximum pseudolikelihood estimator (MPE) for the unknown parameter vector θ will be defined as the vector $\hat{\theta}$ which maximizes the pseudolikelihood function

$$\prod_{k=1}^{K} \prod_{i=1}^{n_k} Pr[y_{ik} \mid y_{jk} \; ; \; j \neq i]$$

$$= \prod_{k=1}^{K} \prod_{i=1}^{n_k} p_{ik}^{y_{ik}} (1 - p_{ik})^{y_{ik}} \qquad (3.1)$$

Pseudolikelihood methods were first proposed by Besag (1975) for estimation of parameters in a general Markov random field context. Strauss and Ikeda (1990) showed that for a logit model similar to (2.1), maximization of (3.1) is equivalent to a maximum likelihood fit of a logit regression model of the form in (2.1) with independent observations y_{ik}. Consequently, estimates can be obtained using an iteratively reweighted least squares procedure. Any standard logistic regression routine can therefore be used to obtain MPE's of the parameters. However, the standard errors of the estimated parameters calculated by the standard programs are not directly applicable because they are based on the assumption of independence of the observations.

4 STANDARD ERROR ESTIMATION

In this section standard errors of MPE's are estimated using a parametric bootstrap procedure (Efron 1982, 1990). An iterative sampling scheme is used to simulate samples from the joint distribution $\tilde{\pi} = Pr\left[Y_1 = y_1, \cdots, Y_n = y_n | \hat{\theta}\right]$ given the conditional probabilities $p_i = Pr\left[Y_i = y_i | (y_j;\ j \neq i),\ \hat{\theta}\right]$ for $i = 1, \cdots, n$. The sampling scheme is as follows: Starting with an arbitrary set of initial values $(y_1^{(0)}, \cdots, y_n^{(0)})$, generate a new value $y_1^{(1)}$ from the distribution of $Y_1 | y_2^{(0)}, y_3^{(0)}, \cdots, y_n^{(0)}$, next, generate $y_2^{(1)}$ from the distribution of $Y_2 | y_1^{(1)}, y_3^{(0)}, \cdots, y_n^{(0)}$ and so on, up to $y_n^{(1)}$ from the distribution of $Y_n | y_1^{(1)}, y_2^{(1)}, \cdots, y_{n-1}^{(1)}$. This is a Markov chain sampling scheme with transition probabilities given by the conditional probabilities p_i. In this scheme only one variable is changed in each transition and after n transitions we arrive at the sample $(y_1^{(n)}, \cdots, y_n^{(n)})$. Hastings (1970) showed that if the matrix $\mathbf{P}$ of transition probabilities is reversible and irreducible then $\tilde{\pi}$ is the unique stationary distribution of the Markov process $\mathbf{P}$. For the present data, the only states with positive transition probabilities are those of the form $s_0 = \{Y_i = 0,\ \mathbf{Y}^{-i} \in s_l^{-i}\}$, and $s_1 = \{Y_i = 1,\ \mathbf{Y}^{-i} \in s_l^{-i}\}$, where $\mathbf{Y}^{-i} = \{Y_j\ ;\ j \neq i\}$ and s_l^{-i} is the l^{th} state space (or possible outcome) of the vector $\mathbf{Y}^{-i}$. Therefore,

$$\begin{aligned}
&\tilde{\pi}_{s_0} p_{s_0, s_1} \\
&\quad = Pr[Y_i = 0, \mathbf{Y}^{-i} \in s_l^{-i}] Pr[Y_i = 1 | \mathbf{Y}^{-i} \in s_l^{-i}] \\
&\quad = Pr[Y_i = 0 | \mathbf{Y}^{-i} \in s_l^{-i}] Pr[\mathbf{Y}^{-i} \in s_l^{-i}] \\
&\qquad \times Pr[Y_i = 1 | \mathbf{Y}^{-i} \in s_l^{-i}] \\
&\quad = p_{s_1, s_0} \tilde{\pi}_{s_1}
\end{aligned}$$

In other words, the Markov chain is reversible. The chain is also irreducible because in any given stand the distance between any two trees is finite and, therefore all the conditional probabilities are nonzero.

Geman and Geman (1984) called this sampling scheme the 'Gibbs sampler' and developed some general results about the convergence and rate of convergence of the joint density of $(Y_1^{(n)}, \cdots, Y_n^{(n)})$ to the true joint density of $(Y_1, \cdots, Y_n)$. For the present problem, t iterations of the above sampling scheme replicated M times will produce M independently identically distributed samples

$(y_{1m}^t, \cdots, y_{nm}^t)$, $(m = 1, \cdots, M)$, from the distribution π_t that has $\hat{\pi}$ as its stationary distribution.

Figures 3-6 are plots of the MPE's of the parameters calculated after each iteration using the spatial locations of trees in stand **A** (see Fig 1). Each iteration involved the generation of $n = 576$ random variates (where n= number of trees in stand **A**). Initial values were generated assuming spatial independence (i.e., assuming a logit model with $\beta_1 = 0$). Results of the simulations seem to indicate that the MPE's are unbiased. The values seem to oscilate around the actual parameter values used to generate the data. Also, the rate of convergence was very fast. The rate of convergence of the sampling scheme did not appear to depend on the initial values. For example, use of $p_i \equiv p$, $(i = 1, \cdots, n)$, to generate initial values gave the same results (i.e., convergence within a few iterations) as the more informative initial values used above.

5 RESULTS

Data from the first three years in stand **A** and the fourth year in stand **B** were used to calculate two sets of estimates (one for each stand) of the parameters in (2.2). Data from the remaining years were not included in the analysis because the numbers of attacked trees were either zero or small (< 10). Table 1 lists the values of the MPE's and two estimates of their standard errors. MPE's were calculated using the GLIM statistical package with the logit link and binomial error options. The standard errors from simulations are the standard deviations of $\hat{\theta}^{\mathbf{m}}$, $(m = 1, \cdots, 200)$, from 200 simulations using the sampling scheme described above. For each simulation, the variates generated after 20 iterations were used to fit the logit model in (2.1)-(2.3).

In both stands **A** and **B** the covariates *dbh* and D seem to have significant effects on the conditional probabilities. Figures 7-8 are contour plots of the estimated conditional probabilities versus $A = 1/\sqrt{D}$. For a given tree, its distance measure, A, is small if the tree is close to other attacked trees. The contour plots seem to indicate that the probability of a small tree (dbh less than 15 cm) being attacked is small unless it is close to other attacked trees. However, large trees seem to be attacked even when their distance measure, A, is large, i.e., even when there are no attacked trees nearby.

Further studies that are in progress include assessing the goodness-of-fit of the auto-logistic model and the use of alternative measures of distance with, perhaps, more biologically meaningful interpretations.

REFERENCES

Besag, J. (1974), "Spatial interaction and the statistical analysis of lattice systems," (with discussion), *Journal of the Royal Statistical Society*, Ser. B, 36, 192-326

—— (1975), "Statistical analysis of non-lattice data" *Statistician*, 24, 179-195.

Efron, B. (1982), "The jackknife, the bootstrap, and other resampling plans," CBMS Monograph #38, Society for Industrial and Applied Mathematics, Philadelphia.

—— (1990), "More efficient bootstrap computations," *Journal of the American Statistical Association*, 85, 79-89.

Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721-741.

Hastings, W. K. (1970), "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 57, 97-109.

Payne, C. D. (ed). 1986. The GLIM system release 3.77 Manual. Numerical Algorithms Group, Oxford, U.K. 305p.

Strauss, D. and Ikeda, M. (1990), "Pseudolikelihood estimation for social networks," *Journal of the Statistical Association*, 85, 204-212.

Waring, R. H., W. G. Theis, and D. Muscato. 1980. Stem growth per unit leaf area: a measure of tree vigor. *Forest Science* 26, 112-117.

Table 1. Pseudolikelihood estimates of parameters and standard errors produced by fitting the auto-logistic model to stand A 1981 and stand B 1984 data.

Parameter	Estimate	Standard Error from Simulations	Standard Error from GLIM
Stand A			
	-20.30	2.252	2.214
	17.66	2.018	2.264
	6.24	0.787	0.753
	-0.003	0.007	0.009
Stand B			
	-12.60	1.313	1.410
	3.06	1.124	1.008
	4.54	0.443	0.484
	-.0013	0.008	0.007

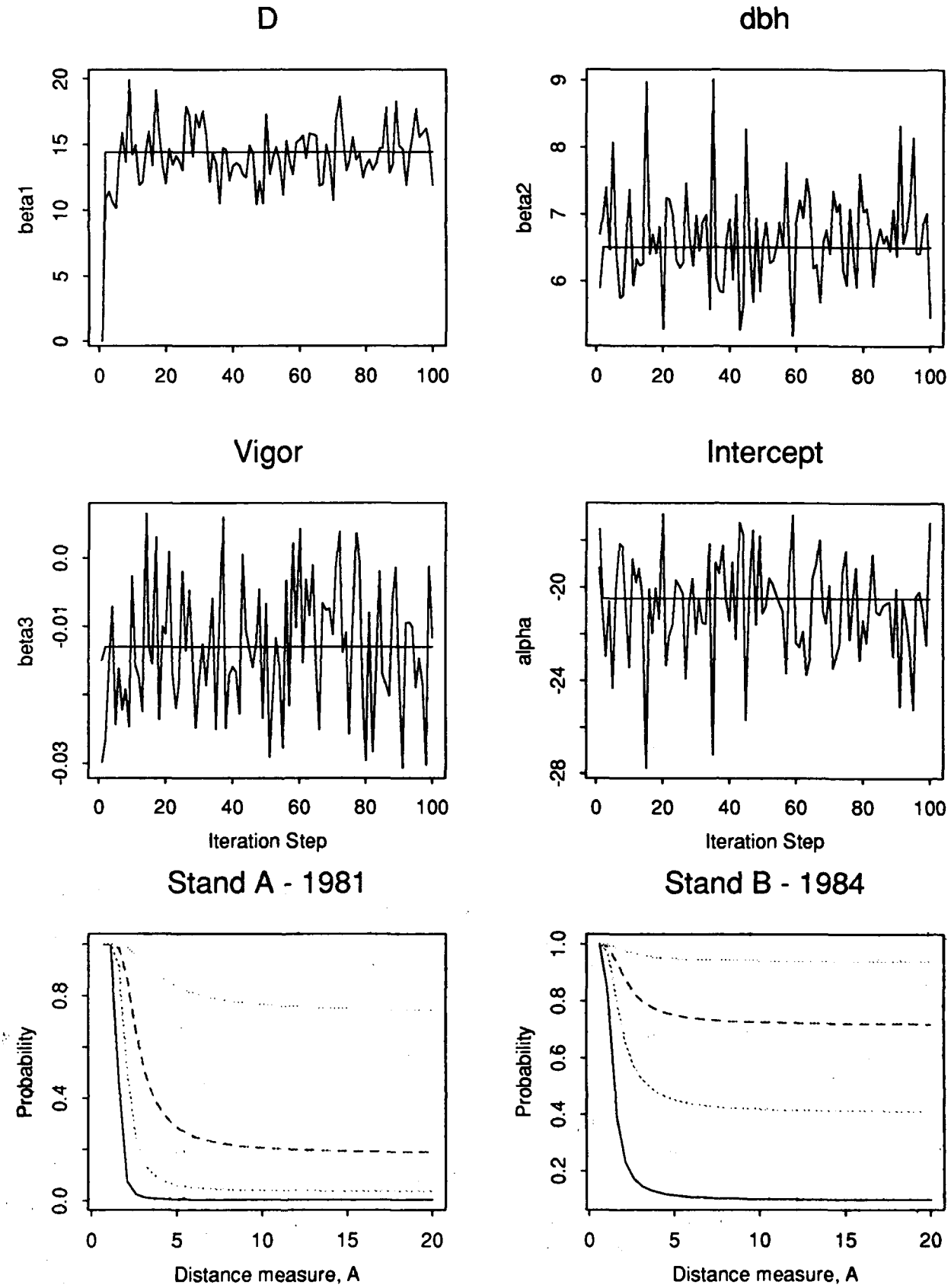


Fig. 3-6. Parameter estimates from simulations.

Fig. 7-8. Estimated conditional probabilities of attack.

Statistical Analysis of Anthropometric Data

David G. Robinson, Michael Grant*
Air Force Institute of Technology
Department of Aeronautics and Astronautics
Wright-Patterson AFB, OH 45433
drobinso@blackbird.afit.af.mil

Abstract

A major source of problems in poor fitting environmental protection equipment is the lack of proper consideration of the possible variation among aircrew facial feature sets. The design of flight equipment is currently based on rather outdated mechanical anthropometric measuring techniques with limited justification for specific size and shape characteristics. This paper discusses the preliminary attempt to statistically capture this variability in an organized manner. Three-dimensional data collected from a laser scan of 200 subjects were statistically summarized for presentation and analysis. However, the techniques applied to the problem go beyond the classical Fourier or trend surface methods. Statistical methods traditionally reserved for geology were applied allowing full consideration of the correlation structure of the facial area. An "average" face along with upper and lower percentiles were then available as input to computer-aided design programs.

1 Introduction

The objective of the study was to develop techniques for statistically analyzing anthropometric data so that physical models could be developed to support the design of flight equipment such as oxygen masks and limited visibility goggles.

The statistical methods used in this study are founded in the spatial analysis techniques associated with kriging. Because relatively very little has been published on the applications of kriging outside the geostatistical co.nmunity, a brief introduction to this technique is provided.

*This study was sponsored by the Human Engineering Division of the Harry G. Armstrong Aerospace Medical Research Laboratory, WPAFB,OH.

2 Kriging

Typically, the kriging process consists of two phases: structural analysis to determine the spatial distribution of the variables, and estimation using a best linear unbiased estimator. In the first phase, the variogram is used to quantify the structural information and is defined as:

$$\gamma(h) = 2\mathrm{Var}[F(x-h)) - F(x)]$$

where $F(x)$ describes a random function over the support $x, h \in R^2$. In practice, a model is fit to the experimental variogram using weighted least squares or graphical techniques. One commonly used theoretical variogram is the spherical model which is defined as follows:

$$\gamma(h) = \begin{cases} C[\frac{3h}{2a} - \frac{h^3}{2a^3}] + C_0 & h \le a \\ C + C_0 & h > a \\ 0 & h = 0 \end{cases}$$

For estimation, we desire an unbiased, linear estimate $\hat{F}(x)$ that has minimum expected estimation error. The estimate of $F(x) = \hat{F}(x)$ is assumed to be a linear estimator involving N observations in the neighborhood of $F(x)$:

$$\hat{F}(x) = \sum_{i=1}^{N} \lambda_i F_i(x)$$

where the weights λ_i's are chosen so that the estimate is unbiased: $E[\hat{F}(x) - F(x)] = 0$ (i.e. $\sum_{i=1}^{N} \lambda_i = 1$), and estimation error variance

$$\sigma_e^2 = \mathrm{Var}[\mathrm{F(x)} - \hat{\mathrm{F}}(\mathrm{x})]$$

is minimized. In terms of the variogram the estimation variance is given by:

$$\sigma_e^2 = -\sum_i \sum_j \lambda_i \lambda_j \gamma(h_{ij}) + 2\sum_j \lambda_j \gamma(h_{jF})$$

Minimizing this variance subject to the unbiased constraint results in the following set of linear equations:

$$\sum_j \lambda_j \gamma(h_{ij}) + \mu = \gamma(h_{iF})$$

$$\sum_j \lambda_j = 1$$

where $i = 1, \ldots, n$, μ is a Lagrange multiplier, h_{ij} is the vector distance between observations $F_i(x)$ and $F_j(x)$, and h_{iF} is the vector distance between $F_i(x)$ and the point to be estimated $F(x)$. This form of the kriging equation is generally refered to as punctual kriging.

The solution to these equations, λ_i^* and μ, are then used to make point estimates of the surface at x:

$$\hat{F}(x) = \sum_j \lambda_j^* F_j(x)$$

along with estimates of the prediction error:

$$\sigma_e^2 = \mu + \sum_j \lambda_j^* \gamma(h_{jF})$$

For a more detailed discussion on kriging, the reader is refered to the geostatistical literature (e.g. Cressie (1989), David (1977) or Journel and Huijbregts (1989)).

3 Problem Solution

The end product of this study was a prototype for eye-protection gear. The following discussion outlines the problem solution and describes data collection and analysis, structural analysis, and spatial estimation.

3.1 Data Collection and Analysis

Personnel from the Armstrong Aerospace Medical Research Lab collected the data using a Cyberware Echo digitizer. The laser scanner is mounted on an arm which rotates around the head of the seated subject, providing measurements of 131072 points over the entire surface of the head (512 locations on the plane of rotation, and 256 locations on the vertical plane). The corresponding third coordinates are determined by measuring the depth at these points using a triangulation procedure based on the scanner's reference point.

Prior to analysis of the spatial properties it was neccessary to orient the subjects relative to a common axis system. The data for the region surrounding the eyes was established by truncating the data sets to the points within the glabella, pronasale, and left and right tragions. The subjects were aligned using a multivariate optimization routine for minimizing the squared euclidian distance between these four landmarks and four predetermined external reference points.

A total of 200 subjects were available for analysis and all associated data sets were aligned to a common axis system. From this population of 200, a random sample of 35 subjects were chosen for structural analysis.

3.2 Structural Analysis

An artificial grid of 50 by 100 was established and superimposed upon each of the 35 data sets. Initial estimates of the global trend were determined using a simple nearest neighbor method for a selected grid structure. After removal of this initial trend from a subject, the residuals were analyzed to determine the nature and extent of the correlation structure. A spherical variogram with parameters $C = 2.226$, $C_0 = 0.689$, and $a = 6.645$ was found to best describe the spatial correlation for the region of the face of interest in this study. Figure 1 displays the theoretical variogram overlayed on the experimental variograms (for four directions) for a typical subject.

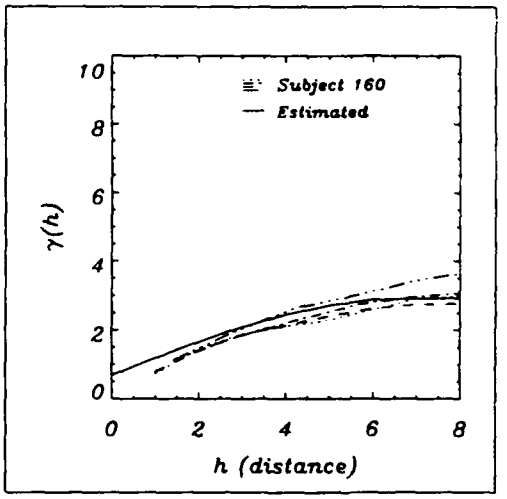


Figure 1. Variograms for Subject 160

3.3 Individual Spatial Estimation

An artificial grid of 50×100 points was superimposed on the residual data sets and the kriging equations established for each of the 5000 points. The resulting estimate when added to the global trend provides an estimate of the function describing the subject face as well as an estimate of the variance of the surface estimate at any point. Figure 2 depicts the results of kriging one subject.

It was assumed that the variogram and the global trend surface based on sampled 35 subjects, represented

the population and would not change. Using the original variogram and trend surface, an estimate for each subject was determined individually and then aggregated sequentially using a recursive relationship to update the population mean and variance.

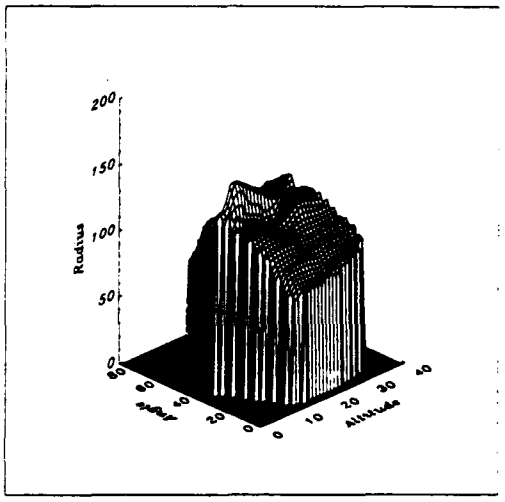


Figure 2. Kriged Surface for Subject 9

3.4 Population Spatial Estimation

If the subjects are assumed to be uncorrelated, an estimate of the population mean and variance at a particular grid location for the k^{th} subject can be estimated by:

$$\mu_k = p_k b_k$$

with:

$$p_k = \frac{1}{\sum_{i=1}^{k} w_i}$$

$$b_k = \sum_{i=1}^{k} \hat{F}_i$$

$$w_i = \frac{1}{1+\sigma_i}$$

where $\hat{F}_i$ and σ_i are the kriging mean and variance, respectively, of the i^{th} subject at the grid location of interest. After a bit of algebra and by slightly modifying the algorithm, the following relationships result:

$$\mu_k = \mu_{k-1} + [\hat{F}_k - \mu_{k-1}]$$

$$\sigma_k^2 = \sigma_{k-1}^2 - p_k[\sigma_{k-1}^2 - (\hat{F}_k - \mu_{k-1})^2]$$

4 Results

This scheme was applied to all 200 subjects. Figure 3 represents the final surface estimate for the limited visibility goggles developed with this procedure.

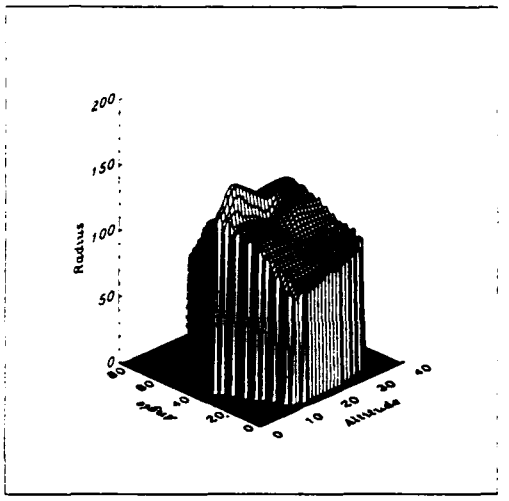


Figure 3. Night Vision Goggles Surface Estimate

Following the estimate of the upper facial surface, the data was reformatted for input to a numerically controlled milling machine and a physical model constructed. This model has since been used to develop and evaluate alternative night-vision goggle support structures.

Engineers now have the statistical methods to support the design of flight apparatus which accounts for the shape of facial features. In addition, research is almost complete on extending the methods to include not only analysis of surface features (2-dimensional data), but also analysis of 3-dimensional data from magnetic resonance image data.

REFERENCES

Cressie, Noel (1989). Geostatistics. *The American Statistician*.**43**,4,197-202.

David, Michael (1977). Geostatistical Ore Reserve Estimation. *Elsevier Scientific*.

Journel, A.G., Ch. J. Huijbregts (1989). Mining Geostatistics. *Academic Press*.

Small, C.G. (1988). Techniques of Shape Analysis on Sets of Points. *International Statistics Review*. **56**,3,243-257.

Hierarchical Modeling: An Aid to Modeling Complex Systems

Claude Ginsburg

Boeing Computer Services
PO Box 24346
Seattle WA 98124

Abstract

Most engineering application programs are designed to model, analyze, design, or monitor complex systems. Such systems can often be represented by schematic diagrams. Hierarchical modeling is a method by which complicated schematic diagrams can be expressed in a more easily comprehended form. This paper describes a software "layer" for editing hierarchical schematic diagrams that can be used as a generic interface into many application programs.

Introduction

Schematic diagrams are a graphical method for representing complex systems. These diagrams consist of icons representing system *elements* and *connectors* representing a logical association of the elements. Examples of the widespread use of schematic diagrams in modeling are in control system analysis, simulation, flowcharting, and communications (Hammond et al.[1989]), Ozden[1991], Sargent[1986], Stanwood et al.[1986]). A schematic diagram is a particularly powerful tool as a user interface to any set of programs that analyze, model, monitor, or control these systems.

One of the key advantages of schematic diagram based interfaces is that many tasks of model validation can be performed within the graphical interface. The user simply does not make many mistakes (such as mis-connections) that are easily missed in other formats. Some application-specific error checking can proceed and trap errors before analysis commences.

This paper describes one such interface created initially for EASY5, a controls analysis and non-linear simulation program written by Boeing Computer Services. An important aspect of the interface is that it was written to be independent of and isolated from the underlying program or application. It was therefore possible to include features generally desirable in many applications of schematic diagrams; the interface is currently being used in at least two quite different areas.

Hierarchical Structure

A natural simplification of the schematic diagram results when collections of elements that perform related functions or operations are grouped into a single new *meta-element*. The meta-element contains the sub-graph of related elements, but appears as a single icon. These meta-elements then represent another "level" in the schematic structure, and convert the two-dimensional graph to a tree-structured acyclic graph; acylic in the sense that data flow between the nodes or levels is directional. Each level contains a sub-graph. Connectors still actually terminate at elements, but appear to terminate at the meta-element that represents the next level down in the tree. If the contents of the meta-element are displayed, the isolated piece of the schematic is visible; connectors to elements in the "parent" schematic appear to run to the edge of the window. The entire schematic becomes a *hierarchy* of meta-elements.

Meta-elements can provide a dramatic simplification of complex schematics. Fig. 1 shows an EASY5 schematic model of a complex aircraft control system before(a) and after(b) grouping elements into meta-elements, and the appearance of the sub-graph within one meta-element formed(c). The schematic structure can become quite complex without the user losing an intuitive grasp of the system.

The above description specifies a graphical process to form a meta-element. No changes to the represented system structure are involved. Meta-elements may also be created in which the functionality of the individual elements is completely isolated from the rest of the schematic. Access to the

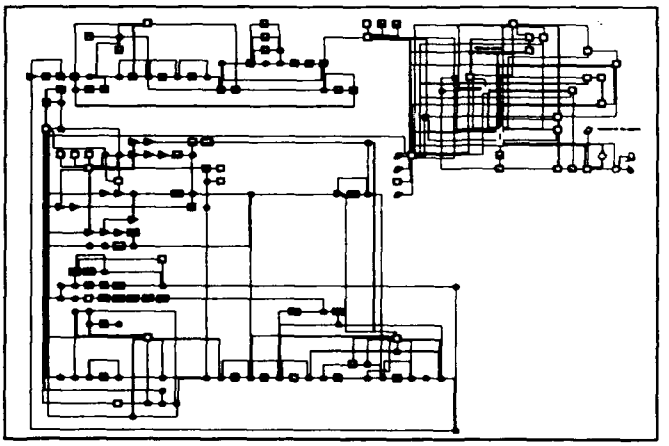


Fig. 1(a). Schematic diagram of complex control system, before forming meta-elements.

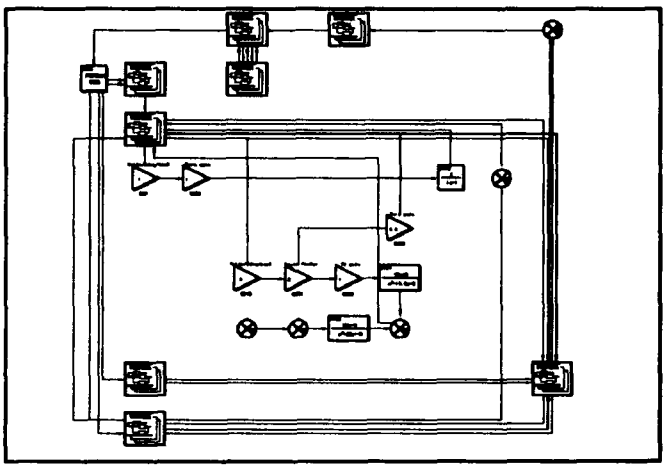


Fig. 1(b). The same schematic diagram but simplified with the formation of meta-elements.

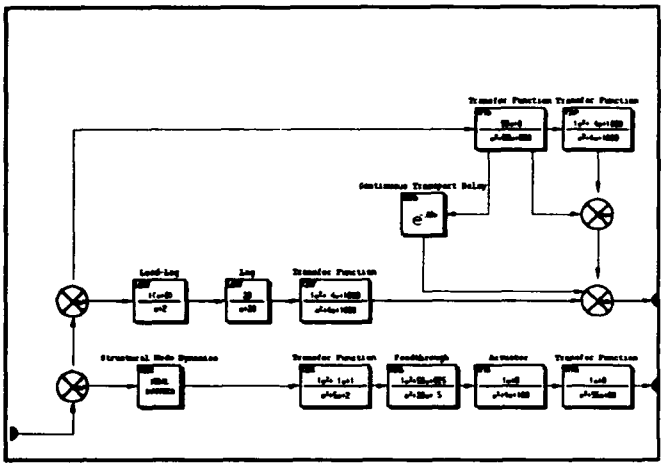


Fig. 1(c). The result of an examine operation on one of the meta-elements formed. Note the off-page connectors.

variables and parameters of the meta-element is provided through a specified subset of all possible connectable inputs and outputs. In this way a grouping of elements becomes an entity similar to a true element, and is referred to as a parameterized meta-element.

Both types of meta-elements have a certain utility in simplification. We chose to implement graphical meta-elements only, as they are easier to understand, create, and manipulate.

Meta-elements are an aid to configuration control, as they can be stored in libraries and included as "plug-in" or reusable subsystems in any system. Testing can proceed on new enhancements to a subsystem while other users of the same meta-element can rely on the older versions; when the testing is complete, all schematics can access the improved version.

Programming Concepts

Schematic diagrams, by their nature, lend themselves to an object-oriented treatment. Each object in the diagram contains common attributes such as appearance-related information (size, color, position), current state ("selected" or other application-related states), and a unique identifier. Application-defined data is stored in "pigeonholes", or pointers to data that the schematic program passes without processing. It is this latter aspect that allows the separation of application end use of the schematic and the schematic interface itself.

Data contained within the pigeonholes can be observed with data viewers. These pieces of the interface are application specific and allow a user to "examine" elements of the schematic diagram for content and change that content, as well as observing and editing aspects of the data flow within the system. For instance, EASY5 elements have defined signal inputs and outputs. If the inputs are not connected or filled by variables from other elements, they become static parameters of the system. Fig. 2 shows a data viewer for an element that contains editable FORTRAN code and definable inputs and outputs.

A schematic diagram is a representation of a system. The topology of the system refers to the mathematical structure represented by the graph itself, and not to the appearance of the schematic diagram. This implies that operations that simply affect the appearance of the schematic, such as moving elements, grouping elements into meta-elements, etc. do not require application-specific interaction.

Interface functions are called by the schematic program whenever modifications are being made to the system topology. These operations include adding or deleting elements or connectors. This allows the specific application to approve or reject the change or inquire for more information.

Information not directly related to or contained within specific elements or connectors may be added to the schematic as *drawings*, i.e., collections of graphics and text information not required by the system but helpful for documentation.

Elements and drawings store their appearance as vector or other scalable graphic information, editable by the user. Bitmap icons, while sometimes superior in appearance, are difficult to scale.

General Appearance of the Interface

The interface appears as a window surrounded by some control panels and menu bars. In the large central window, the user may add elements (icons) from a menu of available functions and position them in the window. The appearance of individual elements may be changed with a graphics editor,

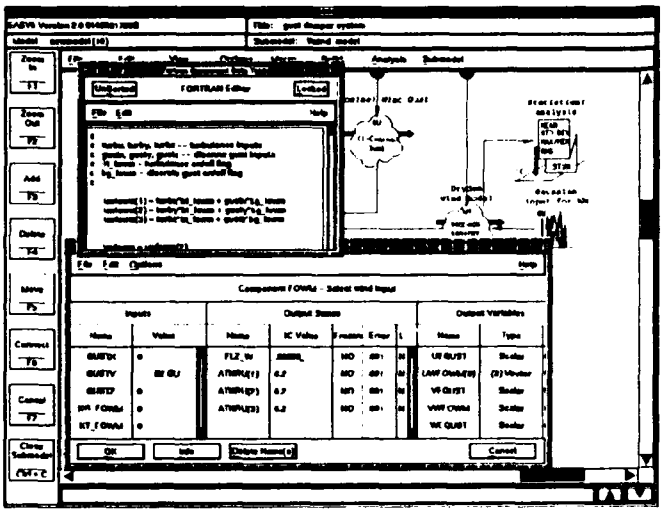


Fig. 2. EASY5 data viewer during an examine operation on a FORTRAN element.

and text or other symbols or drawings may be added to the schematic. By *selecting* (moving a cursor over and clicking a mouse button) an element, it is identified for subsequent operations such as moving its location, deletion, copying, or attaching a connector to another element. Clicking another button on the mouse performs an *examine* operation which gives the user some feedback as to the contents or significance of the examined object (Generally, a data viewer is invoked.). The window may be "panned" over the schematic to view different portions of it, and the magnification of the view may be changed. Application-specific controls in the menus perform analysis tasks. Schematics may be saved and read in, with previous versions automatically numbered. Schematic diagrams generated by completely different applications can still be viewed and graphically edited in any program using the interface.

Specific Requirements

Through much experience with engineering use of schematic diagram interfaces, certain requirements for their construction have been established. These include features that make the interface more intuitive and easy to use as well as performing some of the laborious engineering tasks of modeling:

(1) The schematic must allow scaling operations such as zoom and pan so that the diagram may be viewed in its entirety or enlarged to study a specific portion.

(2) Connectors must initially be routed automatically. This removes the onerous task of cleaning up connectors whenever the position of elements is changed in the schematic. There are many rule-based algorithms for connector routing available (see Heller et al.[1982] or Lee[1961]). Automatic routing here implies rectilinear routing that avoids obstacles such as elements.

(3) Creation and destruction (the inverse operation) of meta-elements should be very transparent to the user.

(4) Navigation or view control between levels of the schematic (i.e., between meta-elements) should be quite facile. By executing an examine operation on any meta-element, the user should then view the level within that meta-element. Navigation is facilitated by a dynamic list of available meta-elements, and labeling that indicates the current level being viewed.

(5) Hardcopy must be presentable as a final document and contain as much information as possible without being cluttered.

Conclusions

A schematic interface is an excellent enhancement to many existing computer programs that are quite mature in their computational ability but require detailed and complex input from the user. It is also a useful paradigm for building an analysis program from scratch. The interface we built is being used for both cases. In the case of EASY5, more than 6 years of experience with a commercial product that has included an increasingly sophisticated schematic front end has demonstrated vastly increased productivity for the engineers that use it.

Acknowledgments

I am very grateful to Brian Ummel and Ron Hammond for their helpful suggestions and comments.

References

Hammond, R., Newman, D., and Yeh, Y. (1989). "On Fly-by-wire Control System and Statistical Analysis of System Performance". *Simulation*, **53**, 159-167.

Heller, W.R., Sorkin, G., and Maling, K. (1982). "The Planar Package Planner for System Designers". *Proc. of 19th Design Automation Conference*, 253-260.

Lee, C.Y.(1961). "An Algorithm for Path Connections and Its Applications". *IRE Trans. on Electronic Computers*, Sept. 1961, 346-365.

Ozden, M. H. (1991). "Graphical Programming of Simulation Models in an Object-Oriented Environment". *Simulation*, **56**, 104-116.

Sargent, Robert G. (1986). "The Use of Graphical Models in Model Validation". *Proc. of 1986 Winter Simulation Conf.*, 237-241.

Stanwood, K., Waller, N. and Marr.,G. (1986). "System Iconic Modeling Facility". *Proc. of 1986 Winter Simulation Conf.*, 531-535.

M++, an Array Language Extension to C++

Ronald Schoenberg

Dyad Software Corp.
16950 - 151st Ave SE
Renton, WA 98058

ABSTRACT

Two major developments in statistical computing in the eighties were array languages and object oriented programming. These developments have been realized only fragmentally until now. M++ is a collection of C++ classes, methods, and functions for array manipulation, linear algebra, eigensystem analysis, matrix factorization, and general numeric and statistical computation. M++ extends C++, creating a powerful, object-oriented array language with direct access to all of the features of C and C++.

1 INTRODUCTION

Scientific and statistical computing have splintered in the eighties into parts: a part using the new array language interpreters such as S, GAUSS, MATLAB, and MATHEMATICA; and another part remaining with FORTRAN. Both of these have ignored, to a large extent, the development of the C programming language, which seems to have captured a major part of the rest of the computing world.

There have been tentative movements toward C from FORTRAN, but the obstacles are formidable. There is a large investment in FORTRAN code that cannot be ignored, nor are C arrays really convenient for matrix computation. C also seems to require a significant amount of training; acquiring FORTRAN is a less intimidating problem.

While many researchers have stayed with FORTRAN, others have been turning to array languages. The ability to write code manipulating arrays of data with a simple, intuitive syntax often proves irresistable. Mathematical expressions translate directly into code and many lines of FORTRAN and C code are replaced by a few statements. Productivity is dramatically improved and this can convert to the undertaking of more advanced projects than would be attempted with FORTRAN or C.

All array languages of any significance, however, are interpreters, have few low level features, tend to be weakly typed, and are not extensible. While array languages provide a convenient method for the manipulation of arrays of data, their syntax may become overburdened when applied to large, complicated tasks.

Many researchers who have moved to the array languages are now beginning to encounter their limitations. They may regret having left FORTRAN's superior numeric standards behind, and they may be frustrated with their inability to develop large, complex applications in the array language environment. On the horizon, though, is something new, an array language extension to C++ called M++ (Dyad Software, 1991), that may alleviate these problems. M++ is a complete array language extension to C++ containing multi-dimensional arrays of all of the C++ built-in data types, along with a full range of statistical and mathematical operators and functions. C++ is a superset of C, giving it and M++ access to all of the low level functionality for which C is well known. Translators and native compilers exist for C++ on most platforms that support C, which means nearly all operating systems that exist, and therefore M++, a non-interpretive, full featured array language with direct access to a powerful, low-level language, will be available on just about every computer.

2 C++, AN OBJECT-ORIENTED EXTENSION OF C

C has a number of features not available in FORTRAN - run-time allocation of memory, more convenient I/O, scope, and structures, for example. Together they may not add up to enough advantage for the researcher to move to C. The features C++ adds to C, however, are substantial and may prove to be worth the attention of the researcher. The implementation of the object-oriented concept in C++ provides for user-defined data types with data-hiding,

derivation and inheritance, function and operator overloading, as well as control over the creation, destruction, declaration, and assignment of objects (Ellis and Stroustrup, 1990). Other implementations of the object-oriented paradigm exist, but C++ is particularly suited for the researcher because of its efficiency and because of the availability of the built-in numerical types and operators inherited from C. Most C++ compilers (not to mention C compilers) have yet, however, to incorporate the IEEE numeric standards that have always been a part of FORTRAN. An exception is the ZORTECH C++ "Science and Engineering" compiler that fully implements the standards set by the ANSI Numerical C Extensions Group, based on the IEEE numeric standard (Ladd, 1991). The researcher typically works with large sets of data, and often the operations on these data can be described mathematically in a simple form. A program written in C, however, must deal with each number in an explicit way. A general program written to compute some basic statistics, for example, must first allocate and initialize memory for every number to be stored. It then reads the numbers in one by one, and provides instructions in loops to compute the statistics, number by number. For example, the following C program reads in data and computes means and standard deviations:

```
#include<stdlib.h>
#include<stdio.h>
#include<math.h>
#include<bios.h>

void main(argc,argv)
int argc;
char **argv;
{
   char *fn;
   fn = argv[1];
   numObs = atoi(argv[2]);
   numVars = atoi(argv[3]);

/*  DECLARATION  */
   FILE * filen;
   int numObs,numVars,i,j,n;
   double **data, **mm, *mn, *sd;

/*  ALLOCATE MEMORY  */
   data = (double**)malloc(numObs*sizeof(double*));
   mm = (double**)malloc(numVars*sizeof(double*));
   for(n = 0; n < numObs; n++)
      data[n] =
        (double*)malloc(numVars*sizeof(double));
   for(i = 0; i < numVars; i++)
      mm[i] =
        (double *)malloc(numVars*sizeof(double));
   mn = (double *)malloc(numVars*sizeof(double));
   sd = (double *)malloc(numVars*sizeof(double));

/*  READ IN DATA  */
   filen = fopen(fn,"r");

   for(n = 0; n < numObs; n++)
     for(i = 0; i < numVars; i++)
        fscanf(filen,"%lf",&data[n][i]);

/*  INITIALIZE ARRAYS  */
   for(i = 0; i < numVars; i++) {
      mn[i] = 0;
      for(j = 0; j < numVars; j++) {
         mm[i][j] = 0;
      }
   }

/*  COMPUTATIONS  */
   for(n = 0; n < numObs; n++) {
      for(i = 0; i < numVars; i++) {
         mn[i] += data[n][i];
         for(j = 0; j < numVars; j++) {
            mm[i][j] += data[n][i]*data[n][j];
         }
      }
   }
   for(i = 0; i < numVars; i++) {
      mn[i] /= numObs;
      for(j = 0; j < numVars; j++) {
         mm[i][j] /= numObs;
      }
   }
   for(i = 0; i < numVars; i++)
      sd[i] = sqrt(mm[i][i] - mn[i]*mn[i]);

/*  PRINT RESULTS  */
   printf("\n\nMeans\n");
   for(i = 0; i < numVars; i++)
      printf(" %f\n",mn[i]);

   printf("\nStandard Deviations\n");
   for(i = 0; i < numVars; i++)
      printf(" %f\n",sd[i]);
}
```

In C++, a class can be designed to take care of much of this tedious work. Initialization and allocation of memory can be handled in the construction of the object. I/O operators can be overloaded to handle the reading in and writing out of objects. Math operators can be overloaded to perform the calculations. All of the loops can be eliminated and essential information about the objects can be hidden away in the object so the researcher doesn't need to be concerned with them once the objects have been declared. The solution of the above problem in M++ with a class called **doubleArray** (for array of double precision numbers) is found in the more readable program below:

```
#include<darray.h>
#include<stdlib.h>
#include<stream.hpp>

void main(int argc, char * argv)
{
   char * fn;
   fn = argv[1];

//  DECLARE AND READ IN DATA
   doubleArray data;
   data.readASCII(fn);

//  COMPUTATIONS
   doubleArray mn,vc,sd;
   mn = mean(data,0);
   vc = transpose(data).product(data)/numObs;
   vc -= transpose(mn)*mn;
   sd = sqrt(vc()(Index(0,numVars,numVars+1)));

//  PRINT RESULTS
   cout << "Means\n" << mn;
   cout << "Standard Deviations\n" << sd;
}
```

Declaration, allocation of memory, and initialization of the arrays are accomplished in single statements replacing many lines of C code. One statement handles the input of the data, dimensioning the array automatically so that command line arguments are no long necessary. The many loops in the C code are reduced to a few lines of code.

The critical feature of C++ is its ability to provide a syntactical interface that fits the conceptual parts of the problem. It is possible to design a set of functions in a non-object-oriented language that performs the above task in just about as few lines of code. There wouldn't, however, be any fit of these functions to the elements of the problem. Each function would require a series of arguments that would have to be documented and referred to for their use because there wouldn't be any natural way to handle them. The researcher's task in a functional language involves assembling and arranging arguments, and thus the problem must be translated into a structural form dictated by the syntax of the programming language.

C++, on the other hand, has a syntax that can be designed to fit the problem. The researcher's problem can be broken down into parts in a natural way. If arrays are a fundamental part of the problem then an array type can be created and the program will now be developed with arrays as a fundamental type.

3 M++, AN ARRAY EXTENSION TO C++

For the researcher the array type is necessary. While C++ offers them an opportunity to design their own such type, they may also turn to a well designed class library such as M++. Three years have been devoted to the development of the M++ class library that turns C++ into a powerful array language with a fundamental array type having four dimensions, but easily extendible to any number of dimensions. It incorporates a full complement of mathematical and statistical operators and functions, including many based on EISPACK and LINPACK.

Using M++ as a base the researcher can go on to more complex abstractions. Consider the problem described in the previous section. The covariance matrix is computed in a standard way. However, the symmetric matrix result entails the calculation of n*(n-1)/2 redundant elements. This duplication of effort as well as certain problems in precision could be avoided if the result could be computed from an update to a Choleski factorization.

To solve this problem, first we create a **Moment** class derived from the M++ **Choleski** class. The **Moment** constructor would take a data set as an argument and compute a Choleski factorization via the update method augmenting the data set with a column of ones. It would then store the result as a private data member in its base class. The **Moment** class would have methods for computing moments, means, covariances, and so on, from the factorization. It would also be able to use the base class methods for computing inverses and determinants.

When actually using this class, analysts wouldn't have to concern themselves with how the data were stored in the object. All they would know is that they have created a **Moment** object that may be interrogated for various kinds of information about the data. For example,

```
#include<darray.h>
#include<dmom.h>
#include<stdlib.h>
#include<stream.hpp>

void main(int argc, char * argv[2])
{
   char * fn;
   fn = argv[1];

//  DECLARE AND READ IN DATA
   doubleArray data;
   data.readASCII(fn);

//  DECLARE MOMENT OBJECT
   doubleMoment M(data);

//  PRINT RESULTS
   cout << "Means\n" << M.mean();
   cout << "Standard Deviations\n" << M.stdDev();
   cout << "Correlation Matrix\n" << M.corr();
}
```

We have now reduced our original problem to the simple act of declaring an object. While we still haven't done anything that couldn't be accomplished by a statistical package, C++ is just beginning whereas the statistical package stops there. Now that we have a **Moment** object, we may treat it like any other object. Operations can be defined to manipulate it. For example, pooled moment matrices can be created by adding **Moment** objects, or they can be updated with more observations by adding the **Moment** object to an array object containing a data set. Or arrays of **Moment** objects can be created and manipulated in higher order array operations. The statistical package is at the end of a creative effort, but C++ is only the beginning of the creative mathematical and statistical imagination, and M++ is a step on the way.

4 CONCLUSION

C++ has opened the way to object-oriented program design for numerical analysis. M++ is the foundation for the application of C++ to numerical problems by providing the array classes for handling memory allocation and initialization as well as the operators and functions for manipulating them. As is, M++ has the functionality of a complete array language, but it needn't stop there. New objects can be derived from the classes available in M++ that fit the problem the analyst has in mind. What is needed? It might be arrays of rational numbers - certain kinds of problems can be posed entirely in rational numbers. Or, perhaps, interval arithmetic is required in which an upper and lower bound is stored rather than a single number. A derived interval class with a set of overloaded operators and

functions would allow an analyst to develop large, complex programs with a syntax that relieved them of having to think about the intervals.

Whatever the analytical problem, the researcher will find in C++ and M++ a powerful tool for solving it.

References

Dyad Software (1991), *M++ Class Library User's Guide*, Dyad Software Corp., Renton, WA.

Ellis, Margaret A. and Stroustrup, Bjarne (1991), *The Annotated C++ Reference Manual.* AT&T Bell Laboratories, Murray Hill, NJ.

Ladd, Scott Robert (1991), *Zortech C++ Compiler V3.0 Numerical Programming Guide.* Zortech, Inc., Woburn, MA.

Building a program for Multifactor Cross-Tabulation : Some Structures & Systems

BP Murphy
Department of Management
University of Western Australia, Nedlands, WA 6009

Introduction

This paper discusses programming principles and practices required to construct a crosstabulation program, a subset of a statistics package substantial enough to illustrate these principles, without involving numerical analysis problems. This is a preface to our construction of a simple but powerful tabulation program via extracts from the code in our own packages (e.g. **StatZ** [1])

Objectives and Limitations

For the present purposes, we envisage users who are moderately experienced researchers, typically wanting many tables of modest complexity from potentially large data - for instance, researchers with surveys of 300 readings on each of 10000 subjects. This is not really ignoring the masses with smaller jobs and less expertise; efficiency considerations will also usually be similar.

The major initial technical decision to be made in the face of limitations concerns data storage. In a sense the decision is finally simple - data cannot be held in memory as it is prone to be too large. Thus it must be treated in segments. 'Paged' memory is an attractive option, but as not all systems are particularly well set up for it, program developers would probably have to implement it themselves for their package input, and it would have system dependencies. Thus the extra programming implied by taking the data in on a per case basis and fully utilising that line to contribute to all the relevant tables, before replacing it with the next line and repeating the table construction calculations, is easily justified, and is in fact the method used by most of the older statistical packages too. Thus our initial program emphasis here is on formal structures in the language Pascal (Wirth, [5]) for the tables description, which dictate the table computations to be done as each data line is read.

We have used Pascal as the programming language to show this work on the grounds that it is the best known language of serious programmers, and more importantly it is a fully structured language. Further discussion on Pascal's general virtues are given in the references [2,3,4].

Data Structure

We only read one case at a time from the data file, so the input structure is simply an array. The tables formed will also be in a single array. Respectively, we define types and variables :

```
type
 DataArray = array[1..maxVars] of real;
 TableArray = array[1..maxCells] of real;

Var DataCase : DataArray; Tables : TableArray;
```

It is of considerable relevance how the second array is actually used; for the moment note again that at any point in the computation the all the tables within the table array are updated using only the data line just read. We assume that data elements are real numbers; more general situations of mixed reals, integers & text, transformations, recodings and selection criteria are merely later refinements.

Essential Structures 1: Record and Set Types in Table Definitions

We consider a table definition as having three components -

1) a **classifying factors** array defining the variables and their order within the table
2) a set defining the **statistics** to be placed in the cell defined by the indices of 1)
3) the **object variable**, if any, to which the statistical computation is applied

The first component is illustrated by the description **AgeGroup * Sex * Country** , say, meaning that we are defininga three way table of 24 cells, where there are, say

4 AgeGroups [<18, 18-37, 38-62, >62 years]
2 Sexes [male female] and
3 Countries [Australia, US, UK] .

If we want simply a table of counts of our cases according to this cross-classification, the statistic required as the second component is 'count', and this is the default computation. However the structure optionally allows us to request means of a (continuous) variable, say Salary, by selecting the statistic to be 'mean' and the component 3 'object variable' as Salary. Component 3 is not relevant unless component 2 is other than 'Count'.

The use of record type constructs, a variable with many component sub-variables, is well entrenched in the structured languages, and we use them heavily throughout our code. It is appropriate here to define a table specification by a record type:

```
type  TableSpec =
      record
        ClassSet : array[1..maxFact] of integer;
        TabSize : integer;
        TabBase : integer;
        StatsWanted : StatSet;
        ObjectVar: integer;
      end;
```

where **TabBase** is the address in the table array of the first cell of the constructed table, and the set

```
type
StatSet=('count', 'mean', 'stdev', 'min', 'max');
```

is the set of possible statistics to be calculated for each object variable given.

This formulation makes it possible to specify many tables to be obtained in a single pass through the data, each specification being stored as an element of the vector of **TableSpec** records

```
var
TabsInPass = array[1..maxTabs] of TableSpec;
```

Each table's memory allocation is visualised as a linear array, and the collection of table arrays are stored contiguously to form the single linear array of all the table cells as declared above. The i'th table then has an origin at location number **TabsInPass[i].TabBase** within this large array. We detail later a further payoff in this structure in that multiple response ('group') variables are efficiently handled within the same structural processing.

At runtime, for each data line we simply cycle through all the elements of the **TabsInPass** array of table definitions. The target cell within the i'th table has an offset from **TabBase** computed from the values of the classification variables found in the data line. The statistic wanted determines what is done in the target cell - if a count is required the cell is merely incremented by unity, while for a mean of Salary, the cell is incremented by the case's value of Salary; or a substitution may be needed if maximum/minimum is required.

As computing efficiency matters do not impact the structural considerations, we pass over them here.

Essential Structures 2: Group Variables

Most surveys use group variables, often more by accident than design. These are of two structurally different, though related, forms.

The first is illustrated by the following. Suppose an opinion pollster conducts a survey on newspaper readership, in which persons are asked to name up to three papers they read regularly. The survey form would have three slots to fill, and these would usually become three variables, say Paper1, Paper2 and Paper3. However these three variables are equivalent in the sense that each has the same range of possibilities - one respondent could name the *Daily Bugle* in Paper2 while another respondent does not have it at all and a third names it in Paper1. The pollster usually wants as one of the results, a table of the form

```
Paper by AgeGp by Sex  ,
```

in effect combining the three simple tables

```
Paper1 by AgeGp by Sex
Paper2 by AgeGp by Sex
Paper3 by AgeGp by Sex  .
```

We solve this by effectively processing each case three times, using Paper1 first time, then Paper2 (with same Age and Sex values), then Paper3. In the present structure this is easy to organise - three tables are formed but the trick is that each is given the same base address. Hence the cumulative table is formed without physically getting the separate constituent tables and adding them. This means we do not have to reserve space for three tables, nor waste time with exceptions-testing code in the innermost loops. All exceptions processing work then occurs in the time non-critical phases of table definition and output configuration.

The second group variable situation is an extension

of the above (conversely, the first is a 'cheap' form of this one). In this, the survey carries a (usually large) list of newspapers to be considered, and the respondents tick all relevant. So we have Paper1, ... Paper25 say, and any number can be ticked by a single respondent. Here we consider the group variable Paper as a 'super variable' having Paper1, ...Paper25 as its 25 'levels'. Again, processing cases is transparent if we define twenty-five tables as above, now with appropriately modified base addresses, and leave the fact that there is only one real result table, a concatenation of the 25, to be easily sorted out at print time (this involves some structural detail for aliased and non-printed tables). The wastage of table definition space is offset by the economy gained by the fact that target index computations for the 25 tables to be updated for a single case have all but one source index identical.

The record structure describing both types of group variables uses a set type variable for group, and is:

```
type  groupvars =
    record
      GrpName        : string[15];
      GrpType        : (e,s)
      GrpLo, GrpHi   : integer;
      GrpIndxs : Array[1..maxInGrp] of integer;
    end;
```

Essential Structures 3: The Driving Program and Overlays

Having considered specific calculation modules, we turn to the main-line or driver program, and the structure it requires.

At one level, the driver is a very simple looping procedure which presents a prompt to the user, and reads in the user's next command. It determines what the command is, and proceeds to call the procedure/module enacting that command. At a second level, it houses global variables, and all utility procedures needed permanently in memory for more than one module. In particular, it holds the case read/select run time procedures, for these will also be needed if in the future we add new statistical modules which must also read the data.

We find this loop structure very useful as it can be extended as the program grows by simply adding new control keys and corresponding procedure calls. It can also be replicated within statistical procedures themselves to handle sub-commands.

On the other hand, modules which are needed in short term only can be put outside the mainline, and this determines another level of structure called overlays or segments, in which independent procedures alternately use the same memory area when they are being executed. The level of power of this structure is that, on most sophisticated compilers on PCs, it can allow sets of procedures to be constructed and compiled as independent units and stored externally to the main program. In execution, the mainline procedures remain in memory at all times, loading an external module into memory only when it is required. The external modules use the same physical memory area, as each overwrites the space used by the earlier one. The program skeleton is

```
Program StatsCrossTables;
{Driver loop and global defintions}
const.....type.....var .....

external procedure Tab(ok : boolean;
external procedure Tran(ok : boolean;

{$I SupportProcs}  {for i/o & general  needs}
{$I SundryStatsProcs} {common simple stats}

var    {Local defintions}
   endIt,  {Quit flag}
   OK :  {operation successful}  BOOLEAN;

begin  {driver}
endIt := false;
 while not endIt do
   begin
   write(outfile,  '==>>');
   readln(wd);   {user give 'command' keyword}
   case left(wd,3) of
    'qui': endIt := true;
    'fil'  : openToRead(FileName, ok);
    'tab': Tables( ok);
    'tra'  : Transforms( ok);
    'lev': defineLevnames( ok);
    'var': defineVarnames( ok);
   end {case}
   end; {while}
write('Goodbye.);
end.   {driver}
```

Such overlaying is essential in constructing large

systems of any kind, though text books on the subject are in fact rare. Note that external modules are not strictly standard in Pascal, but all modern compilers like Apple's MPW do support them (and hence our recent move to Modula2 sytems, where they are defined as standard). A full discussion of these matters is given in [4].

Essential Structures 3: Sets & Pointers

We have referred briefly to sets in earlier sections. This program does not require any more complex usage of sets than that used in the **StatsWanted** item of the **TableSpec** record, but more complex set constructs are useful elsewhere, as shown in [2] for ANOVA and Log-Linear models.

We find pointers very necessary where we must allocate temporary space for large arrays, usually of dimension which cannot be determined until run time. Space is allocated/deallocated in ('spare') heap memory and allowing great flexibility in larger programs. Relevant here is the *LevelsNames* array, an array of text for the levels of a 'factor' or classification variable. The structure is used within the variables information record:

```
type  VariablesRec =
   record
      VarName      : String[15];
      mean  : real;      {if/when available}
      stddev : real;     {if/when available}
      min   : real;      {if/when available}
      max   : real;      {if/when available}
      NoOfLevels     : integer;   {0=not factor}
      LevelsArrayPtr : ^LevelsArray;
   end;
```

and we define the array

```
VarInfo : array[1..maxVars] of VariablesRec
```

to contain useful information about each variable in the data file. The first six items are obvious, and the last points to the

```
LevelsArray= array[1..classes] of string[15];
```

which is allocated at run time when **classes** size is known. This structure also leads to efficiencies of storage as many variables may be allocated the same names (e.g.'Yes', 'No').

Essential Structures 4: Recursive Procedures

For space reasons, discussion of this vital topic must be passed over here. Particularly useful ones in crosstabulation are fully discussed in [2,3].

Essential Structures 5: Graphics User Interface

Besides space problems, there are interlocking reasons for omitting this discussion also - we find that users eventually move to batch methods and so a 'command line' interface, as here, is always also needed. Further, it is relatively easy to put up a grapics interface preprocessor to construct the commands at a later time, in accord with the programming Principle of Successive Refinement.

Summary

We indicate in this note that structures commonly used in developing large commercial packages by professional programmers are equally relevant to statisticians' programming. We also hope that our programs **StatZ** *etc.* using these will be found more than a cerebral exercise.

References

[1] Klobas JE, & Murphy, BP (1990) ***The STATZ Manual***, Westat Associates Pty. Ltd., Nedlands, Western Australia

[2] Murphy B.P., Rohl, J.S., Cribb, B.P.(1986) Recursive Techniques in Statistical Programming. In **Proc. Compstat '86** IASC Conference, Springer-Verlag, Vienna.

[3] Murphy BP ,and Bartlett GA (1988) Further Recursive Algorithms for Multidimensional Table Computations. In **Proc. Compstat '88** IASC Conference, Springer-Verlag, 261-9

[4] Murphy BP & Bartlett GA (1990) *Statistical Platforms - Dreams to Reality*. **Statistical Software Newsletter**,*15,2, 2-7*

[5] Wirth N (1970) **Algorithms + Data Structures = Programs**, Prentice Hall

MPW and **StatZ** and are proprietary products of the software houses indicated.

Experimenting with Semi-Parametric Regression Models and Estimation in "Arizona"

Martin B. Maechler *

Bell Communications Research, 2M-343
445 South Street
Morristown, NJ 07962-1910, USA.
e-mail: maechler@bellcore.com

ABSTRACT

This paper describes how a flexible object-oriented programming environment can greatly enhance possibilities for interactive model building and for the choice, analysis, and estimation of models. In our case we want to model $y = f(x)$ with the non-orthodox requirement that f have as few inflection points as necessary but f is otherwise unconstrained (nonparametric). One possibility is to express f'' as $f''(x) = \pm(x - w_1)\cdots(x - w_J)\exp h_f(x)$, and model h_f as linear spline. There are many ways of dealing with a penalty to be added to the log-likelihood, and of estimating the different groups of parameters. Using the object-oriented CLOS-based ARIZONA environment, these different possibilities are quite easy to investigate.

1 Introduction: Penalizing the Proper Roughness

In order to think about scatter plot smoothing, let us reconsider how we measure smoothness. For convenience, we are going to measure 'roughness' $R[f]$ of a function f as opposed to smoothness. Our approach to estimating f in the model $y_i = f(x_i) + \varepsilon_i$, $i = 1, \ldots, n$, (ε_i are i.i.d. with $\mathbf{E}[\varepsilon_i] = 0$) is a Maximum Penalized Likelihood (MPL) criterion as follows:

$$\min_{f \in C^m[x_1, x_n]} \quad \sum_{i=1}^{n} \rho(y_i - f(x_i)) \quad + \quad \lambda\, R[f]\,. \quad (1)$$

The first term in (1) is the negative log-likelihood when ε_i are i.i.d with density $g(t) \propto e^{-\rho(t)}$. For Gaussian errors it is the usual residual sum of squares. Allowing for a more general ρ, e.g., using "Huber's rho", $\rho_c(x) = \left(x^2 - (|x| - c)_+^2\right)/2$, we make sure that our estimate of f will be robust against outlying observations y_i; see Mächler (1989), and Cox (1983). The smoothing parameter λ determines a balance between fidelity to the data (small residuals) and smoothness of f.

The issue mentioned above can now be stated as

What roughness penalties $R[f]$ are appropriate ?

In nonparametric density estimation, the MPL approach has been investigated in much detail; see Good and Gaskins (1971), Tapia and Thompson (1978), and Silverman (1986), section 5.2, where some of the earlier work is discussed. Although Good and Gaskins (1971) do discuss the choice of penalty, it is usually selected in such a way that the subsequent problem is "tractable".

MPL has a more recent history in the context of regression, but considerable investigation has been done, including generalizations for dealing with generalized linear models; see, e.g., O'Sullivan et al. (1986), Gu (1989) and Wahba (1990), ch. 9. Here, the roughness penalty is always assumed to be the squared seminorm of a (linear) projector in some (function) Hilbert space. The methodology of reproducing kernels leads then to simple characterizations of the solution, sometimes called *generalized splines*.

In contrast, we want to choose a roughness penalty according to a more qualitative notion of smoothness. The penalty which leads to polynomial spline functions, $R[f] = \int f^{(m)^2}(t)\, dt$, ($m = 2$ gives cubic splines), was originally motivated by the fact that $f''(x)$ is in some way "proportional" to the curvature of f at x. Ideally, $R[f]$ would measure an average (squared) local curvature. The domain of integration, here and subsequently, is the full range of $\{x_i\}$. As seen in Mächler (1989) and below, this is not true in general. We have developed a new roughness penalty trying to incorporate a global notion of smoothness. Approximately measuring *change of curvature* instead of curvature leads "naturally" to consideration of *inflection points*, i.e., points where the curve goes from convex to concave or vice versa. The final approach, denoted by *"Wp"*, can be considered a parametric–nonparametric hybrid: Assuming a given number of inflection points J, we consider

*Research supported in part by the National Science Fund of Switzerland

the MPL problem where the roughness penalty now measures the "remaining change of curvature", given J inflection points; J can be varied on top of this as well.

An example with real data is given in figure 1. This data set `hs.data` is available in $\mathcal{S}$ by the statements `s.hs <- sabl(hstart); hs.data <- s.hs $trend + s.hs $irregular`. Cubic splines, "lowess" (Cleveland (1979)) and our "Wp" smoother (with $J = 3$) were tuned to have identical average squared residual. Note that the two classical smoothers suffer somewhat from "erosion" at the two local minima and both have extra inflection points.

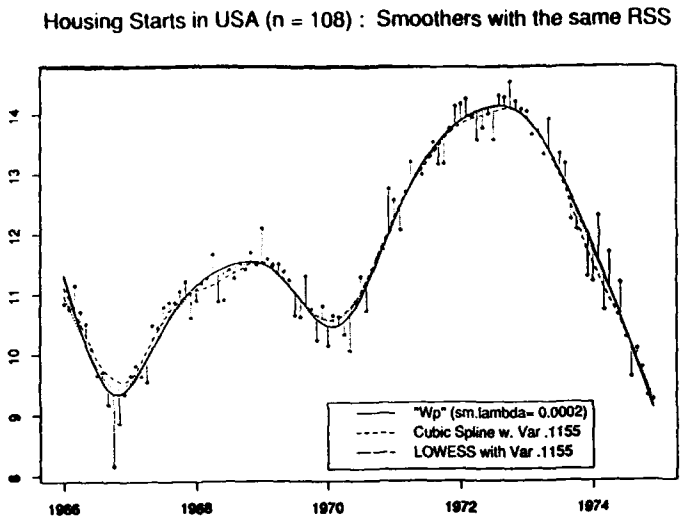


Figure 1: *Example of (deseasonalized) housing starts (in the U.S.) for 108 months. Three different smoothers, tuned to have identical residual sum of squares. "Wp" smoother with $J = 3$, i.e., 3 inflection points.*

Perhaps the most closely related work to ours is work on "isotonic" and "constrained" splines. Wright and Wegman (1980), e.g., prove the existence of splines under restrictions such as (piecewise) monotonicity, convexity, etc. A corresponding algorithm has been provided by Irvine et al. (1986) and put into the IMSL library subsequently (as 'CSCON'). Examples of other approaches using regression splines with a few (possibly) variable knots were taken by e.g., Dierckx (1980), and Ramsay (1988).

These "smooth" curves, however, often are 'nearly' piecewise linear with 'brisk' changes of slope which is very unsatisfactory. This behavior is well explained intuitively: One is looking for a function f minimizing $\int f''^2(t)dt$ under restrictions as $(-1)^j f''(x) \geq 0$ on some sub-intervals $I_j \subset [a, b]$. To solve this problem approximately one might just look for f in each I_j separately requiring, e.g., $f'' \geq 0$, and $f'' = 0$ at the interval boundaries. The minimization of $\int_{I_j} f''^2(t)dt$ now will often yield whole subintervals where $f'' \approx 0$, i.e., f is locally almost linear.

2 "Wp": Change of Curvature Roughness, Inflection Points

The traditional *smoothing splines* approach is motivated by the idea of penalizing high curvature κ with the penalty $R[f] = \int \kappa(t)^2 dt$. Because the curvature is given by $\kappa(x) = f''(x)\left(1 + f'(x)^2\right)^{-3/2}$, it may be approximated by $\kappa(x) \approx c \cdot f''(x)$ (if $f'(x) \approx$ const !) which leads to cubic splines. This approximation to κ need not be good in some cases (see Mächler (1989)) and is used mainly because of the simplicity of f''^2 (and its corresponding solution to the variational problem) compared to κ^2. But the more important issue is

Does high curvature really mean "roughness" ?

Mächler (1989) argues that it may be more 'natural' to take the "standardized Change of curvature"

$$\frac{\kappa'}{\kappa} = f'''/f'' - 3f'f''/(1 + f'^2) \approx f'''/f''$$

as measure of roughness. The approximation $f'''/f'' \approx \kappa'/\kappa$ holds exactly at all the local extrema and inflection points (the most interesting points of the curve) and is qualitatively better than $f''(x) \approx \kappa(x)$ in many situations.

This approximation for the relative change of curvature leads to the preliminary penalty $\tilde{R}[f] := \int (f'''(t)/f''(t))^2\, dt$. Now, let us assume that f has J inflection points, say $w_1, \ldots, w_J$. Equivalently, f'' has exactly the zeros $w_j, j = 1, \ldots, J$. Then f'''/f'' has first order poles at these locations and $\tilde{R}[f]$ "contains J times ∞". But we can "rescale" the problem by expressing the second derivative as

$$\begin{aligned} f''(x) &= \underbrace{\underbrace{s_f}_{\pm 1}(x - w_1)(x - w_2)\cdots(x - w_J)}_{P_w(x)}\; e^{h_f(x)} \\ &= P_w(x)\; e^{h_f(x)}. \end{aligned} \qquad (2)$$

Here, $s_f e^{h_f(x)}$ represents any function with no zeros ($\in \mathbb{R}$), and $P_w(x)$ is a polynomial of degree J. We can now express h_f as $h_f(x) = \log\left(f''(x)/P_w(x)\right) = \log|f''(x)/s_f| - \sum_{j=1}^{J} \log|x - w_j|$. This gives

$$h_f'(x) = \frac{f'''}{f''}(x) - \sum_{j=1}^{J} \frac{1}{x - w_j}. \qquad (3)$$

Expression (3) is f'''/f'' (as in $\tilde{R}[f]$) minus all the poles. If the inflection points $w_1, \ldots, w_J$ are unknown parameters, the choice of the penalty

$$R[f] := \int h_f'(t)^2 \, dt, \tag{4}$$

still penalizes the *change* of curvature and prevents more than J inflection points. The "order" J (the number of inflection points) is assumed to be given. For each J we have a class of functions with a fixed number of inflection points.

In Mächler (1989), the resulting variational problem was considered and a (quite involved) numerical algorithm for its solution was devised. Here, we solve the problem in a restricted (but still rich) function space to make it more feasible for inference. Namely, we model h_f as a polygon, or *linear spline*.

3 "Wp" parametrized; h_f as linear spline

Assume the x_i are ordered and split the data interval into M subintervals $I_k = [t_k, t_{k+1}[$, by knots $x_1 \equiv t_0 < t_1 < \ldots < t_M \equiv x_n$. We now parametrize h_f as a (general) linear spline with knots $\{t_k\}$. In each knot interval, we set

$$h_f(x) = h_k + b_k(x - t_k), \text{ for } x \in I_k. \tag{5}$$

We want h_f to be continuous at all the inner knots, i.e., $h_k(t_{k+1})$ must equal $h_{k+1}(t_{k+1})$, or equivalently,

$$h_{k+1} = h_k + (t_{k+1} - t_k)b_k \equiv h_0 + \sum_{j=0}^{k}(t_{j+1} - t_j)b_j. \tag{6}$$

for $k = 0, \ldots, M-1$. Therefore, h_f is completely specified by the given knots $(t_0, \ldots, t_M)$ and the parameters $(h_0, b_0, \ldots, b_{M-1})$. Even $f(x)$ itself can be seen as a parametric function (though semi-parametric in nature), and, because we restricted h_f appropriately, we can explicitly express $f(x)$ (piecewise as linear plus the product of polynomial and exponential). Because of the double integration from f'' to f, there are two integration constants f_0 and f_0' which are new free parameters for f. The penalty, $\int h_f'^2(t) \, dt$ is trivial to compute, since h_f' is piecewise constant. Our whole MPL criterion (1) is now the minimization of a function of the parameters $(w_1, \ldots, w_J)$, $(b_0, \ldots, b_{M-1})$, h_0, f_0' and f_0 where the last three can easily be determined as (linear or robust, depending on the choice of ρ) regression coefficients (given the other parameters), and where we assume that the "curvature sign" s_f and the knots $t_1, \ldots, t_{M-1}$ are given. Given data, we have the (nonlinear) minimization problem of determining the $\{w_j\}$'s and $\{b_k\}$'s. Also, some investigation about the choice of the knots $\{t_k\}$ (number and location) has to be done and we may want to include this choice into the minimization problem. In the remaining sections, we will see that an object-oriented interactive graphical system such as ARIZONA is ideal for investigating this minimization problem.

4 CACTUS in ARIZONA

Arizona is a software system under development at the University of Washington, Seattle, by John McDonald and coworkers (McDonald (1988)), based on Common Lisp ('defined' in Steele (1990)) and CLOS, the Common Lisp Object System (Keene (1988)). Citing McDonald and Sannella (1991),

> "Arizona is intended to be a portable, public-domain collection of tools supporting scientific computing, quantitative graphics, and data analysis."

The above report and release of ARIZONA are available via anonymous FTP from `belgica.stat.washington.edu` in the directory `pub/az`. More documentation can be found in the LaTeX file collection `doc.tar.Z`. Release 0.0 (as of Feb. 91) contains four modules, TOOLS, GEOMETRY, SLATE and CHART, in hierarchically increasing order. SLATE (relying on GEOMETRY and TOOLS) provides an object oriented user interface to bitmap graphics and event driven user input, i.e., a Lisp toolkit for X11, using CLX. CHART provides output-only high level plot functionality.

CACTUS is a (not yet released) module for numerical and abstract (!) linear algebra and optimization partially described in McDonald (1989). It uses all modules above. The object oriented approach enables flexible experimentation with different optimization methods. Both optimizers and objective functions are objects that can be changed or replaced. At the same time, the modularity of the graphics modules allows *dynamic* graphical monitoring of the minimization process.

5 "Wp" with CACTUS

The interface with CACTUS requires an objective function, i.e., a Lisp function mapping a vector of unknowns (over which to minimize) into a real number. In our case, we have the two parameter vectors $(w_1, \ldots, w_J)$ and $(b_0, \ldots, b_{M-1})$. We now have (at least) two possibilities. First, we may concatenate the two vectors into one vector of unknowns, and pass this to the minimizer(s). Or, we can use two objective functions, one for each of

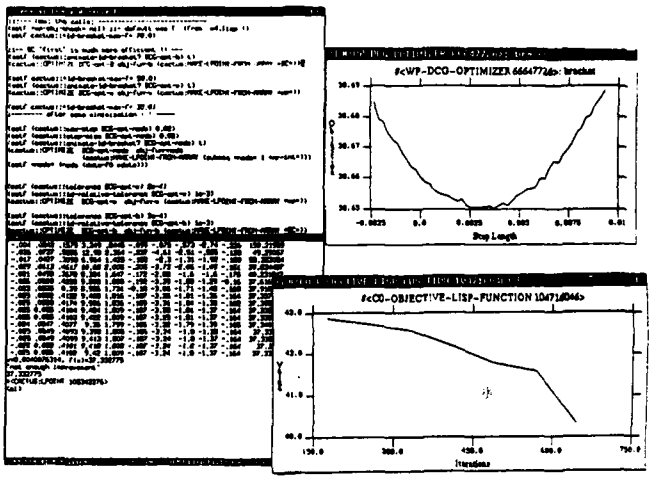


Figure 2: *Example of an* ARIZONA/CACTUS *session. The two plot windows dynamically update themselves displaying the current state of minimization.*

our vectors, with the other held constant, and alternate minimizing the MPL over one vector at a time. The alternating approach may require more minimization steps than the direct approach, but in our case, the MPL criterion can be updated from one evaluation to the next. If only (some) b_k's are changed, the polynomial $P_w(x)$ is unchanged and the evaluation of the objective function is quite fast. Using CACTUS (and modular programming), experiments like these are quite natural to do, since vectors, objective functions and optimizers are "abstract" objects.

A snapshot of an interactive session using ARIZONA is given in figure 2. To the left, we see a two part window comprising the Lisp command interface. In the upper right window, the objective function (the MPL criterion (1)) is shown in the current one-dimensional line search. Below, the objective function is plotted against time (iteration number). Both plot windows dynamically update themselves displaying the current state of minimization.

In conclusion, our "Wp" smoothing methodology leads to a class of interesting minimization problems. The flexibility and interactive nature of ARIZONA/CACTUS make it an attractive environment for investigating a variety of questions arising in our class of problems.

References

Cleveland, W. S. (1979). Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*, 74:828–836.

Cox, D. D. (1983). Asymptotics for M-type Smoothing Splines. *Annals of Statistics*, 11:530–551.

Dierckx, P. (1980). An Algorithm for Cubic Spline Fitting with Convexity Constraints. *Computing*, 24:349–371.

Good, I. J. and Gaskins, R. A. (1971). Non-parametric Roughness Penalties for Probability Densities. *Biometrika*, 58:255–277.

Gu, C. (1989). Generalized Spline Models: A Convenient Algorithm for Optimal Smoothing. Technical Report 853, Department of Statistics, University of Wisconsin, Madison.

Irvine, L. D., Marin, S. P., and Smith, P. W. (1986). Constrained Interpolation and Smoothing. *Constructive Approximation*, 2:129–151.

Keene, S. E. (1988). *Object-oriented programming in Common Lisp: a programmer's guide to CLOS.* Symbolics Press and Addison-Wesley, Reading, MA.

Mächler, M. B. (1989). *'Parametric' Smoothing Quality in Nonparametric Regression: Shape Control by Penalizing Inflection Points.* Dr. diss. nr. 8920, ETH Zurich, Statistik, ETH-Zentrum, CH-8092 Zurich, Switzerland.

McDonald, J. A. (1988). An outline of Arizona. In *Computer Science and Statistics: Proc. 20th Symp. on the Interface*, pages 282–291, Washington, D.C. ASA. Also Tech. Rept. 131, Dept of Statistics, U. of Washington.

McDonald, J. A. (1989). Object-oriented programming for linear algebra. *SIGPLAN Notices (Proceedings OOPSLA '89)*, 24(10):175–184. also Tech. Rept. 175, Dept. of Statistics, GN-22, U. of Washington, Seattle, WA 98195.

McDonald, J. A. and Sannella, M. (1991). Arizona overview and notes for release 0.0. Technical report, Dept. of Statistics, U. of Washington.

O'Sullivan, F., Yandell, B. S., and Raynor, W. J. (1986). Automatic Smoothing of Regression Functions in Generalized Linear Models. *Journal of the American Statistical Association*, 81:96–103.

Ramsay, J. O. (1988). Monotone Regression Splines in Action (with Discussion). *Statistical Science*, 3:425–459.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman Hall, London.

Steele, G. (1990). *Common Lisp, The Language.* Digital Press, second edition.

Tapia, R. A. and Thompson, J. R. (1978). *Nonparametric Probability Density Estimation.* John Hopkins Univ. Press, Baltimore.

Wahba, G. (1990). *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional conference series in applied mathematics.* SIAM.

Wright, I. W. and Wegman, E. J. (1980). Isotonic, Convex and Related Splines. *Annals of Statistics*, 8:1023–1035.

The Symbolic Computation of Asymptotic Expansions

James E. Stafford and David F. Andrews
University of Toronto

Abstract

We describe a collection of procedures, coded in Mathematica, for the systematic computation of asymptotic expansions common in statistical theory and practice. The procedures permit the expansion of maximum likelihood estimates, the associated deviance or drop in likelihood, and more general functions of random variables involving one or an arbitrary number of parameters. General expansions, written in standard notation, are produced by these procedures and they can be evaluated for a particular distribution through the specification of the appropriate moment generating function. These procedures perform complex, lengthy derivations in a fraction of the time it takes by hand, with very little chance for error. They permit the statistician to concentrate on the structure of a symbolic calculation rather than on the detail of term by term evaluation. The procedures are illustrated with examples involving general laws.

1 Introduction

Much of statistical theory and practice is based on asymptotic expansions. Many programs are available to assist in the numerical evaluation of such expansions but there is a need for computational tools to assist in their derivation and symbolic evaluation.

Symbolic computation is an underused facility available to research statisticians. Packages like Mathematica, Maple and Reduce are typically used to perform limited tasks such as obtaining derivatives or integrals. Application of such packages in more general problems is uncommon, although some work does exist (Kendall 1988 and 1990, Barndorff-Nielsen and Blaesild 1986). This may be due to the sparsity of problems that are broad enough to merit the development of symbolic tools to solve them. Deriving asymptotic expansions is such a problem.

The derivation of asymptotic expansions is typically a simple, but tedious, task usually involving complicated and laborious algebra. Consider, for example, the calculation of the asymptotic expansion for the maximum likelihood estimate standardized by the sandwich estimator, $S(\hat{\theta}) = \{\frac{-1}{n}\partial^2 L(\theta)/\partial\theta^2|_{\theta=\hat{\theta}}\}^{-2}\{\frac{1}{n}\sum_{i=1}^{n}\{\partial L_i(\theta)/\partial\theta|_{\theta=\hat{\theta}}\}^2\}$. L is the log-likelihood function with components L_i. The parameter θ is scalar and the expansion is to include the n^{-1} term. An expansion for the maximum likelihood estimate may be accomplished by expanding the score function in a Taylor series about $(\hat{\theta}-\theta)$ to the third order and inverting it. The observed information may be expanded in $(\hat{\theta}-\theta)$ to the second order and the composition of this series with that of the maximum likelihood estimate, while retaining terms of order n^{-1}, will give an asymptotic expansion for the observed information. The expansion for the estimate of the variance of the score, $\frac{1}{n}\sum_{i=1}^{n}\{\partial L(\theta)/\partial\theta|_{\theta=\hat{\theta}}\}^2$, may be found in a similar way. This must then be composed with an expansion for $(1+x)^{\frac{-1}{2}}$ and then multiplied by the expansions for the maximum likelihood estimate and the observed information. When this is done, and terms of order n^{-1} retained, the expansion is compete. By hand, the expansion requires several hours of algebra, checking and rechecking. When done by computer the correct expansion is obtained in a couple of minutes,

```
Lawley[AsymptoticExpansion[(thetahat-
theta)*Sandwich(thetahat)^(-1/2),2]]
```

$$\frac{l_\theta}{\sqrt{\lambda_{\theta,\theta}}} - \frac{\lambda_{\theta\theta,\theta}l_\theta^2}{\lambda_{\theta,\theta}^{3/2}} - \frac{\lambda_{\theta\theta\theta}l_\theta^3}{2n\lambda_{\theta,\theta}^{3/2}} + \frac{l_\theta l_{\theta,\theta}}{2\lambda_{\theta,\theta}^{3/2}} +$$

$$\frac{3\,l_\theta l_{\theta,\theta}^2}{8\,\lambda_{\theta,\theta}^{5/2}} - \frac{l_\theta^2 l_{\theta\theta,\theta}}{\lambda_{\theta,\theta}^{5/2}} - \frac{l_\theta^2 l_{\theta\theta\theta}}{2\,\lambda_{\theta,\theta}^{5/2}} +$$

$$\frac{3\,\lambda_{\theta\theta,\theta}l_\theta^2 l_{\theta,\theta}}{2\,\lambda_{\theta,\theta}^{7/2}} + \frac{3\,\lambda_{\theta\theta\theta}l_\theta^2 l_{\theta,\theta}}{12\,\lambda_{\theta,\theta}^{7/2}} - \frac{\lambda_{\theta\theta,\theta}l_\theta^2 l_{\theta\theta}}{\lambda_{\theta,\theta}^{7/2}} + \frac{\lambda_{\theta\theta\theta}l_\theta^2 l_{\theta\theta}}{\lambda_{\theta,\theta}^{7/2}} -$$

$$\frac{\lambda_{\theta\theta,\theta\theta}l_\theta^3}{2\,\lambda_{\theta,\theta}^{7/2}} + \frac{\lambda_{\theta\theta\theta,\theta}l_\theta^3}{2\,\lambda_{\theta,\theta}^{7/2}} - \frac{\lambda_{\theta\theta\theta\theta}l_\theta^3}{3\,\lambda_{\theta,\theta}^{7/2}} + \frac{3\,\lambda_{\theta\theta,\theta}^2 l_\theta^3}{2\,\lambda_{\theta,\theta}^{9/2}} - \frac{\lambda_{\theta\theta\theta}^2 l_\theta^3}{2\,\lambda_{\theta,\theta}^{9/2}}$$

In general the task of deriving asymptotic expansions, though simple, is very large. When performed by hand, the probability of error increases with the length of the expression. Few statisticians would willingly take on this challenge. Fewer would do it correctly. The computer procedures presented here perform these expansions in a small fraction of the time it takes by hand.

Asymptotic expansions are useful both in their general form and in particular cases. General expansions provide an avenue for the comparison of statistics and families of distributions. For example, in a robustness study the above expression could be compared to the expansion of the maximum likelihood estimate when it is standardized by the observed information. In particular cases, explicit formulae, such as Edgeworth approximations, are required

for application. Such formulae are easily obtained by evaluating general formulae through the specification of the appropriate moment generating function.

Section 2 presents a summary of the notation and procedures used. Section 3 presents applications of the procedures to derive general asymptotic expansions. Section 4 contains concluding remarks.

2 Notation and Procedures

We shall assume that the reader is familiar with the summation convention and use the notation below which is similar to that of Lawley(1956).

$l_r = n\,\partial L_1/\partial\theta_r, \quad L_{rs} = n\,\partial^2 L_1/\partial\theta_r\partial\theta_s, \quad L_{rst} = n\,\partial^3 L_1/\partial\theta_r\partial\theta_s\partial\theta_t,$

$L_{r,s} = n\,\partial L_1/\partial\theta_r \partial L_1/\partial\theta_s, \quad L_{r,st} = n\,\partial L_1/\partial\theta_r \partial^2 L_1/\partial\theta_s\partial\theta_t \quad etc.,$

$\lambda_{rs} = E(L_{rs}), \quad \lambda_{rst} = E(L_{rst}), \quad \lambda_{r,s} = E(L_{r,s}), \quad \lambda_{r,st} = E(L_{r,st}), \quad etc.,$

$l_{rs} = L_{rs} - \lambda_{rs}, \quad l_{rst} = L_{rst} - \lambda_{rst}\, l_{r,s} = L_{r,s} - \lambda_{r,s}, \quad l_{r,st} = L_{r,st} - \lambda_{r,st}, \quad etc.,$

$\lambda^{rs} = \lambda_{rs}^{-1}.$

The procedures that have been written to derive asymptotic expansions, produce output that is peculiar to the symbolic package Mathematica. A procedure, called Lawley, was written to translate output into the above notation thus making it readable and greatly simplifying the preparation of this paper. Styles of other authors can be emulated. Computer input is presented in which subscripts are denoted as lists in braces so that they may be entered from a keyboard. For example, l_{rs} is represented by l[{r,s}]. Greek letters are spelled out. The following display illustrates both the input and output of the system.

```
Lawley[lambda[{r,s}]]

-λ_rs
```

The only hand operation required to produce a typeset version of this paper was to insert line breaks in long equations.

2.1 Maximum Likelihood Estimates

To obtain an expansion for the maximum likelihood estimate we use an algebraic analogue of Fisher's scoring method. Consider the algorithm based on

$$\hat\theta_r[0] = \theta_r$$

$$\Delta\hat\theta_r[i] = -\lambda^{rs} l_s(\hat\theta[i-1]) = -\lambda^{rs}\{\, l_s(\hat\theta[i-2]) - l_{st}\lambda^{tu} l_u(\hat\theta[i-2]) + \cdots \}$$

$$\hat\theta_r[i] = \hat\theta_r[i-1] + \Delta\hat\theta_r[i]$$

$\Delta\theta[i]$ is of order $n^{\frac{(i)}{2}}$. The series converges to the maximum likelihood estimate if it is unique.

The function *MaxLikEst[i]* returns the expansion of the maximum likelihood estimate correct to order $n^{\frac{-i}{2}}$

2.2 General Functions

The change in log-likelihood, $2\,[L(\hat\theta) - L(\theta)]$ may be expanded in a Taylor series in $\hat\theta$–θ and the maximum likelihood estimate, to the correct order, substituted. This requires a procedure which calculates expansions of functions, whose arguments are themselves expansions, while retaining only terms of the required order. The following definition of the expansion of $f(g)$ where, $g[i]$ is an expansion of g correct to order $n^{\frac{-i}{2}}$, can be used

$$ExpandF[f,\, g,\, order] = \sum_{i=1}^{order}\Big(\prod_{j=1}^{i}\frac{g_{k_j}[order+1-i]}{i!}\Big) E(f_{k_1,k_2,\cdots k_i}) +$$

$$\sum_{i=1}^{order}\Big(\prod_{j=1}^{i}\frac{g_{k_j}[order-i]}{i!}\Big)[f_{k_1,k_2,\cdots k_i} - E(f_{k_1,k_2,\cdots k_i})],$$

where $f_{k_1,k_2,\cdots k_i} - E(f_{k_1,k_2,\cdots k_i})$ has order $n^{\frac{-1}{2}}$. Repeated use of this procedure allows the expansion of such functions as the square root of the observed information.

2.3 Expected Values

The expected value operator is defined by the usual basic relations:

$$E(X+Y) = E(X) + E(Y),$$

$$E(a\,X) = a\,E(X),$$

$$E(a) = a$$

The only difficulty arises with terms involving sums of arbitrary length. These are evaluated using

$$Expect\Big[\prod_{k=1}^{m}\sum_{i=1}^{n} X_{k,i}\Big] = \sum_{s\in S}\frac{n!}{s_0!}\, Expect\Big[\prod_{k=1}^{m} X_{k,i_k}\Big]$$

where s_0 denotes the number of distinct subscripts in s, and S denotes the set of ordered sets of m subscripts s such that the first subscript is 1 and any following subscript is at most 1 larger than the maximum of those preceding. The

$X_{k,i}$'s are independent of each other with respect to i. Here they are derivatives of the log-likelihood, although other applications are possible. The definition is quite general and leads to direct algorithmic implementation. The product of m sums is evaluated with less than m! terms, independent of n.

The above procedures are central building blocks derivation of asymptotic expansions and may be used for the to obtain expansions of many common statistics.

2.4 Identities

Reduction of expressions, usually moments or cumulants, makes use of many identities involving expected values of the derivatives of L. All of these follow directly from *one* basic identity:

$$\frac{\partial E(g)}{\partial \theta_r} = E(\frac{\partial g}{\partial \theta_r}) + E(g\frac{\partial L}{\partial \theta_r})$$

This is used to define $E(\frac{\partial g}{\partial \theta_r})$.

Bartlett identities are a group of identities useful in simplifying expressions. They equate to zero linear combinations of the expectation of derivatives of the log-likelihood function. The k^{th} order Bartlett identity is derived by,

i. noting that $\int e^{l(X;\theta)}dx = c$,

ii. differentiating both sides, with respect to k non-distinct components of θ,

The first two of these identities are well known:

$\lambda_r = 0$, and

$\lambda_{r,s} = -\lambda_{rs}$

Machines are useful for repetitive tasks -- there are infinitely many more of these identities.

3 Examples

In the following examples we derive asymptotic expansions for the maximum likelihood estimate, the likelihood ratio statistic and its expected value. The expressions are correct to order n^{-1} and the examples are displayed showing the computer input and output.

```
Lawley[MaxLikEst[2]]
```

$$-\left(l_s\lambda^{rs}\right) + l_s l_m \lambda^{rs}\lambda^{tm} - \frac{l_s l_m \lambda_{tm}\lambda^{rs}\lambda^{tm}}{2}$$

```
Lawley[ExpandF[2(L(thetahat)-
L(theta)),thetahat,2]]
```

$$-\left(l_r l_s \lambda^{rs}\right) + \frac{l_m l_s l_w \lambda_{rs}\lambda^{rm}\lambda^{sw}\lambda^{tr}}{2} -$$

$$\frac{l_m l_s l_w \lambda_{rs}\lambda^{rm}\lambda^{sw}\lambda^{tr}}{3} + nl_m l_s l_w l_{gh}\lambda_{rs}\lambda^{hr}\lambda^{gs}\lambda^{tm}\lambda^{tw} -$$

$$\frac{l_g l_h l_s l_w \lambda_{rstu}\lambda^{rh}\lambda^{sg}\lambda^{tw}\lambda^{tu}}{6} + \frac{l_g l_h l_s l_w \lambda_{rstu}\lambda^{rg}\lambda^{sw}\lambda^{th}\lambda^{tu}}{12} -$$

$$l_r l_s l_{tu}\lambda^{ru}\lambda^{tw} - \frac{l_m l_s l_w \lambda_{rs}\lambda^{rm}\lambda^{sw}\lambda^{tr}}{2} + l_r l_s l_{tu}\lambda^{rt}\lambda^{tu} +$$

$$l_r l_s l_{tu}\lambda^{rt}\lambda^{tu} + \frac{l_g l_h l_s l_w \lambda_{rstu}\lambda^{sw}\lambda^{th}\lambda^{tr}}{6} +$$

$$\frac{l_g l_h l_i l_j \lambda_{rs}\lambda_{tuvw}\lambda^{rt}\lambda^{su}\lambda^{tv}\lambda^{tw}}{4} - \frac{l_g l_h l_i l_j \lambda_{rs}\lambda_{tuvw}\lambda^{rv}\lambda^{su}\lambda^{tg}\lambda^{tw}}{6} -$$

$$\frac{l_g l_h l_i l_j \lambda_{rs}\lambda_{tuvw}\lambda^{rg}\lambda^{su}\lambda^{tv}\lambda^{tw}}{4} - \frac{l_g l_h l_i l_j \lambda_{rs}\lambda_{tuvw}\lambda^{rg}\lambda^{su}\lambda^{tv}\lambda^{tw}}{6} +$$

$$\frac{l_g l_h l_i l_j \lambda_{rs}\lambda_{tuvw}\lambda^{rt}\lambda^{su}\lambda^{tv}\lambda^{tw}}{4} + \frac{l_g l_h l_i l_j \lambda_{rs}\lambda_{tuvw}\lambda^{rv}\lambda^{su}\lambda^{tw}\lambda^{tv}}{4} -$$

$$\frac{l_g l_h l_i l_j \lambda_{rs}\lambda_{tuvw}\lambda^{rg}\lambda^{su}\lambda^{tv}\lambda^{tw}}{4} - \frac{l_g l_h l_i l_j \lambda_{rs}\lambda_{tuvw}\lambda^{rg}\lambda^{su}\lambda^{tv}\lambda^{tw}}{6} -$$

$$l_r l_s l_{tu} l_{vw}\lambda^{rv}\lambda^{tw}\lambda^{tu} + l_r l_s l_{tu} l_{vw}\lambda^{rw}\lambda^{tv}\lambda^{tu} +$$

$$l_r l_s l_{tu} l_{vw}\lambda^{rv}\lambda^{tw}\lambda^{su} - l_r l_s l_{tu} l_{vw}\lambda^{rw}\lambda^{tv}\lambda^{su} -$$

$$l_r l_s l_{tu} l_{vw}\lambda^{rw}\lambda^{tv}\lambda^{su}$$

```
Lawley[Expect[ExpandF[L(thetahat)-
L(theta),thetahat,2]]]
```

$$I(rr) + \frac{\lambda_{rstu}\lambda^{rs}\lambda^{tu}}{4} + \frac{4\lambda_{rstu}\lambda_{vwx}\lambda^{rv}\lambda^{sw}\lambda^{tx}}{3} -$$

$$\frac{25\lambda_{rstu}\lambda_{vwx}\lambda^{rv}\lambda^{sw}\lambda^{tx}}{12} - \lambda_{rstu}\lambda_{vwx}\lambda^{rv}\lambda^{sw}\lambda^{tx} -$$

$$\frac{\lambda_{rstu}\lambda_{vwx}\lambda^{rv}\lambda^{sx}\lambda^{tw}}{3} + \frac{3\lambda_{rstu}\lambda_{vwx}\lambda^{rs}\lambda^{tv}\lambda^{tw}}{2} +$$

$$\frac{\lambda_{rstu}\lambda_{vwx}\lambda^{rv}\lambda^{sw}\lambda^{tx}}{6} +$$

$$\frac{7\lambda_{rstu}\lambda^{rv}\lambda^{sw}\lambda^{tx}\lambda_{vwx}(\,)}{6} - \frac{2\lambda_{rstu}\lambda^{rs}\lambda^{tv}\lambda^{tw}\lambda_{vw}(\,)}{3} +$$

$$\frac{7\lambda_{\dots}\lambda^{\dots}\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)}{6} - \frac{2\lambda_{\dots}\lambda^{\dots}\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)}{3} +$$

$$\frac{\lambda_{\dots}\lambda^{\dots}\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)}{2} + \frac{\lambda_{\dots}\lambda^{\dots}\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)}{2} +$$

$$\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)\lambda_{\dots}(.) - \lambda^{\dots}\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)\lambda_{\dots}(.) -$$

$$\lambda^{\dots}\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)\lambda_{\dots}(.) - \frac{7\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)}{6} + \frac{2\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)}{3} -$$

$$\frac{\lambda^{\dots}\lambda^{\dots}\lambda_{\dots}(.)}{2}$$

This last expression agrees with Lawley's equation 4 except for an error in the printed version of his term divided by 6. Our algorithm for collecting terms is not quite efficient; a further reduction is possible. However most of the reduction from 6^6 terms to 20 has been achieved.

Evaluating expressions, like the ones above, for specific distributions simply requires a translation of the summation convention to an actual sum and then a term by term evaluation of the sum. Discussion of such procedures may be found in Andrews, Stafford, and Wang(1991).

4 Concluding Remarks

Symbolic computation is a useful tool that can relieve statisticians from hours of tedious and laborious algebra. The use of the above procedures greatly reduces the likelihood of producing errors in expansions. In fact, their use lead to the discovery of errors in Lawley(1956) and the printed versions of the Ph.D. dissertations by DiCiccio and Ferguson. These procedures reproduce expressions from Barndorff-Nielsen & Cox(1989), DiCiccio(1984), Ferguson(1989) and McCullagh(1987) without error. Such tools are meant to accelerate research and encourage ambitious projects in the area of asymptotics. The development of symbolic procedures in other areas of research is highly recommended.

Acknowledgements

We are very grateful to Rob Tibshirani for his help in the preparation of this paper. We are also grateful to NSERC of Canada for their support of our research.

References

Andrews, D. F., & Stafford, J. E., (1991), Tools for the symbolic computation of Asymptotic expansions. *Technical Report, Department of Statistics, University of Toronto.*

Andrews, D. F., Stafford, J. E., & Wang Y., (1991), On Reading Lawley(1956): An Application of Symbolic Computation. *Technical Report, Department of Statistics, University of Toronto.*

Barndorff-Nielsen, O. and Blaesild, P. (1986) A Note on the Calculation of Bartlett Adjustments. *J. R. Statist. Soc.* B, **48**, 351-358.

Barndorff-Nielsen, O. and Cox, D. R. (1989) *Asymptotic Techniques for Use in Statistics.* New York: Chapman and Hall.

DiCiccio, T. J. (1984) On Parameter Transformations and Interval Estimation. *Biometrica*, **71**, 477-485. p

Ferguson, H. (1989). *Asymptotic Properties of a Conditional Maximum Likelihood Estimator*, Ph. D thesis, University of Toronto.

Lawley, D. N. (1956), A general method for approximating the distribution of likelihood ratio criteria, *Biometrika*, **43**, 295-303.

Kendall, W. S. (1988). Symbolic computation and the diffusion of shapes of triads. *Advances in Applied Probability*, **20**, 775-797.

Kendall, W. S. (1990). The diffusion of Euclidean Shape. In: *Disorder in Physical Systems*, (Grimmett, G. and Welsh, D. ed.) pp.203-217, Oxford: University Press.

McCullagh, P. (1987) *Tensor Methods in Statistics.* New York: Chapman and Hall.

Wolfram, S. (1988). *Mathematica A System for Doing Mathematics by Computer.* Addison Wesley, NewYork.

MAXIMUM LIKELIHOOD ESTIMATION OF THE ACCURACY RATES OF DIAGNOSTIC TESTS BY MEANS OF THE EM ALGORITHM

T. S. Weng
FDA/CDRH/OST/Division of Biometric Sciences
12200 Wilkins Ave., Rockville, MD 20852

Abstract

An EM algorithm has been developed for computing the maximum likelihood estimates and standard errors of the accuracy rates (i.e., sensitivity and specificity) of a new diagnostic test and an established reference test, based on the outcomes of both tests when applied to individuals sampled from an arbitrary number of populations with different prevalence rates of a given disease. This algorithm is heuristically appealing in that it also estimates the prevalence rate within each source population and aids the perception of the effects of numerical constraints imposed on some of the rate parameters. An example is given to illustrate the application of this algorithm to practical clinical situations.

NOTE: The views presented here are those of the author. No support or endorsement by the Food and Drug Administration is intended or should be inferred.

1 Introduction

Consider a new diagnostic test T which is to be evaluated against an established reference test R when both tests are used to detect a disease D in a given population of which each individual is assumed to be either diseased (D1) or non-diseased (D2). If the outcome from each individual is also expressed dichotomously as positive (T1) or negative (T2), the accuracy of T may then be assessed by its **sensitivity** (η) and **specificity** (ξ) defined as $\eta = \Pr(T1|D1)$ and $\xi = \Pr(T2|D2)$, respectively. These quantities are generally referred to as the **accuracy rates** of T, and their complements $\alpha = 1-\xi$ and $\beta = 1-\eta$, as the **error rates** of T. For the case in which the accuracy rates of both T and R are unknown, Hui and Walter [1] employed the standard maximum likelihood method to estimate the accuracy rates of T and R when both tests were simultaneously applied to individuals sampled from two populations with different prevalence rates of D. This method, however, has been found to have too many problems to be practical [2].

In this paper, an attempt is made to expand the method of Hui and Walter into a more widely applicable alternative for evaluating the performance characteristics of clinical diagnostic tests. Specifically, the purpose is to compute the maximum likelihood estimates (MLE's) and standard errors (SE's) for the accuracy rates of both T and R, as well as for the different prevalence rates of D in an arbitrary number of populations (NB, Hui and Walter's formulas for computing the MLE's are limited to only two populations), presuming that R may be less than perfect in accuracy and the disease state of each individual is not known. Thus, instead of working directly with the likelihood equations per se, an EM algorithm [3] has been worked out which is easy to program and to embed with numerical constraints selectively imposed on the rate parameters. This approach is extremely versatile, permitting the user to extend the applicability of the maximum likelihood principle to the computation of MLE's and SE's for the rate parameters in a wide variety of cases encountered in clinical practice, such as: (1) when both R and T have unknown accuracy rates; (2) when R has known accuracy rates; (3) when both R and T have a specificity equal to 1; (4) when R alone has a specificity equal to 1; or (5) when some or all of the source populations have known disease prevalence. For Cases (2) through (5), in particular, the value(s) of the known parameter(s) can be embedded as numerical constraint(s) in the EM algorithm set up for Case (1), thereby to yield "constrained estimates" for the remaining parameters.

2 Nature of Problem

The essential problem here is to evaluate the new test T against the reference test R by comparing the MLE's of these tests' accuracy rates (namely, η_h and ξ_h, where $0 \le \eta_h \le 1$, $0 \le \xi_h \le 1$, h = t for test T or r for test R, and $1 < \eta_r + \xi_r \le 2$), based on random samples of size N_k drawn each from K populations with prevalence rates π_k, k=1,...,K, where at least two π_k's must be distinct from each other. If no numerical constraints are imposed, the parameters to be estimated may be represented by the (K+4)-vector

$$\theta = (\pi_1, \ldots, \pi_k, \eta_t, \eta_r, \xi_t, \xi_r).$$

3 Nature of Data

When the sample from the k-th population is subjected to testing by T and R, the outcome data may be summarized by the 4-vector of counts

$$y_k = (y_{k11}, y_{k12}, y_{k21}, y_{k22}),\ k=1, \ldots, K,$$

where the values 1 or 2 of the second and third subscripts denote, respectively, the outcomes T1 or T2 for test T and R1 or R2 for test R. The data vector y_k, though observable, is "incomplete" in the sense of Dempster et al. [3]. It should be noted that each y_{kij} count (i, j=1, 2) can be regarded as a pooling of two unobservable component counts x_{kij1} and x_{kij2}, where the fourth subscript indexes the unknown disease state of the individual tested, with 1 or 2 denoting D1 or D2, respectively. Thus, in contrast to y_k, the unobservable 8-vector of counts

$$x_k = (x_{k111}, x_{k112}, \ldots, x_{k112}),\ k=1, \ldots, K,$$

is referred to as the "complete" data vector.

4 Theoretical Basis

The EM algorithm developed here is based on the idea of Dempster et al. [3]. To fix the idea, let **A** and **B** denote, respectively, the sample spaces of the random vectors **X** and **Y** with the associated polynomial distributions $P_X(x|\theta)$ and $P_Y(y|\theta)$. For the problem in question, **X** is not directly observable but some image of $\mathbf{X} = x \in \mathbf{A}$ can be observed in the form $\mathbf{Y}(x) = y \in \mathbf{B}$, where the mapping **Y**: $\mathbf{A} \to \mathbf{B}$ is many-to-one. Now consider finding the MLE for θ utilizing y instead of x, of which the latter is only known to lie in a subset A(y) defined by the mapping **Y**: $\mathbf{A} \to \mathbf{B}$. In this context, the idea of the EM algorithm is to utilize the fact that the likelihood of y: $L(y|\theta) = \Pi_k P_Y(y_k|\theta)$ is related to that of x: $L(x|\theta) = \Pi_k P_X(x_k|\theta)$ by the equation

$$L(y|\theta) = \Sigma_{A(y)} L(x|\theta)$$

and to find a value $\hat{\theta}$ of θ which maximize $\log L(y|\theta)$ by iteratively maximizing the expected value of $\log L(X|\theta)$ given **Y**=y. Specifically, it proceeds by introducing an initial estimate $\theta^{(0)}$ and generates a sequence $\{\theta^{(n)}\}$ by repeating the following double step at each iteration:

E-step: Evaluate $Q(\theta|\theta^{(n-1)}) = E[\log L(X|\theta)|y, \theta^{(n-1)}]$
M-step: Find $\theta = \theta^{(n)}$ to maximize $Q(\theta|\theta^{(n-1)})$

Continue until $\{\theta^{(n)}\}$ converges.

5 Derivations for EM Procedure

The overall data, incomplete as well as complete, may be respectively denoted by a 4K-vector

$$y = (y_1, y_2, \ldots, y_K)$$

and by an 8K-vector

$$x = (x_{1111}, x_{1112}, \ldots, x_{K221}, x_{K222}).$$

The likelihood for θ given **X**=x, say, is proportional to

$$\begin{aligned} L(x|\theta) = \Pi_k &(\pi_k\eta_t\eta_r)^{x_{k111}} (\pi_k\eta_t\beta_r)^{x_{k121}} \\ &\times (\pi_k\beta_t\eta_r)^{x_{k211}} (\pi_k\beta_t\beta_r)^{x_{k221}} \\ &\times (\tau_k\alpha_t\eta_r)^{x_{k112}} (\tau_k\alpha_t\xi_r)^{x_{k122}} \\ &\times (\tau_k\xi_t\alpha_r)^{x_{k212}} (\pi_k\xi_t\xi_r)^{x_{k222}}, \end{aligned}$$

where $\tau_k = 1 - \pi_k$ (k=1,...,K), $\alpha_h = 1 - \xi_h$, and $\beta_h = 1 - \eta_h$ (h=t or r). For the parameters in θ to be estimable it requires that $K \geq 2$ and that at least two of the π_k's be distinct from each other. For the special case K=1, appropriate numerical constraints may be imposed on some of the accuracy rates of R and/or T.

The E-step consists of setting the components of x equal to their conditional expectations given **Y**=y. For k=1,...,K, this yields

$$x_{k111}^{(n)} = E(X_{k111}|y, \theta^{(n-1)})$$

$$= \frac{y_{k11}\pi_k^{(n-1)}\eta_t^{(n-1)}\eta_r^{(n-1)}}{\pi_k^{(n-1)}\eta_t^{(n-1)}\eta_r^{(n-1)} + \tau_k^{(n-1)}\alpha_t^{(n-1)}\alpha_r^{(n-1)}}$$

$$x_{k112}^{(n)} = y_{k11} - x_{k111}^{(n)}$$

.....

$$x_{k222}^{(n)} = y_{k22} - x_{k221}^{(n)}.$$

The M-step is executed by equating to zero the gradient vector $(\partial/\partial\theta)\log L(x^{(n)}|\theta^{(n-1)})$ and solving the resulting equation for $\theta^{(n-1)}$. The solution, denoted by $\theta^{(n)}$, is an

improved estimate for θ of which the components are expressed as follows:

$$\pi_k^{(n)} = (\Sigma_i \Sigma_j x_{kij1})/N_k, \quad k=1, \ldots, K,$$

$$\eta_t^{(n)} = (\Sigma_k \Sigma_j x_{k1j1}^{(n)})/N_{D1}^{(n)}$$

.....

$$\xi_r^{(n)} = (\Sigma_k \Sigma_i x_{ki22}^{(n)})/N_{D2}^{(n)}$$

where $N_{D1}^{(n)} = \Sigma_k \Sigma_i \Sigma_j x_{kij1}^{(n)}$ and $N_{D2}^{(n)} = \Sigma_k \Sigma_i \Sigma_j x_{kij2}^{(n)}$. Here $N_{D1}^{(n)}$ and $N_{D2}^{(n)}$ are readily identified as the estimates for the total numbers of the diseased (D1) and the non-diseased (D2), respectively. It is also noted that

$$N_{D1}^{(n)} + N_{D2}^{(n)} = \Sigma_k N_k = N,$$

which is the grand total of all individuals tested.

6 Standard errors & Confidence Intervals

Following Louis [4], let

$$\hat{x} = E(x|y, \hat{\theta}) = (\hat{x}_{1111}, \hat{x}_{1112}, \ldots, \hat{x}_{1112}),$$

where $\hat{\theta} = (\hat{\pi}_1, \ldots, \hat{\pi}_K, \hat{\eta}_t, \hat{\eta}_r, \hat{\xi}_t, \hat{\xi}_r)$ is the MLE for θ obtained at the last EM iteration, and let $S(x, \theta)$ and $H(x, \theta)$ denote respectively the gradient vector of $\log L(x|\theta)$ and the negative of the associated second derivative matrix (also known as the curvature matrix). Then the observed information matrix for θ given the data vector y is expressed as

$$I(\hat{\theta}) = \text{Diag}\{E[H(X, \theta)|y, \hat{\theta}] - \text{Var}[S(X, \theta)|y, \hat{\theta}]\}$$

and the diagonal elements of $I(\hat{\theta})$ as

$$I(\hat{\pi}_k) = (1/\hat{\pi}_k \hat{f}_k)[N_k - (1/\hat{\pi}_k \hat{f}_k) \Sigma_i \Sigma_j (\hat{x}_{kij1} \hat{x}_{kij2}/y_{kij})], \quad k=1, \ldots, K,$$

$$I(\hat{\eta}_t) = (1/\hat{\eta}_t^2) \Sigma_k \Sigma_j [\hat{x}_{k1j1} - (\hat{x}_{k1j1} \hat{x}_{k1j2}/y_{k1j})] + (1/\hat{\beta}_t^2) \Sigma_k \Sigma_j [\hat{x}_{k2j1} - (\hat{x}_{k2j1} \hat{x}_{k2j2}/y_{k2j})]$$

.....

$$I(\hat{\xi}_r) = (1/\hat{\xi}_r^2) \Sigma_k \Sigma_i [\hat{x}_{ki22} - (\hat{x}_{ki21} \hat{x}_{ki22}/y_{ki2})] + (1/\hat{\alpha}_r^2) \Sigma_k \Sigma_i [\hat{x}_{ki12} - (\hat{x}_{ki11} \hat{x}_{ki12}/y_{ki1})].$$

Being diagonal, the information matrix $I(\hat{\theta})$ can be easily inverted to give an asymptotic estimate for the variance-covariance matrix of $\hat{\theta}$.

Since the MLE's $\hat{\pi}_k$ (k=1,...,K), $\hat{\eta}_t$, $\hat{\eta}_r$, $\hat{\xi}_t$, and $\hat{\xi}_r$ are approximately normally distributed with large N, confidence intervals for these estimates may also be easily obtained. Taking, for example, the estimated sensitivity $\hat{\eta}_t$ of test T, the 95% confidence interval for its expected value η_t may be calculated from

$$\Pr\{-1.96 \le (\hat{\eta}_t - \eta_t)/\hat{\sigma}_t \le 1.96\} = 0.95,$$

where $\hat{\sigma}_t$ is the standard error of $\hat{\eta}_t$ obtained as the positive square root of $1/I(\hat{\eta}_t)$, the observed variance of $\hat{\eta}_t$. Standard errors and confidence intervals for the remaining MLE's can be calculated in similar fashion.

7 Example

The data in Table I are reproduced from Table 1 of Gart and Buck [5]. These data have been rearranged to keep with the format and notations adopted in this paper.

Table I.

Outcomes of VDRL (T) and FTA (R) slide tests for syphilis from a sample of the population of Maichew, Ethiopia (Source: Buck & Spruyt [6], cited by Gart & Buck [5])

k (age grp)	Test Outcomes T1R1	T1R2	T2R1	T2R2	Total
1 (5-14)	1	10	4	62	77
2 (15-24)	5	5	2	31	43
3 (25-34)	14	14	6	27	61
4 (35-44)	20	17	5	19	61
5 (45+)	18	9	5	17	49

T1R1 = Positive to both T and R;
T1R2 = Positive to T but negative to R; etc.

Here the new diagnostic test VDRL (coded T) was to be evaluated against the reference test FTA (coded R) for its accuracy in detecting syphilis based on a random sample of individuals from the town of Maichew, Tigre Province, Ethiopia. The random sample had been stratified by age decade beginning with age 5. Following Gart and Buck who posited that $\eta_r = 0.95$ and $\xi_r = 0.90$ on the basis of past experience, we embedded these specified values as constraints in the EM algorithm and estimated the appropriate sets of parameters from each age group as well as from all age groups combined. The results are shown in Table II, along with those reported in Gart and Buck. We then remove the constraints from the EM

Table II.
Parameter estimates ± SE for the data of Table I.

Source	Age group	π_1	π_2	π_3	π_4	π_5	η_t	η_r	ξ_t	ξ_r
	1	.0000±.033	---	---	---	---	---	0.95*	.8571±.040	0.90*
	2	---	.0735±.066	---	---	---	1.0000±.854	0.95*	.8919±.058	0.90*
Gart &	3	---	---	.2681±.071	---	---	.8059±.149	0.95*	.6681±.040	0.90*
Buck*	4	---	---	---	.3645±.074	---	.8621±.099	0.95*	.5400±.124	0.90*
	5	---	---	---	---	.4346±.084	.8455±.091	0.95*	.6223±.095	0.90*
Weighted Mean		---	---	---	---	---	.8459±.061	---	.7675±.024	---
	1	.0000±.014	---	---	---	---	1.0000	0.95*	.8572±.040	0.90*
	2	---	.1094±.003	---	---	---	.9999±.075	0.95*	.8918±.004	0.90*
EM-de-	3	---	---	.2681±.068	---	---	.8058±.132	0.95*	.6680±.075	0.90*
rived*	4	---	---	---	.3645±.072	---	.8624±.095	0.95*	.5402±.085	0.90*
	5	---	---	---	---	.4346±.081	.8453±.098	0.95*	.6752±.096	0.90*
	All	.0000±.016	.1027±.056	.2633±.066	.3859±.073	.4358±.082	.8792±.058	0.95*	.7539±.030	0.90*
EM-der#	All	.0140±.027	.1823±.075	.4953±.083	.6779±.078	.6476±.086	.8091±.048	.6254±.053	.8758±.031	.9451±.022

*Constrained with $\eta_r = 0.95$ and $\xi_r = 0.90$
#Unconstrained

iterative procedure, thereby to compute the unconstrained estimates using data from all age groups. The results are given at the bottom of Table II. As can be seen from Table II, the EM-derived, constrained estimates are all closely comparable to those estimates obtained by Gart and Buck. In addition, the constrained EM estimates computed from the combined data of all age groups are also seen to be quite comparable to the weighted means of Gart and Buck. Specifically, the sensitivity and specificity of T are given as $\hat{\eta}_t = 0.8792 \pm 0.058$ and $\hat{\xi}_t = 0.7539 \pm 0.030$ compared to Gart and Buck's weighted means $\hat{\eta}_t = 0.8459 \pm 0.061$ and $\hat{\xi}_t = 0.7675 \pm 0.024$, respectively. The question now arises as to how reliable the constrained estimates are. Let us address this question by comparing them with the unconstrained EM estimates. First of all, let us construct a 95% confidence interval for the sensitivity and specificity of R utilizing the standard errors associated with the unconstrained estimates $\hat{\eta}_r = 0.6254$ and $\hat{\xi}_r = 0.9451$, yielding (0.5224, 0.7285) and (0.9016, 0.9885) for η_r and ξ_r, respectively. It is no small surprise to find that none of the specified values $\eta_r = 0.95$ and $\xi_r = 0.90$ is contained in the corresponding confidence interval. We are thus led to infer that the specified values did not fit the data well and that the unconstrained estimates for the sensitivity and specificity of T, namely, $\hat{\eta}_t = 0.8091 \pm 0.048$ and $\hat{\xi}_t = 0.8758 \pm 0.031$, would be more preferable to the constrained ones. It also follows that most of the constrained estimates for the prevalence rates of syphilis in different age groups (or subpopulations) may have been underestimated in the light of their counterparts obtained by the unconstrained EM procedure.

8 References

[1] Hui, SL, Walter, SD. Estimating the error rates of diagnostic tests. Biometrics, 1980; 36: 167-171.

[2] Schlain, B. Statistical estimation of diagnostic error rates for Chlamydia tests (Paper presented to NCCLS Subcommittee on Qualitative Testing, Washington, D.C., November 1990)

[3] Dempster, PA, Laird, NM, Rubin, DB. Maximum likelihood from incomplete data via the EM algorithm. J. R. Statist, Soc, B., 1977; 39: 1-38.

[4] Louis, TA. Finding the observed information matrix using the EM algorithm. J. R. Statist, Soc, B., 1982; 44: 226-233.

[5] Gart, JJ, Buck, AA. Comparison of a screening test and a reference test in epidemiologic studies. II. a probabilistic model for the comparison of diagnostic tests. American Journal of Epidemiology, 1966; 593-602.

[6] Buck, AA, Spruyt, DJ. Seroreactivities in the venereal disease research laboratory slide test and the fluorescent treponemal antibody test. Amer. Jour. Hyg, 1964; 80: 90-102.

Improved Methods for Estimating Parameters in Discrete Data Analysis

DAVID J SCOTT
School of Information and Computing Sciences
Bond University
Gold Coast, Queensland
Australia

WANG DONG QIAN
La Trobe University
Bundoora, Victoria
Australia

1 Introduction

For data which take the form of a two-way contingency table, many authors have examined models other than the common model of independence of two classifications. In the literature are quasi independence models, clustered sampling models, intraclass models, and the Bradley-Terry model, among others. In order to estimate model parameters, iterative methods are required, and this leads to the problem of developing efficient algorithms.

2 Quasi Independence Models

If there are cells in a contingency table which *a priori* have a zero count, and there are cells in that row or column which have non-zero counts, the independence model is not appropriate. To cope with this problem of so-called structural zeroes, the notion of quasi independence is useful. In a quasi independence model the row and column classifications are independent, provided the cells with *a priori* zero counts are ignored.

Examples given in the literature of quasi independence models include the random pairing models of Larntz and Weisberg (1976) and de Jong, Greig and Madan (1983), the mover-stayer model as discussed by Morgan and Titterington (1977) and the model of Lemon and Chatfield (1971), which is an alternative to a Markov chain model. Larntz and Weisberg's model can be obtained from Lemon and Chatfield's by folding along the main diagonal.

These models are all for data which take the form of a contingency table with entries on the main diagonal which are zero *a priori*, with the exception of Larntz and Weisberg's model, where the entries on or below the main diagonal are all zero *a priori*.

In order to fit these models to data, various methods have been proposed. Iterative proportional fitting (IPF) is commonly used. IPF requires that the model be log linear, which is not the case for de Jong, Greig and Madan's model. The Newton-Raphson method converges quickly, but it is not easy to implement if the Hessian is not diagonal. The Hessian is diagonal for the mover-stayer model and de Jong, Greig and Madan's random pairing model, but not for the other models. Fixed point iterations have been used by a number of authors, but these can be slow to converge. Brown (1974) developed a method for dealing with *a priori* ze-

roes, which iterates over the cells which are known to be zero. Brown's method becomes less and less efficient with an increasing number of zero cells. de Jong, Greig and Madan (1983) developed a method for fitting their random pairing model which involves a reparameterisation, then fixed point iteration. This method can also be adapted to fit other quasi independence models.

It can be shown that when the table is symmetric, the parameter estimates obtained for Lemon and Chatfield's (1971) model are identical with those obtained from fitting the mover-stayer model. This means that the models of both Lemon and Chatfield, and Larntz and Weisberg may be fitted by symmetrising the data and fitting the mover-stayer model. Thus a readily implementable Newton-Raphson approach is available for these models.

All the methods so far discussed estimate a probability distribution and involve a multivariable iteration. The authors have developed new methods for fitting quasi independence models which require the solution of a nonlinear equation in a single unknown. This equation is readily solved using Newton's method. This gives fast, very easily-implemented methods. No programming is required, and well-known packages such as Minitab or a spreadsheet may be used to do the calculations.

3 Other Models

In the clustered sampling model with clusters of size two, if a number of clusters are observed, and each member of the cluster is classified according to some characteristic, the data take the form of a two-way contingency table. Cohen (1976) has given a method which requires a two-stage iteration procedure, with one stage being the solution of a non-linear equation. Using a reparameterisation the authors were able to reduce the computations required. A two-stage iteration is still required, but only simple expressions need to be evaluated at any stage. This work is reported in Scott and Wang (1990).

Attempts were made to develop improved methods for intraclass models (see Haber(1982)) and the Bradley-Terry model, without success.

References

[1] Cohen, J.E. (1976). The distribution of the chi-squared statistic under clustered sampling. *J. Amer. Statist. Assoc.* **71**, 665–670.

[2] Brown, M.B. (1974). Identification of the sources of significance in two-way contingency tables. *Appl. Statist.*, **23**, 405 413.

[3] de Jong, P., Greig, M., and Madan, D. (1983). Testing for random pairing. *J. Amer. Statist. Assoc.*, **78**, 332–336.

[4] Haber, M. (1982) Testing for independence in intra class contingency tables. *Biometrics*, **38**, 93-103.

[5] Larntz, K. and Weisberg, S. (1976). Multiplicative models for dyad formation. *J. Amer.Statist. Assoc.*, **71**, 455–461.

[6] Lemon, R.E. and Chatfield, C. (1971). Organisation of song in cardinals. *Anim. Behav.* **19**, 1–17.

[7] Morgan, B.J.T. and Titterington, D.M. (1977).A comparison of iterative methods for obtaining maximum likelihood estimates in contingency tables with missing diagonal. *Biometrika*, **64**, 265–269.

[8] Scott, David J., and Wang Dong Qian (1990). An iterative method for estimation of parameters in a clustered sampling model. *Aust. J. Statist.*, **32**, 317-325.

Short-run Stock Market Forecasting with Adjusted Insider Trading Data

H. D. Vinod K. Dadak

Department of Economics
Fordham University
Bronx, NY 10458

Abstract

This paper is concerned with the open-market transactions of corporate insiders. The Securities and Exchange Commission (SEC) publishes information on the buying and selling activities of insiders, which market analysts use to uncover insider sentiment about the prospects of their own corporations and of the entire market. However, the law requires that these transactions be reported to the SEC by the tenth day of the month following a transaction. Moreover, many insiders do not comply with this regulation. Therefore, the available data are always out-of-date in a random manner. We also found that the time lags in reporting buy and sell transactions are different. Since the available data are truncated, it was necessary to adjust the observed probability density function (pdf). We used distributed lag models to study their out-of-sample forecasting performance.

1. Introduction

Important officers of a corporation, called "insiders," trade on the stock market in the securities of the firms they work for according to their own hunches about the company, the overall market, personal financial needs and other circumstances. Market analysts abstract from the personal behavior of insiders and the prospects of the companies they work for by aggregating the data on trading by these investors. Stock market analysts routinely use the insider trading data as indicators of major trends and turning points in the market. Insider trading has also been the subject of many academic studies. The overwhelming majority of these papers supports the notion that insider trading provides valuable long-term information. For instance, Finnerty (1976) showed the usefulness of insider trading information for the purpose of selecting individual securities. The prices of stocks purchased by insiders tend to appreciate faster than the stock market, while securities sold by these investors tend to do worse than the overall market. A study by Seyhun (1988), in turn, showed that insider trading provides advanced signals about the stock market movements. According to Seyhun, insiders increase purchases before stock market rallies and increase sales before stock market corrections.

However, there is a significant time lag between actual insider transactions and their full reporting by the SEC. Also, the inflow of reports is subject to ups and downs due, for instance, to deadline effects on reporting, etc. Moreover, our study of time lags in reporting showed different distributions for buy and for sell transactions. For instance, the mean time lag between the actual transaction date and the filing date with the SEC was only 30.3 days for sales and as much as 32.6 days for purchases.

Many of the above factors are subjective in nature and do not exhibit steady patterns. This makes it very difficult to capture fully their effects. For example, an attempt to explain the insider behavior with the help of Rao's (1965) weighted distributions, as in Vinod (1991), has not yielded useful patterns.

Instead of focusing attention on explaining insider behavior, in this paper we study the information content in insider activity and its effect on market agents. We consider the problem of forecasting Standard and Poor's 500 Index with the help of insider trading data. Thus, we make insider transactions a part of our conditioning set. However, the reporting lag and different lag distributions for purchase and sale transactions mean that the forecast of major turning points and trends based directly on the initial and incomplete SEC insider trading data may be misleading. Therefore, this paper attempts to answer the following questions: (i) Are the available insider trading data worth using? (ii) How should we distill useful information regarding short-term market

trends from insider data? (iii) How do we evaluate the information objectively?

2. Data Analysis

We studied the phenomenon from January 6 through May 29 of 1987, or during 101 trading days (N). We took the insider trading data from a computer tape provided by the SEC. The tape showed that corporate officers executed 22,509 open-market transactions over this period. The data on insider trading over these 101 business days are separated into purchase (PUR) and sale (SALE) transactions. Thus, we visualize two massive matrices with rows representing dates when transactions occurred, and columns showing dates when transactions were reported. So, $\text{pur}(i,j)$ and $\text{sale}(i,j)$ indicate the number of transactions executed on day i and reported on day j. Note however, that because of the reporting lag, $\text{pur}(i,j)$ and $\text{sale}(i,j) = 0$ for all $j \leq i + \text{MINLAG}$, where MINLAG is the minimum time lag between the transaction and the arrival of the insider report at the SEC (at least one day).

Now define cumulative sum matrices

$$\text{CUMPUR}(i,j) = \sum_{j=i}^{N} \text{pur}(i,j)$$

$$\text{CUMSALE}(i,j) = \sum_{j=i}^{N} \text{sale}(i,j)$$

for purchases and sales, respectively, giving the cumulative number of buy and sale transactions for day i as known on day j. For instance, the elements along principal diagonals of the matrices ($j = i$) show the information on insider buying or selling activity as known on the same day.

At any point, one can obtain data on these initial cumulative daily numbers with a time lag (L) represented by $(j - i)$. The greater the L, the more accurate the data we can read from these matrices. However, our objective is to obtain a good approximation of the true insider activity as early as possible, because only the information that is not widely distributed among investors really matters in the marketplace. Therefore, we seek an expansion of the early data which would allow us to predict the actual cumulative amounts of purchases and sales as they are eventually reported. We propose smoothed reciprocals of smoothed lag distributions for purchases and sales as our expansion factors.

3. Smoothed Expansion Factors

From a study of the 22,509 insider transactions we constructed two vectors, separate for purchases and sales, showing the number of transactions reported to the SEC with a certain time lag in days. For example, of the 8,192 buy transactions, only two reports came within one day of the actual transaction date. Similarly, out of the 14,317 sale transactions, only one was reported within one day. Most filing (80 percent) is done with a lag of fifteen days or more. The data on the number of purchases and on the number of sales reported with a certain time lag are denoted by PUR(LAG) and SALE(LAG), respectively, where LAG = 1, 2, ..., N. We obtain separate "unadjusted" lag distributions for purchases and sales, denoted by UP(LAG) and US(LAG), by dividing PUR(LAG) and SALE(LAG) by the total number of purchases and sales, respectively. These aggregated data are found to be representative of the fundamental lag structure.

In principle there is a separate lag distribution for each row of data, and it is not a reliable guide for any other day's lag distribution. The flow of data is erratic, however, and, for that reason, our lag distributions UP(LAG) and US(LAG) are not sufficiently smooth to be a useful guide to the distribution of an arbitrary lag. To overcome this difficulty we use Tukey's (1977) "smoother" called 3RSSH. Here, 3R stands for the moving median of three consecutive values repeated three times, which is followed by an "end correction." SS stands for "split-smooth" applied twice, and, finally, H stands for "hanning" or a weighted average of consecutive three values with weights 0.25, 0.50 and 0.25. We denote AP(LAG) and AS(LAG) these "adjusted" smoothed UP(LAG) and US(LAG), respectively. We use the reciprocals of AP(LAG) and AS(LAG), which are smoothed again, as our expansion factors. The expansion factors are positive and decline to one as we get more complete data on insider activity, i.e., $\lim_{\text{LAG}\to\infty} \text{AP(LAG)} = 1$ and $\lim_{\text{LAG}\to\infty} \text{AS(LAG)} = 1$

Applying the expansion we get

$$\text{P}(i,j) = \text{CUMPUR}(i,j)/\text{AP(LAG)}$$

$$\text{S}(i,j) = \text{CUMSALE}(i,j)/\text{AS(LAG)}$$

where $\text{LAG} = j - i + 1$ for $j \geq i$.

Finally, we calculate the adjusted purchase-to-sale ratios (PSR's) for each day.

$$\text{PSR}(i,j) = \text{P}(i,j)/\text{S}(i,j) \qquad (1)$$

Statistical properties of PSR's may be studied by bootstrapping surveyed in Vinod (1992).

The following section discusses an econometric application of the above procedure to the study of insider trading, where the data on purchase-to-sales ratios of insider transactions are necessarily truncated because of

the lag between transaction dates and their reporting to the SEC.

4. An Application of the Expansion to Stock Market Forecasting

In order to answer the questions that the introduction poses, we employ a model combining insider trading data with the level of interest rates, represented here by the end-of-the-day yield on the 30-year Treasury Bond.

Of the 22,509 insider open-market transactions executed over the period of January 6 through May 29, 1987, only 3,590 buys and 6,887 sales were reported to the SEC by the last day of the above time range. This particular subset is the basis for our calculations of PSR's and for the Ordinary Least Squares (OLS) estimation.

The reader should note that the above structure of transactions is consistent with the overall pattern of insider trading as reported by various stock market forecasters. They agree that on average there are two sales for each purchase. The disparity between the number of purchases and sales stems from the fact that insiders can obtain shares of their companies through non-open-market transactions, such as various incentive plans, pension plans, etc.

The PSR's are calculated on a daily basis. However, the data on the most recent insider activity is very limited, and it is normal for the reported number of total sales or buys to be zero. In the former case the PSR's are not defined, in the latter, the ratios are equal to zero. Nevertheless, in both instances, we set such PSR's to 0.

A researcher faces a dilemma here: either to wait for more data and to deal with more reliable numbers, or to make early predictions and to risk significant errors. On the basis of empirical tests we choose the lag of fifteen days to be an optimal one. For the 15-day lag we have enough data to construct a sufficient number of PSR's, and at the same time we are close enough to actual insider activity to be able to use this information to our advantage. We propose a new technique for adjusting for the time lag between the transaction date and the report date.

We propose the following so-called rational distributed lags model (see Judge et al., 1985) for forecasting the S&P 500 index denoted by s_t.

$$s_t = \beta_0 + \beta_1 s_{t-1} + \beta_2 y_t + \beta_3 p_{t-15} + \epsilon_t \qquad (2)$$

where y_t is the end-of-the-day yield on the 30-year Treasury Bond and p_t is the PSR and ϵ_t is the error term. Writing (2) in terms of the lag operator, we have

$$(1 - \beta_1 L)s_t = \beta_0 + \beta_2 y_t + \beta_3 p_{t-15} + \epsilon_t \qquad (3)$$

Note, that dividing both sides by $(1 - \beta_1 L)$ (2) represents an infinite order lag structure with exponentially declining weights provided $|\beta_1| < 1$.

Tables 1 and 2 report our Ordinary Least Squares estimates of (2) when the data points having zero PSR's are omitted. Equation I refers to the OLS estimates for the expanded PSR's, equation II has unadjusted PSR's, and equation III has PSR's omitted. Note that the last equation uses the same input data matrix as the first two.

Table 1

Eq.	Coef.	Estimate	St. Error	t-stat.
I	β_0	−14.39	32.05	−0.45
	β_1	0.92	0.04	20.51
	β_2	4.79	5.28	0.91
	β_3	2.10	1.18	1.79
II	β_0	−5.63	34.06	−0.17
	β_1	0.91	0.52	17.35
	β_2	4.31	5.62	0.77
	β_3	−0.80	1.61	−0.50
III	β_0	−6.95	32.58	−0.21
	β_1	0.91	0.05	19.94
	β_2	4.25	5.41	0.79

Table 2

Eq	$\bar{R}^2$	F-val	SFE	SDFE	MFE	H-M
I	0.952	299.0	13.20	2.19	4.70	0.78
II	0.950	425.4	14.82	2.30	5.11	0.56
III	0.936	208.9	13.72	2.28	4.61	0.78

The results in Table 1 clearly show the superiority of the model containing adjusted PSR's, although β_3 in eq. I is statistically significant only at the 10 percent level, when we compare the t-statistic with the t tables. It should be noted, however, that the t-statistic for adjusted PSR's is better than that of β_2, the yield on Treasury Bonds. From the t-statistics alone, one may be tempted to omit the Treasury Bond yield variable. However, its omission leads to worse overall out-of-sample forecasts. That the interest rate has an important effect on stock prices is also well known in the financial economics literature (see Lorie et al., 1985).

The adjusted PSR's, as expected, vary directly with the S&P 500 index, while the unadjusted ratios show an inverse relationship to the stock market. The former indicates that insiders correctly anticipate changes in the market direction, and therefore, investors can learn from them. The latter, contrary to the popular view, would

result in losses to insiders and those who follow in their footsteps.

The abbreviations SFE, SDFE, MFE, and H-M in Table 2 stand for the sum of out-of-sample forecast errors, the standard deviation of out-of-sample forecast errors, the maximum of out-of-sample forecast errors, and the Henriksson-Merton (1981) test, explained below, respectively.

The Henrikson-Merton statistic indicates the results of a nonparametric (distribution free) test of timing. The test is based on the direction of the predicted movement, not the magnitude. It is not sensitive to the distribution of stock prices, it does not assume symmetry in the ability to make "up" forecasts and "down" forecasts, and it allows for nonstationarities. The null hypothesis is that the forecaster has no skills and forecasts randomly. Therefore, sometimes he can make correct predictions.

Let N_1 and N_2 be the number of down and up observations, respectively. Thus, the number of total observations $N = N_1 + N_2$. Let n_1 denote the number of correct down predictions that must be in the range of

$$\underline{n}_1 \equiv max(0, n - N_2) \leq n_1 \leq min(N_1, n) \equiv \bar{n}_1$$

It has been shown that n_1 has a *hypergeometric distribution*

$$\binom{N_1}{x}\binom{N_2}{m-x}\Big/\binom{N}{m}$$

where m is the number of forecasts made, and $x = n_1$ is the number of correct forecasts. It is argued that the one tail test is relevant here. One rejects the null hypothesis if the number of correct forecasts exceeds a number $x^*(c)$ based on

$$\sum_{x=x^*}^{\bar{n}_1}\binom{N_1}{x}\binom{N_2}{m-x}\Big/\binom{N}{m} = 1 - c$$

The test gives a confidence score on a scale of 0 to 1, with a high score for procedures that predict the direction most accurately. Our tabulated results suggest a confidence score of 0.78 for the adjusted PSR's. It is based on the nine out-of-sample forecasts. By contrast, for unadjusted data the confidence score is only 0.55. Hence, the H-M test supports the usefulness of the expansion.

SFE, SDFE, and MFE statistics are smaller for the equations containing the expanded ratios than those for the model having unadjusted PSR's. Similarly, the comparison of these statistics for the equations II and III, except for MFE, is favorable for the model with adjusted PSR's. Thus, our study generally shows that the inclusion of adjusted PSR's improves the stock market forecast.

Conclusions

Our paper illustrates the benefits one can obtain from the application of smoothing techniques developed in the context of robust statistical estimation. Expanded PSR's (1) not only offer a better short-run forecast, but they give researchers the correct picture of insider sentiment. On the other hand, the unadjusted data show no statistically significant relationship with the stock market. Therefore, we posit that adjusted PSR's should be included in models forecasting the stock market in the near-term.

References

Finnerty, Joseph E. (1976), "Insiders and Market Efficiency," *Journal of Finance*, 31, pp. 1141 48.

Henriksson, R.D. and R.C. Merton (1981), "On Market Timing and Investment Performance, II: Statistical Procedures for Evaluating Forecasting Skills," *Journal of Business*, 54, 513-533.

Judge, G.G., W.E. Griffiths, R.C. Hill, H. Lutkepol, and T.C Lee (1985), *The Theory and Practice of Econometrics*, 2nd edition, J. Wiley, New York.

Lorie, J.H., P. Dodd, M.H. Kimpton (1985), *The Stock Market: Theories and Evidence*, 2nd edition, Dow Jones-Irwin, Homewood.

Rao C.R. (1965), "On Discrete Distributions Arising out of Methods of Ascertainment," in G.P. Patil (ed.) *Classical and Contagious Discrete Distributions*, Calcutta Statistical Publishing Society, 320-333, rep. in Sankhya, A, 27, 311-324.

Seyhun, H. Neyat. (1988), "The Information Content of Aggregate Insider Trading," *Journal of Business*, 61, pp. 1-24.

Tukey, J.W. (1977), *Exploratory Data Analysis*, Addison Wesley Pub. Co., Reading.

Vinod, H.D. (1991), "Rao's Weighted Distribution in Econometrics: An Application to Unemployment Statistics and Okun's Law," *Journal of Quantitative Economics*, 7, (to appear).

Vinod, H.D. (1992), "Bootstrap, Jacknife Resampling and Simulation Methods: Application in Econometrics," in C.R. Rao, G.S. Maddala and H.D. Vinod (eds) *Handbook of Statistics: Econometrics*, vol 13, North Holland, (to appear).

MULTIPLE SENSOR FUSION

Deva C. Doss*
Quality & Research Centre For Productivity
Canadian Union College
College Heights, AB, Canada T0C 0Z0

Abstract

In a study of two sensors polling data on emitting targets one sensor may observe a target while the other may fail; even if both sensors observe a target, then there is a random noise that distorts the picture. In this paper a general algorithm is developed for detecting the pairs of observations made by the sensors on same targets and for fusing each pair as a single target.

1 Introduction

MICOM/TACOM initiated in 1982 the Setter program in which three sensors, namely, radio frequency interframeter, non-imaging and radar poll emitting targets, detect for potential threat in an air-land battle scenario, and provide the operator/gunner a synergistic effect through a microprocessor based data management system and an integrated display with enhanced real-time integrated information. The sensors exhibit variation in their detecting capabilities of different types of targets and also under different terrain and weather conditions. The complexity of the problem arises when the different sensors detect the same targets with varying noise levels or one sensor detects a target while others may fail to detect. The first task is to determine the number and positions of targets from the data collected independently by multiple sensors; this should take into consideration the fact that a single target may be distorted as multiple targets by sensors and *vice versa* in the presence of random noise accompanying the data. Once the resolution of targets is accomplished by the processor, the task of determining the nature and priority will follow.

In section 2 we define the problem and set forth some criteria and their rationale for fusion of the data. In section 3 we present a general discussion of the problem which leads to the development of an algorithm given in section 4. This algorithm finds the number and locations of targets from the data received from two independent sensors for identification of targets. Finally, in section 5 we discuss the lines of research to be followed in the future.

2 Criteria

Let X_i, i = 1, . . . , n be observations detected by a sensor, and Y_j, j = 1, . . . , m be observations by another sensor. We assume X's and Y's are normally and independently distributed with true positions of the targets as their means and known standard deviations σ_1 and σ_2 respectively. These observations may indicate that there are at least the maximum of m and n targets, but not more than m+n targets. Once a pair (X_i, Y_j) is isolated for fusion and determined as a single target detected by both sensors, then an efficient estimator is given by a weighted mean

$$(\sigma_2^2 X_i + \sigma_1^2 Y_j)/(\sigma_1^2 + \sigma_2^2). \qquad (2.1)$$

We present the following criteria for possible fusion of a pair (X_i, Y_j):

1. Compatibility X_i and Y_j are said to be compatible or matchable if

$$|X_i - Y_j| \leq z_\alpha \sqrt{\sigma_1^2 + \sigma_2^2} \qquad (2.2)$$

with preassigned probability $1 - \alpha$, where z_α is the value of the standardized normal variate corresponding to the tail probability $\alpha/2$. Otherwise, they are said to be incompatible and represent 2 targets.

2. Monogamy Given Y_j there is at most only one X_i that can be fused with Y_j.

3. Maximum number of fusions The number of fusions of X's and Y's subject to the compatibility and monogamy criteria is maximized.

4. Least square error (LS) The optimality condition for selecting one of several X's compatible with a given

*Research supported by US MICOM DAAH01-82-D-A008 while at the University of Alabama in Huntsville.

Y_j and one of several Y's compatible with a given X_i is that

$$\sum (X_i - Y_j)^2 \qquad (2.3)$$

where the summation is taken over all possible sets of pairs (X_i, Y_j) that satisfy the preceding criteria. This is simply a Euclidean distance. If we use a different metric for least square error, we get an entirely different set of fusions.

The error probability α can be so chosen that the overall error probability of misclassifying some single targets as multiple ones being bounded by a multiple of α meets a certain threshold. The monogamy criterion is to rule out the possibility of a sensor seeing an object as two images. This may eliminate studying a possible malfunction of a sensor seeing double vision from our present analysis. The criterion of making maximum number of fusions reduces α of viewing a single target as multiple targets based on two sensors. Finally the optimality condition is the same as the well-known least square error principle in statistics.

3. Discussion

Without loss of generality, X's and Y's are sorted in a nondecreasing order. They are all plotted on a real line and for each Y_j associate all X's that satisfy (2.2). For illustration, let Y_1be compatible with X_1, X_2 and X_3, Y_2 with X_2 and X_3, Y_3 with X_3, X_4 and X_5, and so on. This information can be presented in a n x m table like Figure 3.1 given below. Compatibility of (X_i, Y_j) is noted in Figure 3.1 by a symbol such as *, +, •, x, and □.

Observe X_7 is not compatible with any Y, i. e., X_7 is observed only by the first sensor; similarly, Y_{10} is observed only by the second sensor. X_6 is the only one compatible with Y_5 so we can fuse them into a single target. On the other hand, X_5 is compatible with Y_3 and Y_4. Since Y_3 has two other compatible X's, X_5 should be fused with Y_4 in spite of the fact that X_5 may be closer to Y_3 than Y_4; otherwise, Y_4 will have no matches thus violating the third criterion of achieving maximum number of fusions. Now that X_5 is fused with Y_4, X_4 is automatically fused with Y_3. In case of X_8 it has 3 possible Y's for possible matching. Obviously the Y closest to X_8 will be chosen for fusion.

After we have matched the X's and Y's in the discussion above, do recursively this obvious type of matching and dropping the matched or fused pairs as well as those X's and Y's that cannot be matched, until we arrive at contiguous figurations in which each Y has at least two X's compatible and certain geometrical properties will hold good.

FIGURE 3.1

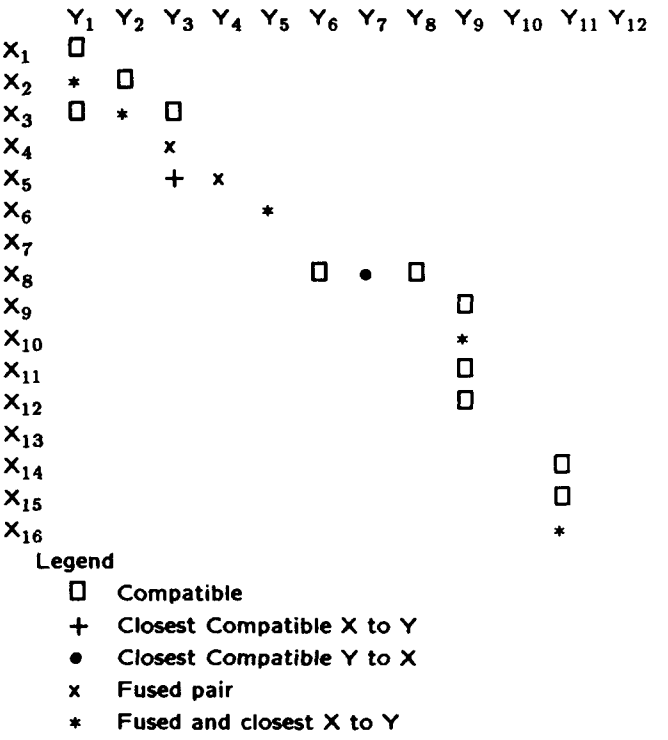


Figure 3.2 given below is a good example of table pruned for obvious matches and nonmatches. Let us consider the portion of the table with 7 rows and 4 columns giving rise to the first contiguous block. Since there are 3 compatible X's for Y_1, 2 for Y_2, 3 for Y_3 and 3 for Y_4, we have 3x2x3x3 = 54 possible sets of 4 pairs. However, by the monogamy criterion this number is reduced to 19 which can be enumerated as

2345, 2346, 2347, 2356, 2357, 2365, 2367, 2456, 2457, 2467, 2465, 3456, 3465, 3457, 3467, 4356, 4357, 4365, 4367

where, e. g., 2345 denotes the set of pairs (X_2, Y_1), (X_3, Y_2), (X_4, Y_3) and (X_5, Y_4). It can be shown that the LS criterion implies that $X_i \leq X_j \leq X_k \leq X_l$ since their counterparts satisfy $Y_1 \leq Y_2 \leq Y_3 \leq Y_4$. We call this principle *seniority protocal*, that is, if Y's are sorted in increasing order, then their corresponding X's are also sorted in increasing order. By this principle we can eliminate 7 of 19 sets, e. g., 4365. Among the 12 sets, only two sets contain a maximum number of

X's that are nearest to their Y's so that the least square error will be smaller. They are 2346 and 3456 for which we need to compute the LS error and choose the one with a smaller value.

An easy way of arriving at these two sets is to begin with 3446 composed of the closest X's to Y's. To avoid duplication of the second and third digits, there are two possible choices, namely, either to decrease the second digit by one or increase the third one by one. When the second is decreased by one, which is the same as the first digit, then we decrease the first by one thus obtaining 2346. Similarly, by increasing the third digit by one in 3446 we obtain the second possible set 3456.

In the second contiguous block of Figure 3.2 consisting of rows from 8 to 16 and columns from 5 to 7, we may arrive at 3 possible sets for verification of the LS criterion respectively. This is one of the worst cases one may encounter for verification of the LS criterion. It can be established that the number of possible sets of pairs to be checked for the LS criterion is at most the minimum of number of X's and that of Y's in a contiguous block.

Using these observations we formulate a general algorithm in the next section.

FIGURE 3.2

	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}
X_1												
X_2	□											
X_3	+	□										
X_4	□	+	+									
X_5			□	□								
X_6			□	+								
X_7				□								
X_8					□	□	□					
X_9					□	□	□					
X_{10}					□	□	□					
X_{11}					□	□	□					
X_{12}					+	+	+					
X_{13}					□	□	□					
X_{14}					□	□	□					
X_{15}					□	□	□					
X_{16}					□	□	□					

Legend

□ Compatible

\+ Closest Compatible X to Y

4. Algorithm

The algorithm we develop here will separate X's and Y's into three sets A, B and C, where A is the set of matchable pairs (X_i, Y_j), while B consists of unmatched X's and C of unmatched Y's. Then the number of distinct targets is the sum of the sizes of these sets. We present the algorithm in a language-free step-by-step format:

1. Screen the data and categorize them into two arrays X(1: n) and Y(1: m).
2. Sort X(1: n) and Y(1: m) in a nondecreasing order.
3. Associate with each Y_j, f_j, l_j and c_j, the subscript of the first, last and closest compatible X respectively. Form a linked list of distances of compatible X's from Y_j.
4. Do recursively until each Y has at least 2 compatible X's:
   i. Remove Y's with no compatible X's and place them in set C.
   ii. If a Y has only one compatible X_i then consider the set of all Y's which are compatible only with X_i. Match the closest Y with X_i. Remove and place this pair in set A. Remove all other unmatched Y's and and place them in set C.
5. Partition the remaining X's and Y's into subgroups such that each Y in a subgroup has at least two compatible X's and each X has at least two compatible Y's within the subgroup. Moreover, $f_j \leq l_{j-1}$ is true for all j with Y_j in the subgroup.
6. Without loss of generality, we assume a subgroup consists of X(1: p) and Y(1: q). Consider the array (1: q) of subscripts of closest X's to Y's. If elements of this array are distinct, then we have found the set of matched pairs $(X_{c(j)}, Y_j)$. Otherwise, from c(1: q) we form several arrays such that the resulting ones contain as many elements of c(1: q) as possible and other elements are closer to those of corresponding positions in c(1: q). The elements of these arrays are such that they are nondecreasing and conform to the monogamy and maximum number of matches criteria.
7. Verify for each array obtained in step 6 the LS criterion and select the one with the least value. Remove the X's and Y's corresponding to the set of the matched pairs and place them in set A. Remove unmatched X's and place them in set B

while unmatched Y's are removed and place in set C.

8. Repeat steps 6 and 7 for all subgroups.

In implementing this algorithm one can combine several steps and do them in a single step. The algorithm employs essentially the backtracking strategy with criteria as bounding functions.

The space complexity is not of concern since n will be less than 50 and m less than 20. The time complexity includes times for screening, sorting and computation of the LS criterion. It is estimated as of order O(mn).

5. Extension

There are two aspects to be considered for the algorithm to be useful:

1. Extending to the three-dimensional observations;
2. Fusing the observations from more than 2 sensors.

Our discussion has been centered on real-valued X's and Y's. With some modifications we can extend the algorithm to the case when the observations are real vectors.

As an extension to vector observations, an efficient estimator for the target position vector when a fusion of two observations made by two sensors is made, involves *a priori* knowledge of the covariance matrix of observations for each sensor. One possible and simple suggestion is to use (2.1) for each component of the target position. There are several estimators which are based on different estimation criteria.

The compatibility criterion given in (2.2) can be defined as $(1-\alpha)$ probability ellipsoid region centered at the origin with the axes determined by the sum of the covariance matrices of the two sensors. The least square error defined in (2.3) can be replaced with any other metric.

In order to use the algorithm the metric defined in (2.3) should lead us to define an ordering on X's and Y's for sorting and for maintaining the seniority protocol. This will be a study for the future.

From fusion of data obtained from two sensors we go to fusion of data from three sensors. This may be done in two stages — fusing the data for the first two sensors and then fusing them with the third sensor. However, this will introduce a large error probability. This requires a further research in controlling error probabilities.

No algorithm is applicable unless it is implemented on a processor and real time for computations performed should be studied. The question of practicality in a simulated battle scenario is to be explored, which will, in turn, force us to refine our algorithm.

Since the targets are moving and the sensors are continuously polling, the fusion of the data can be dynamically verified and updated. This would be an immediate line of extension of this study.

Finally, the results obtained in the study have applications in areas such as medicine where multiple sensors may be used for monitoring patients' conditions. How the brain of an animal processes the information from the data received through several senses will be another application. Fussy logic which seems to play a larger role in electronics and its applications to photography can be interfaced with the multiple sensors.

The author acknowledges generously the US Army Missle Command and Mr. Richard Jones for his patient and helpful briefings and suggestions.

REFERENCES

Horowitz, E. and Sahini, S. (1978), *Fundamentals of Computer Algorithms.* Computer Sciences Press, Inc.

Johnson, R. A. and Wichern, D. W. (1988), *Applied Multivariate Statistical Analysis.* Second edition, Prentice Hall, Englewood Cliffs, NJ

Kendall, M. G. and Stuart, A. (1979), *Advanced Methods of Statistics.* Volume 2, Fourth edition, MacMillan Publishing Co., New York.

Control Charts Under Linear Trend

F. F. Gan
National University of Singapore
Department of Mathematics
10 Kent Ridge Crescent
Singapore 0511

Abstract

The performance of control charts is usually evaluated by assuming a step change in the process mean. However, it is more appropriate to evaluate the performance of control charts by assuming a drift in the mean for processes where a gradual drift models the shift in the mean more accurately. Three major methods for computing the average run length (ARL) of control charts assuming a step change in the mean are reviewed. Generalizations of these methods for computing the ARL of control charts assuming a drift in the mean are then examined.

KEY WORDS: Average run length; Cumulative sum; Exponentially weighted moving average; Integral equation; Linear drift; Markov chain; Normal distribution; Statistical process control.

1 Introduction

Let $\bar{x}_1, \bar{x}_2, \ldots$, be a sequence of independent and identically distributed measurements of quality from a manufacturing process and $f_\mu(x)$ be the probability density function of $\bar{x}_1$ where μ is the mean of $\bar{x}_1$. Without loss of generality, assume that the in-control process mean to be zero and the standard deviation of $\bar{x}_1$ to be one.

An upper-sided cumulative sum (CUSUM) chart is obtained by plotting

$$S_t = \max\{0, S_{t-1} + \bar{x}_t - k\},$$

against the sample number t for $t = 1, 2, \ldots$ where k is a positive chart parameter and $S_0 = u$, $0 \leq u < h$. An out-of-control signal is issued at the first t for which $S_t \geq h$. A lower-sided CUSUM chart is obtained by plotting

$$T_t = \min\{0, T_{t-1} + \bar{x}_t + k\},$$

against the number t for $t = 1, 2, \ldots$ where $T_0 = u$, $-h < u \leq 0$. An out-of control signal is issued at the first t for which $T_t \leq -h$. A two-sided CUSUM chart is obtained by running the lower-sided and upper-sided CUSUM charts simultaneously.

An exponentially weighted moving average (EWMA) chart is obtained by plotting

$$Q_t = (1 - \lambda)Q_{t-1} + \lambda\bar{x}_t,$$

against $t = 1, 2, \ldots$, where λ is a smoothing constant such that $0 < \lambda \leq 1$ and $Q_0 = u, -h < u < h$. An out-of-control signal is issued by an EWMA chart when $Q_t < -h$ or $Q_t > h$.

The run length of a control chart is defined to be the sample number when an out-of-control signal is first issued. The ARL of a control chart which is the expectation of run length is often used as a measure of performance of a control chart. Three major methods for computing the ARL of control charts assuming a step change in the mean are reviewed in Section 2. Generalizations of these methods for computing the ARL of control charts assuming a drift in the mean are then examined in Section 3.

2 Step Changes

A common method for computing the ARL of a control chart assuming a step change in the mean is through the use of integral equation. The ARL function of a CUSUM chart was first derived by Page (1954) as an integral equation

$$L(u) = 1 + L(0)\Pr(\bar{X} \leq k - u) + \int_0^h L(x) f_\mu(x + k - u)\,dx,$$

where $L(u)$ denotes the ARL of a CUSUM chart given that $S_0 = u$.

Using an argument similar to Page (1954), the ARL of an EWMA chart was expressed by Crowder (1987) as

an integral equation

$$L(u) = 1 + \frac{1}{\lambda}\int_{-h}^{h} L(x) f_\mu\left(\frac{x-(1-\lambda)u}{\lambda}\right)dx,$$

where $L(u)$ here denotes the ARL function of a two-sided EWMA chart given that $Q_0 = u$.

The CUSUM integral equation can be approximated numerically by replacing the equation with a system of linear algebraic equations using a Gaussian quadrature and solving the system of linear equations. A comprehensive discussion of methods used to obtain approximate solutions to integral equation can be found in Baker (1977).

Let $w_1, w_2, \ldots, w_n$ and $u_1, u_2, \ldots, u_n$ be the n-point weights and abscissas of a Gaussian quadrature such that

$$\int_0^h g(x)dx \approx \sum_{i=1}^{n} w_i g(u_i).$$

Using the Gaussian quadrature, the CUSUM integral equation can then be replaced by a system of linear equations in $n+1$ unknowns $L(u_1), \ldots, L(u_{n+1})$,

$$L(u_j) \approx 1 + L(0)\Pr(\bar{X} \le k - u_j) + \sum_{i=1}^{n} w_i L(u_i) f_\mu(u_i + k - u_j)$$

where $j = 1, 2, \ldots, n, n+1$ and $u_{n+1} = 0 < u_1 < u_2 < \ldots < u_n < h$. The ARL function $L(u)$, $0 \le u < h$ can then be approximated as

$$L(u) \approx 1 + L(u_{n+1})\Pr(\bar{X} \le k - u) + \sum_{i=1}^{n} w_i L(u_i) f_\mu(u_i + k - u).$$

The EWMA integral equation can be approximated in a similar manner.

The second method for computing the ARL of a control chart is to use standard results from the Markov chain theory. This method is first proposed by Brook and Evans (1972). Consider an EWMA chart with chart limits $(-h, h)$. This interval is partitioned into n subintervals and let $\mathbf{R}$ be the matrix containing the one-step transition probability for the transient states. The ARL vector $\bar{\mathbf{u}} = (u_1, u_2, \ldots, u_n)^T$ is then given by

$$\bar{\mathbf{u}} = (\mathbf{I} - \mathbf{R})^{-1}\bar{\mathbf{1}}$$

where $\mathbf{I}$ is the $n \times n$ identical matrix and $\bar{\mathbf{1}}$ is an $n \times 1$ vector of 1's. The Markov chain method may also be used in a similar manner to compute the ARL of a CUSUM chart.

The third method is Monte Carlo method which is easily programmed but highly inefficient. These three methods allow the ARL of a control chart to be evaluated for any particular value of μ and hence allow the performance of control charts to be evaluated assuming a step change in the mean.

3 Linear Drift

The Monte Carlo method can be generalized easily to handle the case when the process mean is drifted and will not be discussed any further in this paper.

The performance of CUSUM charts under linear drifts in the process mean was investigated by Bissell (1984). It is assumed that the first sample is taken when the mean is in control and the mean is drifted gradually at a rate of $\Delta\sigma_{\bar{X}}$ per sampling interval where $\sigma_{\bar{X}}$ is the standard deviation of sample mean and Δ is a positive constant. Based on a modification of the Markov chain method developed by Brook and Evans (1972), Bissell computed the ARL of a CUSUM chart under a linear drift. A nonhomogeneous Markov chain is obtained for the linear drift case. However, Bissell noted in a corrigendum that the ARL computed using his Markov chain method is not accurate, possibly due to rounding errors. Based on simulation results, Bissell showed that the ARL computed using his Markov chain method is at least two times larger than the actual ARL for a CUSUM chart with $h = 5.0$, $k = 0.5$ and a drift coefficient $\Delta = 0.005$. The accuracy of the Markov chain method has been greatly improved due to a refinement of Asbagh (1985).

Gan (1991a, 1991b) generalized the integral equation method to handle the case when the mean is drifted. Let the mean be $\mu_0, \mu_1, \ldots, \mu_{m-1}, \mu_m, \mu_m, \ldots$ when random samples of products are taken from a production process. Note that μ_m can be set to an arbitrary large number to approximate a linear wear process. Let the sample mean

be independently distributed with probability density function $f_{\mu_j}(x)$. Note that μ_m is the stabilized process mean. Suppose that $L_j(u, \mu_j)$, $j = 0, 1, 2, ..., m$ is the ARL of a two-sided EWMA chart given that $Q_0 = u$ and random samples of products are taken when the mean is at $\mu_j, \mu_{j+1}, ..., \mu_m, \mu_m, ...$. Gan(1991a) showed that

$$L_j(u, \mu_j) = 1 + \frac{1}{\lambda}\int_{-h}^{h} L_{j+1}(x, \mu_{j+1}) f_{\mu_j}\left(\frac{x-(1-\lambda)u}{\lambda}\right)dx,$$

for $j = 0, 1, 2, ..., m-1$ and

$$L_m(u, \mu_m) = 1 + \frac{1}{\lambda}\int_{-h}^{h} L_m(x, \mu_m) f_{\mu_m}\left(\frac{x-(1-\lambda)u}{\lambda}\right)dx.$$

The last equation is an integral equation and the ARL function $L_m(u, \mu_m)$ can be approximated numerically by replacing the equation with a system of linear algebraic equations using a Gaussian quadrature and solving the system of equations. Once the ARL function $L_m(u, \mu_m)$ is found, simple substitution method may be employed to compute $L_j(u, \mu_j)$, $j = m-1, m-2, ..., 2, 1, 0$ recursively. Note that $L_0(0, \mu_0)$ is the ARL of an EWMA chart with $Q_0 = 0$, assuming that the first sample is taken when the mean is at μ_0 and subsequent samples are taken when the mean is at μ_j for $j = 1, 2, ..., m$. Similar ARL equations for a CUSUM chart assuming a drift in the mean are obtained by Gan (1991b).

4 Conclusions

Three major methods for computing the ARL of CUSUM and EWMA control charts under step shifts and linear drifts are reviewed in this paper. Both the Markov chain and integral equation methods yield accurate ARL values of control charts under linear drifts. The methods discussed in this paper may also be used to study the run length properties of control charts with drift that is not linear in nature.

References

Asbagh, Nooshin A. (1985). "Performance of CUSUM and Combined Shewhart-CUSUM Charts Under Linear Trend", unpublished M.Sc. Thesis, University of Southwestern Louisiana, Department of Statistics.

Baker, C. T. H. (1977). *The Numerical Treatment of Integral Equations*. Clarendon Press, Oxford, England.

Bissell, A. F. (1984). "The Performance of Control Charts and Cusums Under Linear Trend," *Applied Statistics*, 33, 145–151 (Corrigendum, *Applied Statistics* 35).

Brook, D., and Evans, D. A. (1972). "An Approach to the Probability Distribution of Cusum Run Length," *Biometrika*, 59, 539–549.

Crowder, S. V. (1987). "A Simple Method for Studying Run-Length Distributions of Exponentially Weighted Moving Average Charts," *Technometrics*, 29, 401–407.

Gan, F. F. (1991a). "EWMA Control Chart Under Linear Drift," *Journal of Statistical Computation and Simulation* (to appear).

Gan, F. F. (1991b). "CUSUM Control Chart Under Linear Drift," *The Statistician* (to appear).

Page, E. S. (1954). "Continuous Inspection Schemes," *Biometrika*, 41, 100–115.

Computation and Interpretation of Deformations for Landmark Data in Morphometrics and Environmetrics

Paul D. Sampson, Steven Lewis and Peter Guttorp*
Department of Statistics
University of Washington
Seattle, Washington 98195

Fred L. Bookstein+
Center for Human Growth and Development
University of Michigan
Ann Arbor, Michigan 48109

Catherine B. Hurley
Department of Statistics
George Washington University
Washington, D.C. 20052

1. Introduction

Analytical problems in a number of scientific disciplines concern comparisons of two sets of distances among labelled points. These two sets of distances (or metrics) correspond to different Euclidean representations of the points. The comparison between these two sets of distances, or between the two configurations of points in space, is often expressed best in terms of a deformation that maps the set of labelled points in one representation into the corresponding set in the second representation. In this paper we discuss the computation and interpretation of these deformations for two particular fields of application and the visualization of these deformations using the graphical technique of biorthogonal grids (Bookstein 1978).

Perhaps the primary examples of comparisons of sets of points through deformations is in the field of cartography. Distortions induced by representing the surface of the earth with (planar) maps are studied in terms of the properties of various map projections (Richardus and Adler 1972). In fact the basis of the method of biorthogonal grids presented here was established a century ago by Tissot (1881) for just such problems. Tissot's theorem shows that the image of any small (infinitesimal) circle under a continuous transformation is an ellipse—known in cartography as *Tissot's indicatrix*. The axes of the ellipse represent the local principal (maximum and minimum) strains of the transformation and the ratio of the area of the ellipse to the area of the circle represents the proportionate change (distortion) in surface area. The theorem further states that at any point in the domain image (e.g. the surface of the earth) there is a unique pair of (infinitesimal) lines or directions at 90° that intersect also at 90° in the response image (e.g., planar map), unless the transformation is conformal. These lines are the axes of Tissot's indicatrix. The distinctive feature of these cartographic applications is that projections relating the surface of the earth to the map are known analytically. Tobler (1978) suggests other problems in the study of geographic patterns where the mappings must be computed from data.

Problems in biology, or more specifically morphometrics—the measurement of biologic shapes, their variation and change—motivated the development of the algorithms presented here. In 1917, D'Arcy Thompson introduced the idea of using mathematical deformations for describing or reifying the theoretical construct of biological homology. Two biological forms were to be compared in terms of a deformation of one form into the other. Images to be compared might be of two distinct biologic species related in evolutionary terms, or images of the same biologic specimen observed at two different ages in a longitudinal study of growth. For visualization Thompson used the method of transformation grids in which a square or regular grid superimposed on the image of one biologic form is tranformed into an irregular grid over a second form.

Many investigators attempted to place Thompson's seminal idea on a precise mathematical foundation suitable for computer analysis and *measurement* of shape change. This was not achieved until Bookstein's (1978) introduction of the method of biorthogonal grids. (Closely related methods were suggested independently by Tobler (1978) at about the same time.) Bookstein's approach focusses on labelled points called *landmarks* of anatomical or evolutionary significance in the biologic images to be compared. The purpose of Bookstein's analysis was the depiction and

* Research supported by a contract with the Electric Power Research Institute, Palo Alto, CA.
+ Research supported by NIH grants NS-26529 and GM-37251.

measurement of shape change interpolated smoothly from the sets of corresponding labelled landmarks in the two images. The method involves (a) interpolation of a correspondence between n pairs of points in R^2 into a differentiable mapping defined everywhere in the plane, and (b) the drawing (and labelling) of integral curves of the infinitesimal perpendicular lines guaranteed to exist (for differentiable transformations) according to Tissot's Theorem. A formal definition for these curves is as follows.

Definition. Through almost every point of a differentiable transformation pass just two differentials which are at 90° both before and after transformation. The integral curves of these differentials form a grid whose intersections are at 90° in both images. These are called the *biorthogonal grids* of the transformation.

Another field to which the method of biorthogonal grids has been applied is the statistical analysis of spatial data obtained in routine monitoring of environmental processes. Examples include spatial analyses of mesoscale variation in solar radiation (Sampson and Guttorp 1991), wind speed (Guttorp and Sampson 1989), rainfall, and acid deposition (Guttorp et al 1991). Sampson and Guttorp suggested that the spatial covariance structure of environmental monitoring data could be represented and estimated in terms of a function mapping the geographic locations of a set of monitoring stations (with coordinates generally being planar coordinates from a map projection whose effects are being ignored) into a second synthetic set of planar coordinates computed to encode the spatial covariance structure: distances between the stations encode observed spatial covariances so that greater covariances are represented by smaller distances. In this application, the biorthogonal grids reflect spatially varying anisotropic spatial covariance structure—what may be called a "moving principal components analysis of the spatial covariance structure."

While the algorithms and graphics are the same for each of the applications cited above, there are important, albeit subtle differences of interpretation. For the analysis of map projections the analytically specified mappings apply to all locations in the domain image, and thus Tissot's indicatrix can be computed and interpreted everywhere. In the environmental monitoring problem we compute a smooth mapping from data at a finite set of points considered as a spatial sample of an underlying process. Thus the biorthogonal grids computed by interpolation may be interpreted as *estimates* of a phenomenon (spatial covariance pictured as deformation) which is defined (and theoretically observable) everywhere. That is, pairs of images can conceivably be generated for monitoring stations located anywhere in the geographic region of interest.

However, in the morphometric applications correspondences between images cannot generally be established except at a finite (or one-dimensional) set of points. The biorthogonal grids provide an illustration referring only to the landmarks available for analysis. One cannot argue that they represent (estimates of) real deformation as correspondences (homologies) cannot be defined everywhere. (See Bookstein 1991.)

In Section 2 we explain the interpretation of biorthogonal grids for a pair of simple examples based on a hypothetical square configuration of four landmarks. We utilize thin-plate splines to represent the deformations we compute from corresponding landmarks in pairs of images. Section 3 explains the rationale for the choice of thin-plate splines and reviews their algebra. Section 4 details the algorithms for drawing biorthogonal grids for a specified mapping. The last section presents a pair of (very different) real applications.

2. Interpretation of biorthogonal grids

We begin with simple linear or affine transformations, $u = f(x) = Ax$, where A is a 2×2 matrix and x and u represent coordinate vectors in two images to be compared, respectively. Linear mappings are characterized by a single pair of *principal axes* given by the eigenvectors of A^TA (or left singular vectors of A). The direction or axis corresponding to the largest eigenvalue is the direction in which the plane is (relatively) most stretched. The ratios of distance in the second image to distance in the first for pairs of points x_i and x_j aligned with the principal axes (e.g. $|u_i - u_j| / |x_i - x_j|$) are called the principal strains. These are the square roots of the eigenvalues of A^TA (or the singular values of A).

Figure 1 depicts the effect of a linear transformation on a starting configuration of four points arranged in a square. The resulting figure is a parallelogram. The families of perpendicular lines indicate the directions of the principal axes. They correspond between the two images and are the biorthogonal grids for the linear transformation. The figure refers to the principal strains as gradients ("grad"). Those for the transformation from parallelogram to square are the inverses of those for the transformation from square to parallelogram. In this case the two principal strains are 0.856 (coded by dotted lines in the left panel of Figure 1) and 0.506 (coded by dashed lines).

A simple nonlinear mapping of a square into an arbitrary quadrilateral is depicted in Figure 2 (approximately replicating Figure VI-6 in Bookstein 1978). We will write such a nonlinear mapping $f: R^2 \to R^2$ as

$$\begin{bmatrix} u \\ v \end{bmatrix} = f\begin{bmatrix} x \\ y \end{bmatrix}.$$

In the neighborhood of any point (x,y) we can compute a local linear approximation

$$f(x,y) = A_{x,y}\begin{bmatrix} x \\ y \end{bmatrix} + \cdots.$$

where the matrix $A_{x,y}$ is the (affine) derivative matrix evaluated at the point (x,y),

$$A_{x,y} = \begin{bmatrix} \partial u/\partial x & \partial u/\partial y \\ \partial v/\partial x & \partial v/\partial y \end{bmatrix}_{x,y}.$$

The left singular vectors of $A_{x,y}$ indicate the *local* principal axes of the nonlinear transformation. These are the differentials referred to in the definition of biorthogonal grids given above. In Figure 2 we can identify these directions along the curves drawn. For example, in the neighborhood of point 4, the first local principal axis (with a principal strain between 0.5 and 0.75) points in a direction at approximately 30° below the horizontal. In the vicinity of point 2, the first principal axis (with a strain greater than 0.75) has rotated to an angle of approximately 45° from the horizontal. Note that the biorthogonal grids—a sampling of integral curves of these local principal directions—depict the spatial variation in the principal strains (derivative) of a nonlinear mapping.

3. Algebra of thin-plate splines for plane mappings

Our problem is to determine a smooth mapping $f: R^2 \to R^2$ that interpolates a correspondence between labelled points (x_i,y_i) and (u_i,v_i), respectively, $i=1,2,\ldots,n$, in two images. That is, f must satisfy

$$(u_i,v_i) = f(x_i,y_i) = (u(x_i,y_i), v(x_i,y_i)). \qquad (1)$$

As noted above, biorthogonal grids depict the spatial variation in the derivative of a mapping f. Therefore, as a measure of smoothness it is natural to consider the family of thin-plate splines which minimize variation in the derivatives as expresssed in the following quantity.

$$I_f = \int\int_{R^2} \left[\left(\frac{\partial^2 u}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 u}{\partial x\,\partial y}\right)^2 + \left(\frac{\partial^2 u}{\partial y^2}\right) \right] + \left[\left(\frac{\partial^2 v}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 v}{\partial x\,\partial y}\right)^2 + \left(\frac{\partial^2 v}{\partial y^2}\right) \right] dx\,dy.$$

The terms involving u and v are, separately, equations for the bending energy of an idealized thin metal plate displaced to pass through "vertical" coordinates u_i or v_i respectively, at the points (x_i,y_i). Note that linear functions of x and y have zero bending energy; i.e., they are perfectly smooth.

We review here the algebra of thin-plate splines. Similar presentations appear in Bookstein (1989, 1991). Consider first the problem for just one of the response coordinates, $v(x,y)$. Let $P_i = (x_i,y_i)^T$ and denote the distance between points i and j by $r_{ij} = |P_i - P_j|$. Define the function $U(|r|) = r^2 \log r^2$. Then the solution $v(x,y)$ minimizing (the v components of) I_f subject to the interpolation constraints (1) is

$$v(x,y) = \sum_{i=1}^{n} w_{vi}\, U(|P_i - (x,y)^T|) + a_{v0} + a_{vx}x + a_{vy}y,$$

where the coefficient vector defined as $\theta = (w_{v1}, \cdots, w_{vn}, a_{v0}, a_{vx}, a_{vy})$ satisfies a linear system

$$Y = L\,\theta.$$

$Y = (v_1, v_2, \cdots, v_n, 0, 0, 0)^T$ and the $(n+3)\times(n+3)$ matrix L is

$$L = \begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix},$$

where

$$K = \begin{bmatrix} 0 & U(r_{12}) & \cdots & U(r_{1n}) \\ U(r_{21}) & 0 & \cdots & U(r_{2n}) \\ \cdots & U(r_{32}) & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ U(r_{n1}) & U(r_{n2}) & \cdots & 0 \end{bmatrix},$$

and

$$P = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \cdots & \cdots & \cdots \\ 1 & x_n & y_n \end{bmatrix}.$$

For further discussion of the role and interpretation of the function U, see Bookstein (1991, Ch. 8).

The value of the (minimal) bending energy is proportional to a quadratic form in the coefficients $W = (w_{v1}, \cdots, w_{vn})^T$, or in the observations $V = (v_1, v_2, \cdots, v_n)^T$,

$$I_f \propto W^T K\, W = V^T (L_n^{-1} K L_n^{-1}) V = V^T (L_n^{-1}) V,$$

where L_n^{-1} refers to the upper left $n \times n$ submatrix of L^{-1}. Note that this quadratic form is zero if and only if $w_{v1} = w_{v2} = \cdots = 0$, which means that the fitted spline is linear, $v(x,y) = a_{v0} + a_{vx}x + a_{vy}y$.

For the two-dimensional response problem of interest here we define V as an $n \times 2$ matrix,

$$V = \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \\ \cdots & \cdots \\ u_n & v_n \end{bmatrix}$$

and

$$I_f \propto tr(V^T(L_n^{-1})V).$$

From this expression it is easy to see that the quantity minimized is invariant under arbitrary translation and rotation of the coordinates (u,v). In fact, the whole procedure is invariant under translation and rotation of either set of coordinates, (x,y) and/or (u,v).

4. Computing and drawing biorthogonal grids

This section explains how we compute and draw biorthogonal grids as a visualization of a differentiable mapping $f: R^2 \rightarrow R^2$ over a specified region of the plane. For a given mapping f, e.g., a thin-plate spline, we can compute local linear approximations in terms of the affine derivative matrix $A_{x,y}$ defined above. From this matrix we can compute the differentials at (x,y) corresponding to the sets of curves of the biorthogonal grids as the left singular vectors of $A_{x,y}$. Write the direction of greatest principal strain as

$$\begin{bmatrix} dy \\ dx \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}_{x,y}.$$

In order to draw out the integral curve of the differentials emanating from the point (x,y) we solve a system of differential equations

$$\begin{bmatrix} dy/dt \\ dx/dt \end{bmatrix} = \begin{bmatrix} g_x(t,x,y) \\ g_y(t,x,y) \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}_{x,y},$$

where t denotes arclength along the curve in the direction of the local principal axis. Points along the curve are computed by running a (Runge-Kutta) differential equation solver in "one-step" mode. That is, we evaluate

$$\int_{t_0}^{t_0+t_{\max}} \begin{bmatrix} dx/dt \\ dy/dt \end{bmatrix} dt,$$

where $t_{\max}$ is the maximum step size to be taken. In one step mode this returns

$$\begin{bmatrix} x(t_0+\delta t) \\ y(t_0+\delta t) \end{bmatrix}.$$

The size of the steps δt, i.e., the spacing of points generated, depends on the curvature of the grids or the spatial rate of change of the affine derivative matrix. Associated with each step is the principal strain, the singular value of the affine derivative evaluated at the starting point. These values are used to code or label the line segments connecting the sequence of points generated along the curves.

The user must consider a number of graphical design decisions in drawing out a sample of curves for a biorthogonal grid. Our implementation in the S system (Splus, Statistical Sciences 1990) provides the user with a variety of options. A default, semi-automatic procedure starts by generating points along the curves in both the major (greatest local strain) and minor (least local strain) principal directions emanating from a user-specified starting point on the first image. Then, at points approximately equally spaced along the curve of greatest local strain just generated, the program initiates a series of points along the curves in the direction of least local strain. It similarly initiates series of points defining curves in the direction of greatest local strain from approximately equally spaced starting points along the first curve in the minor direction.

We connect the points in curves with the value of the principal strain encoded in the line type, line width, and/or line color. Color seems the most effective visual cue for recognizing the relative magnitude of the principal strains as they vary in space (although we do not demonstrate color here).

Using Splus' interactive graphics capability the user may initiate biorthogonal grid curves from any location in the first image. The user may also point at plotted curve in order to print the value of the strain at that point. A system of Splus functions for the calculations and graphics discussed here is available from the authors.

5. Applications

Our first example is drawn from a classical morphometric study of a congenital craniofacial deformity known as Apert Syndrome (see Bookstein 1991). Landmark coordinates were digitized from lateral cephalograms of 14 cases of the syndrome. Pictured in Figure 3 are the mean coordinates of eight landmarks for these 14 cases and a similar mean configuration for a sample of age and sex-matched controls ("normals"). Our interest is in describing the difference in these two configurations by viewing the mean Apert configuration as a deformation of the mean configuration of the controls.

In applications it is often useful to visualize plane mappings using both the image of the mapping of a regular grid of points (after D'Arcy Thompson) and biorthogonal grids. At the bottom of Figure 3 we depict the thin-plate spline mapping the normal mean land-

mark configuration into the Apert mean landmark configuration by the mapping of a regular grid and in the upper right corner we show the corresponding biorthogonal grid. We utilize varying line width to encode the principal strains of the mapping. The strains vary from 0.33 (in the vicinity of the points PtM and PNS) to 1.20 (between points Sel and SER). Bookstein (1989, 1991) shows how the shape change represented in Figure 3 can be usefully decomposed into features or components of varying geometric scales.

Our second example concerns an application of plane mappings in spatial statistics. A number of monitoring networks have been measuring acid deposition in rainfall over the past two decades. Problems concerning the estimation of acid deposition at unmonitored locations and the design or evaluation of monitoring networks all require information about the spatial covariance structure of the environmental process being monitored. In this application we consider (log) hydrogen ion deposition accumulated for four-weekly intervals from data measured between 1981 and 1986 at 17 monitoring sites from the UAPSP monitoring network. We denote by $Z(t,x)$ the observations at location x and time (month) t. We are interested in modeling the "spatial dispersion" $Var(Z(t,x_a)-Z(t,x_b))$ as a function of arbitrary pairs of geographic locations x_a and x_b. The sample data provides estimates of these variances for pairs of monitoring sites,

$$d_{ij}^2 = \hat{Var}(Z(t,x_i)-Z(t,x_j)).$$

Sampson and Guttorp (1991) introduced a family of models of the form

$$Var(Z(t,x_a)-Z(t,x_b)) = g(|f(x_a)-f(x_b)|),$$

where f is a nonlinear mapping of the geographic coordinates of the sampling sites and g is a monotone "variogram" function relating the d_{ij}^2 to the distances among the transformed points $|f(x_i)-f(x_j)|$.

Figure 4 shows the location of 17 monitoring stations and depicts the thin-plate spline mapping f that represents the nature of the spatial covariance structure. Coordinates of sites in the lower right image were computed in two steps as described in Sampson and Guttorp (1991). First we applied multidimensional scaling to the matrix of spatial dispersions d_{ij}^2 to generate a configuration in which pairs of sites x_i and x_j with relatively high spatial dispersion (low covariance) would be located relatively far apart. Second we computed a thin-plate *smoothing* spline to approximate these new coordinates as a smooth deformation of the geographic configuration.

The biorthogonal grid specifies (our estimates of) the geographic directions in which spatial dispersion is greatest and weakest. Different line types encode the range of values of the principal strains. Variation in these principal strains reflects nonstationarity in the spatial covariance structure, and can be understood from the perspective of the atmospheric processes underlying the monitored data. The large scale feature of the mapping in Figure 4 is a relative compression along an axis running WSW-ENE corresponding to a strong spatial covariance in that direction and relatively weak covariance in the direction at 90°.

References

Bookstein, F.L. (1978), *The Measurement of Biological Shape and Shape Change*. Lecture Notes in Biomathematics, v. 24. Berlin: Springer.

Bookstein, F.L. (1989), Principal warps: Thin-plate splines and the decomposition of deformations, *I.E.E.E. Trans. Patt. Anal. Mach. Intell.*, **11**, 567-585.

Bookstein, F.L. (1991), *Morphometric Tools for Landmark Data*, Cambridge University Press, in press.

Guttorp, P. and Sampson, P.D. (1989), Discussion of Haslett & Raftery, *Applied Statistics (JRSS C)*, **38**, 32-34.

Guttorp, P., Sampson, P.D., and Newman, K. (1991), Nonparametric estimation of spatial covariance with application to monitoring network evaluation, in *Statistics in Environmental and Earth Sciences*, eds. P. Guttorp and A. Walden, London: Griffin, in press.

Richardus, P. and Adler, R.K. (1972), *Map Projections for Geodesists, Cartographers and Geographers*, Amsterdam: North Holland.

Sampson, P.D. and Guttorp, P. (1991), Nonparametric estimation of nonstationary spatial covariance structure, SIMS Tech. Rpt. No. 148 (rev), Dept. of Statistics, Univ. of Washington, (submitted).

Statistical Sciences, Inc. (1990), *S-PLUS User's Manual*, Seattle, WA.

Thompson, D'A. W. (1961), *On Growth and Form*, Abridged edition, ed. J.T. Bonner, Cambridge Univ. Press.

Tissot, M.A. (1981), *Memoires sur les Representations des Surfaces*, Paris: Gauthier et Cie.

Tobler, W.R. (1978), The comparison of plane forms, *Geographical Analysis*,**10**, 154-162.

Figure 1. Biorthogonal grids for a linear mapping

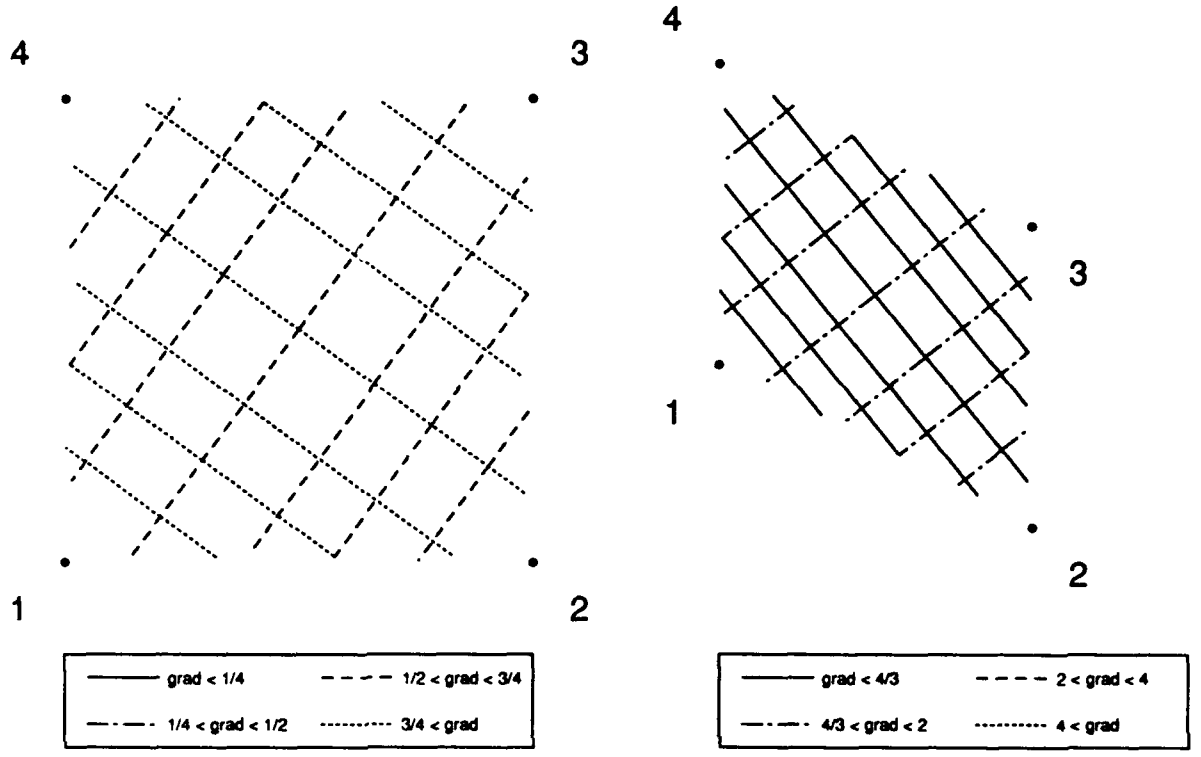


Figure 2. Biorthogonal grids for a nonlinear mapping

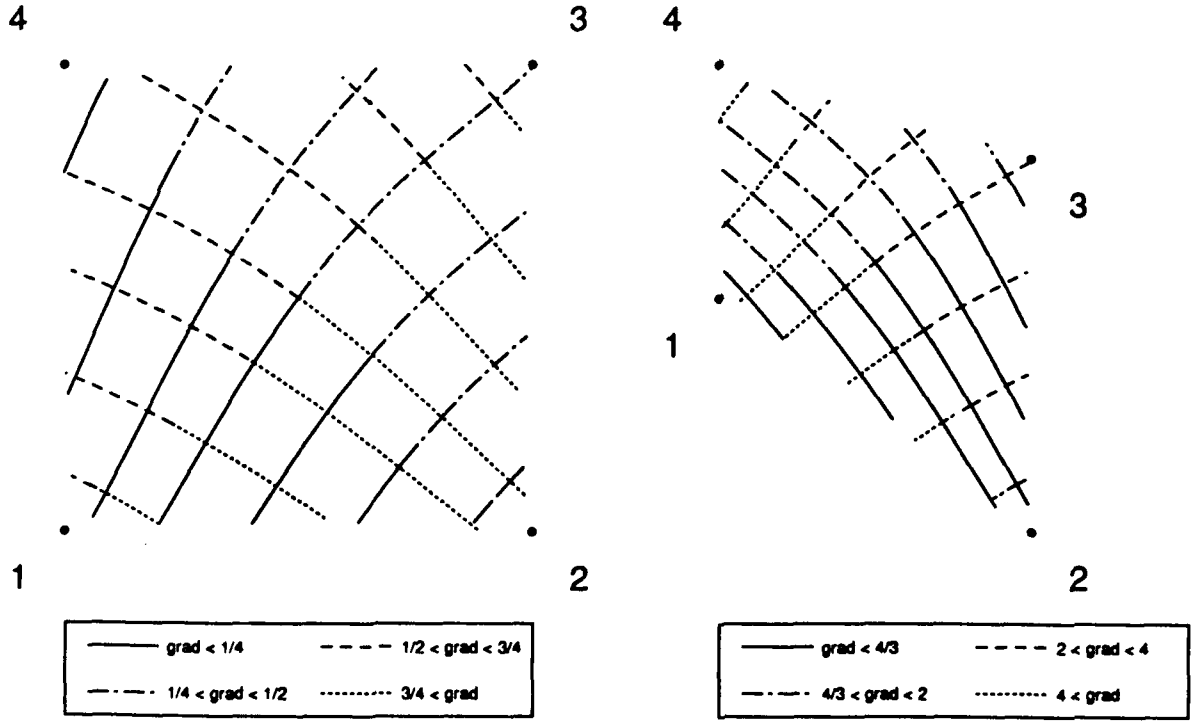
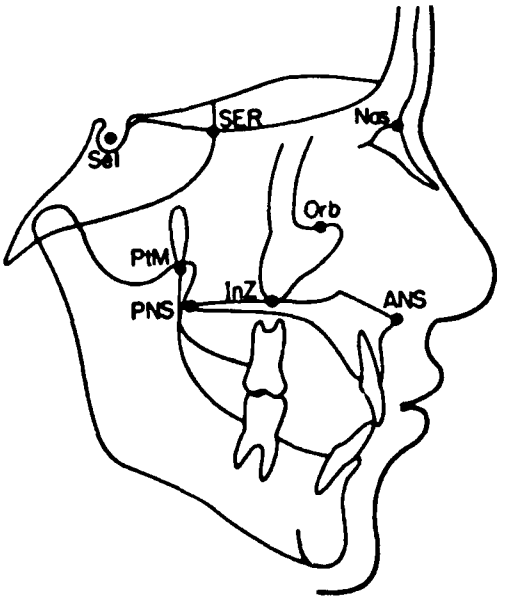
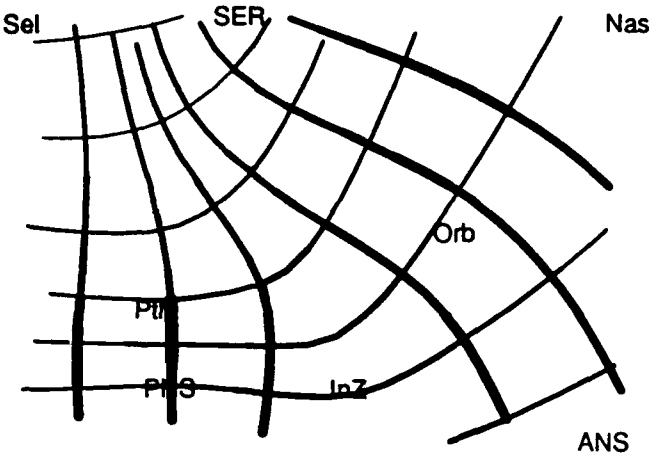


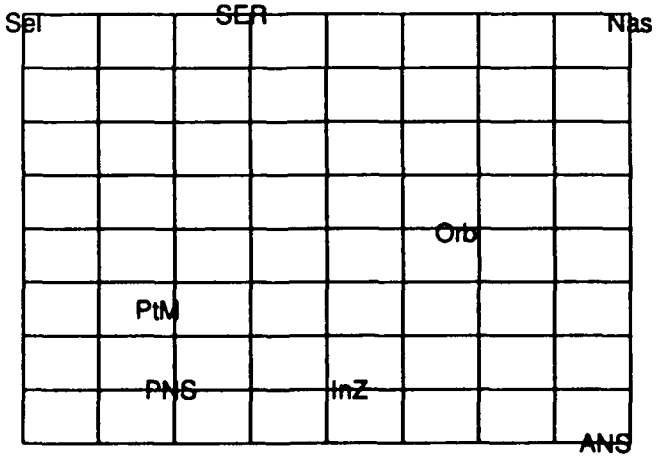
Figure 3. Thin-plate spline mapping of Normal into Apert means



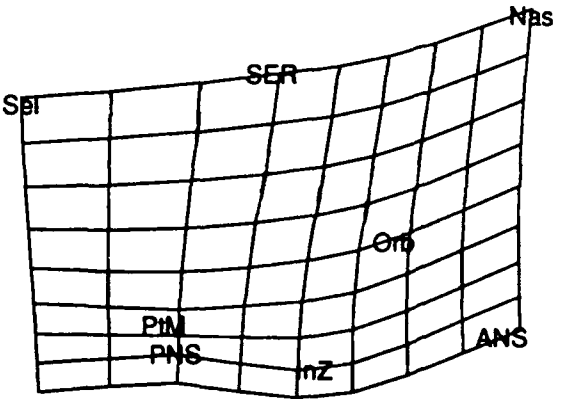
Craniofacial Landmarks



Biorthogonal Grid

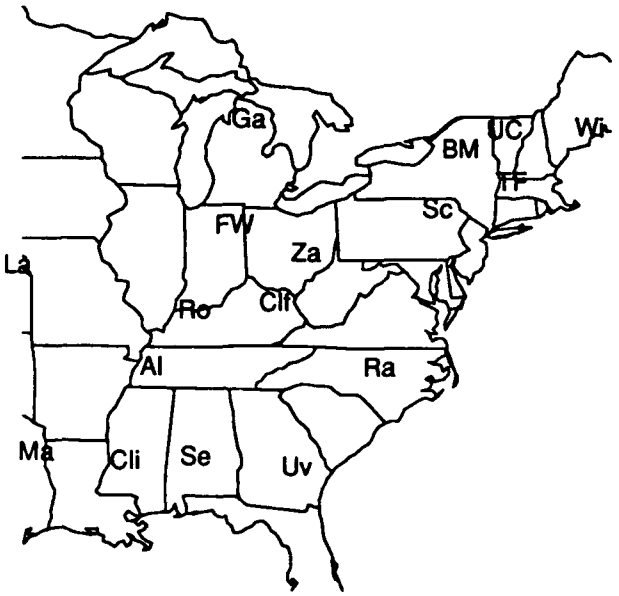


Normals Mean Landmark Configuration

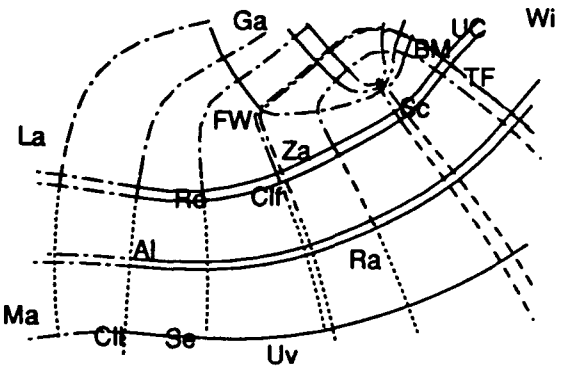


Apert Mean Landmark Configuration

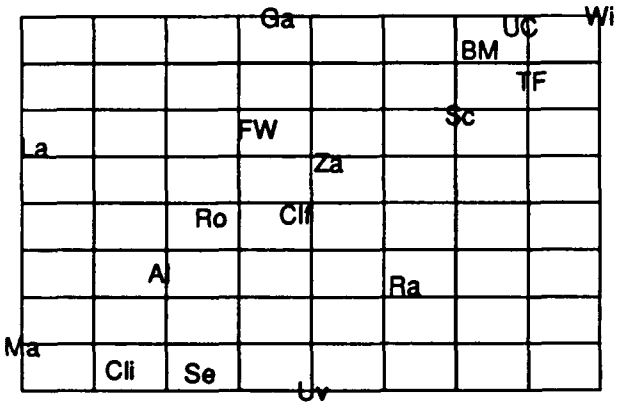
Figure 4. Thin-plate spline mapping of UAPSP monitoring stations



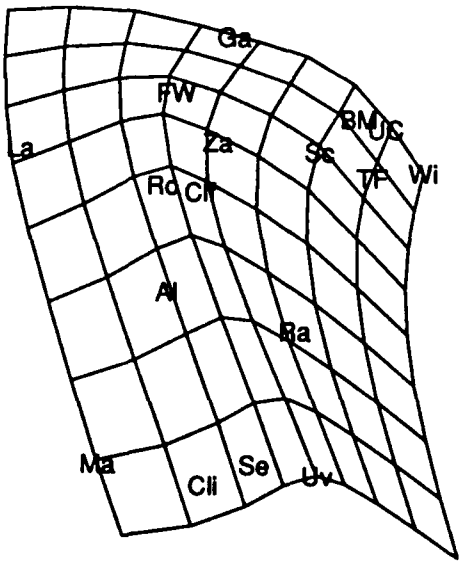
Locations of UAPSP monitoring stations



Biorthogonal Grid



Geographic Coordinates



Coordinates Representing Spatial Dispersion

Some sharpening and registration methods applied to SPECT images of pediatric brain tumors

Nicholas Lange[1]* Lorcan A. O'Tuama[3,4] Kevin M. Manbeck[2]

Robert E. Zimmerman[3,4] Donald E. McClure[2] Stuart Geman[2]

1 Division of Biology and Medicine, Brown University 2 Division of Applied Mathematics, Brown University
3 Division of Nuclear Medicine, Children's Hospital 4 Department of Radiology, Harvard Medical School

Abstract

This report describes initial experience with several image sharpening and registration tools and their applications in a longitudinal study of pediatric brain tumors (astrocytoma, medulloblastoma). Image sharpening is defined as the removal of unnecessary blurring of boundaries and shapes by some automatic method. We define registration as the superimposition or optimal matching of repeated pictures taken for a single patient over time or across different imaging modalities. A long-range goal is to develop, apply and extend these tools to extract the most clinically useful information from sets of serial single photon emission computed tomography (SPECT) and magnetic resonance (MR) whole brain scans taken both pre- and post-surgery for roughly seventy pediatric patients per year over a period of several years. We use two radioisotopically-labelled tracers, Thallium-201 and a technetium tracer, 99mTcHMPAO. SPECT images are more blurred than they need to be. Corrections for uniform photon attenuation (i.e. assuming only one medium such as soft tissue) have been shown to have initial success. Objective and highly-automated image registration through estimation of pixel-by-pixel deformation maps from one image to the next have also shown promise. The initial successes of these two developments in other contexts indicate that objective and highly automated methods could be developed to yield accurate, repeatable and verifiable methods for the extraction of useful SPECT/MR image summary measures in biomedical longitudinal studies, and in particular for the study of childhood brain tumors. The combination of these two technologies could improve both the quality of serial images and of biostatistical analyses of extracted imaging data in general. We give some initial results on the use of a method for sharpening SPECT images through estimation of attenuation and scattering functions and the use of a deformable template method to register a few SPECT slices for the same patient at different times.

* *Address for correspondence*: Nicholas Lange, Division of Biology and Medicine, Box G-A424, Brown University, Providence, RI 02912.

1. Motivation and context

The field of childhood brain tumors has shown substantial areas of promise in recent years with the development of new treatment protocols that improve outcome, both in terms of longevity and survival (e.g. in medulloblastoma, astrocytoma). Because of this improvement, there is a corresponding need for more precise methods of assessing tumor growth in order to be able to assess the response to treatment, and to be able with confidence to distinguish the presence of tumor from brain damage due to complications of the treatment. Early and accurate discrimination between these two types of post-treatment changes can be vital to the management of the patient.

There are two newly emerging biomedical imaging technologies: single photon emission computed tomography (SPECT) and magnetic resonance (MR) imaging. SPECT images provide data on internal functional, metabolic events through use of one or more radioisotopically labelled tracers. MR images, obtained without such tracers, provide data on structural anatomic features.

Among emission tomography techniques, as opposed to transmission tomography methods such as computed tomograph (CT), single photon emission computed tomography (SPECT) differs from positron emission tomography (PET) in at least two important ways. In SPECT, only one photon is released and recorded when a radioactive decay occurs, whereas in PET two photons are propagated in opposite directions and help to verify each others' emission points within the brain. SPECT is thus more prone to measurement errors. A benefit of SPECT technology, however, is that it is much less expensive and less cumbersome to operate, not needing a cyclotron. SPECT is the imaging modality of choice in our clinical setting due to this advantage. For a more complete description of SPECT, see for example Geman and McClure (1985, 1987), Manbeck (1990) and references therein.

Use of the radioisotopically labelled tracer Thallium-201 in SPECT provides a promising biological marker for the extent of biologically active tumor (Kaplan et al., 1987). The diagnostic specificity of Thallium-201 has been improved by performing an additional scan with the technetium

tracer 99mTcHMPAO to estimate cerebral perfusion. The demonstration of increased perfusion at the site of Thallium-201 abnormality favors active tumor.

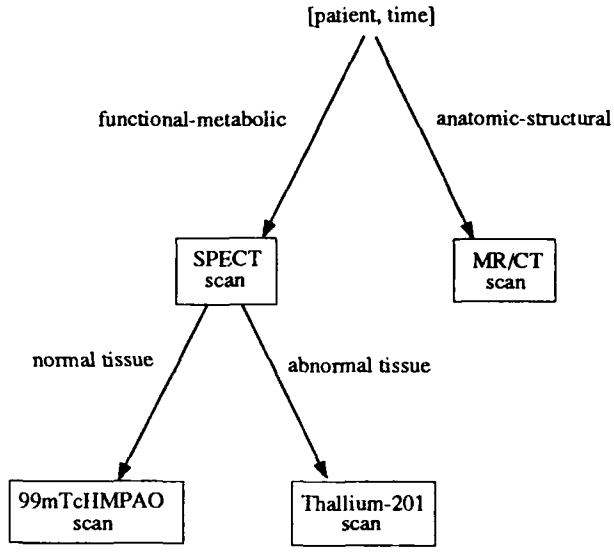
In this report, we explore both the global sharpening of SPECT images and an objective pixel-by-pixel registration method for repeated images taken on the same patient. Image sharpening is defined as the removal of unneccessary blurring of boundaries and shapes by an automatic objective method. Image registration is defined as the superimposition or optimal matching of serial images taken over time or across different imaging modalities, such as SPECT and MR, for a single individual, assisted in some cases by other sources including data from different individuals or, only if absolutely required, a knowledgeable expert.

SPECT images are much more blurred than they need to be. Many current "canned" and widely-used reconstruction algorithms, usually of the filtered back-projection type, do not correct for nonuniform photon attenuation and depth-dependent scatter, and do not account for the random nature of photon emission. Current image registration methods are often global in nature, highly operator-assisted, rely heavily on subjective judgements and human interactions, and yield results that are often non-repeatable and not objectively verifiable.

Two prospective clinical trials of radio/chemotherapy and surgery for astrocytomas and medulloblastomas, conducted by the Dana-Farber Cancer Institute, Boston, MA, provide a database of test images. Imaging for the clinical trials is performed at the Division of Nuclear Medicine, Children's Hospital, Boston, MA. Initial whole brain scans are obtained during pre-surgical evaluation as part of routine examination. At most seven triplets of SPECT and MR images at 1, 3, 6, 9, 12, 18 and 24 months are obtained during the course of the two year treatment schedule. Four additional sets of images for each patient are obtained, one for each year of follow-up observation. Differing numbers of pictures per patient arise because imaging data are missing at certain occasions and recorded at unscheduled times, and also due to patient death and censoring. These somewhat irregularly spaced measurements pose no problems whatsoever to our study. Our test database of SPECT/MR images contains such data for 50–75 pediatric patients at present. Figure 1 gives a diagram of the structure of images taken for each patient at each time.

Two problems hinder extraction of accurate and objective functional image summary measures from serial scans: (1) SPECT images are not as sharp as they could be if corrected for photon attenuation and scatter, and (2) there is a lack of an objectively derived common frame of reference within which to compare repeated images on the same patient over time, or to compare a picture for a particular

Figure 1.



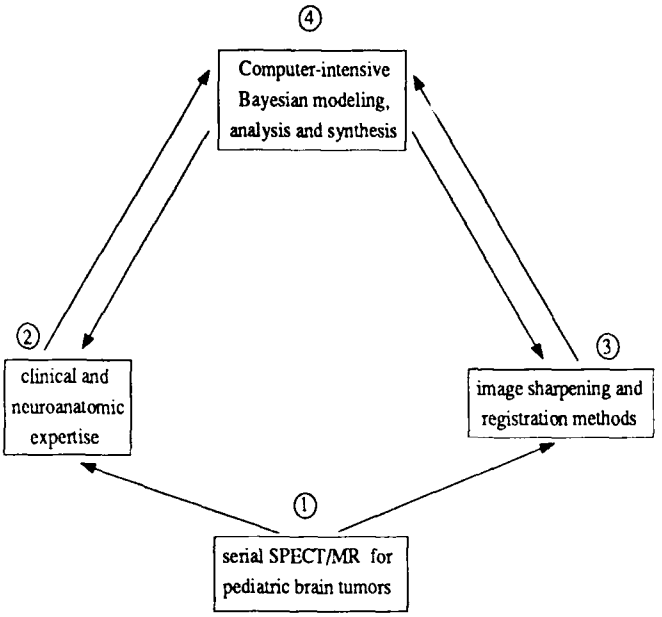
patient at a particular time to a corresponding normal individual's picture.

Stochastic models for repeated image summary measures require objective, well-defined criteria and repeatable procedures for their extraction if one is to be able to analyze time trends and treatment effects beyond otherwise requisite subjective and somewhat incommensurate clinical judgements. Trends in functional image summaries, such as may be present in repeated tumor/brain ratios and areas over time (O'Tuama et al., 1991), can be modeled through direct extensions of recent biostatistical methodology (see, for instance, Laird, Lange and Stram, 1987; Lange and Ryan, 1989; Lange and Laird, 1989; Gelfand and Smith, 1990; Lange, Carlin and Gelfand, 1991; Diggle, Lange and Beneš, 1991). In addition, the combination of functional-metabolic (SPECT) and structural-neuroanatomic (MR) findings holds great promise in providing more knowledge about the clinical state of the patient than could be provided by either one of these technologies alone (Pelizarri, Chen et al. 1989).

2. Goals and tools

Our long range goals are to obtain useful sets of objective, verifiable and repeatable image summary meaures, to model the stochastic processes generating these measures longitudinally over time, and to use the model results to improve clinical interpretations of the repeated images. Our immediate goals are to match SPECT slices for each patient over time, to obtain artefact-free SPECT reconstructions, and to try deformable template methods to obtain initial characterizations of tumor changes over time. Interactions between

Figure 2.



components of our long- and short-term goals are shown in Figure 2.

Among the tools available for our goals are landmark based global registration methods such as thin-plate splines (Bookstein, 1989), principal axes transformations (Alpert et al., 1990) and the "head and hat" method (Pelizarri, Chen et al. 1989). Possible complements to landmark-based global registration tools are methods for obtaining pixel-by-pixel image mappings. These include the use of "atlases" (Mowforth and Jin, 1988), "multi-resolution elastic matching" (Basesy and Kovačič, 1989), and the related deformable template methods (Chow, Grenander, and Keenan, 1988; Amit, Grenander and Piccioni, 1991).

The "head and hat" registration method, used at the Children's Hospital Medical Center, works as follows. Surface points on slices from SPECT and MR scans for a single patient are identified and thinned semi-automatically through manual editing of results obtained from standard outline extraction software. Once these external points have been identified, one has a rough SPECT "hat" which is to be fit to the MR "head". The fitting problem is solved by Pelizzari and Chen, et al. (1989) as a multivariate nonlinear regression, minimizing the sum of squared residual distances from the "hat" to the "head" along vectors through the center of the "head". Custom fitted "hats" are thus produced, and interior features interpolated linearly.

Available tools for SPECT reconstructions are the widely-used filtered back-projection methods. Also available are Bayesian reconstruction methods that use Markov random field image models with isotropic priors (Besag, 1974; Geman and Geman, 1984; Vardi, Shepp and Kaufman, 1985; Geman and McClure, 1985, 1987). Filtered back-projection methods can induce artifacts (boundary and shape blurring and smearing) when the filter applied to marginal projections does not anticipate certain asymmetries in these projections. Corrections also need to be made for photon attenuation and scatter effects. Weighted distance methods, such as the "Chang algorithm" (Chang, 1978), are widely used. Other methods estimate and correct for SPECT machine-specific parameters (Geman, Manbeck and McClure, 1991).

As has been described by Geman and McClure (1985, 1987) and by Geman, Manbeck and McClure (1991), photon attenuation can be accommodated through specification of a matrix $\mathbf{A}$, a discrete attenuated Radon transform, operating on an unobserved true image's isotope concentration map $\mathbf{X}$ to yield an expected observed image $\mathrm{E}(\mathbf{Y})$. A Bayesian image reconstruction model typically assumes that the image actually observed is $\mathrm{E}(\mathbf{Y})$ together with Poisson noise, i.e., that $\Pr(\mathbf{Y}|\mathbf{X})$ is Poisson with mean $\mathbf{AX}$. The reconstruction problem is to estimate $\mathbf{X}$ from $\mathbf{Y}$ while accounting for $\mathbf{A}$ to find an approximation to the posterior mean $\sum_X \mathbf{X}\Pr(\mathbf{X}|\mathbf{Y})$, by the method of iterated conditional expectations (Owen, 1986) for instance.

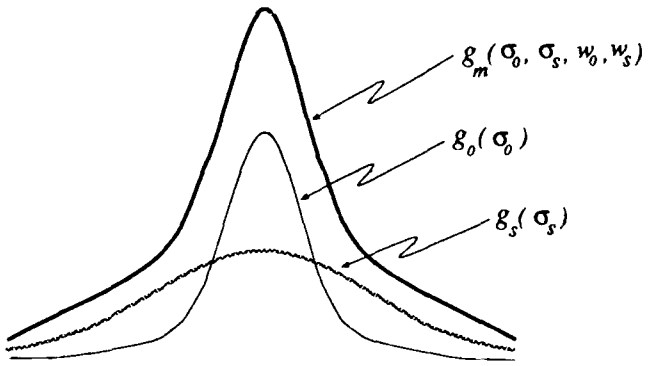
Two methods for estimating the non-uniform attenuation suggest themselves: to use a CT scan that measures attenuation directly, and/or estimate attenuation from a MR scan, or to estimate attenuation directly from the raw SPECT data. We choose the latter approach, a cleaner albeit scientifically more challenging solution. As described by Geman, Manbeck and McClure (1991), differences between observed and actual photon counts arise from three main sources: collimator effect, scatter fraction and attenuation. Collimator effect arises from "stray" photons being recorded in collimators other than those directly in line with their original trajectories. Scatter fraction is the proportion of "stray" photons among the total number detected that account for a depth-dependent blurring of the image. Attenuation is the process by which some emitted photons are not detected, due to insufficient energy to complete their paths to collimator(s) through differing media such as bone and soft tissue. We describe the attenuation correction method in some detail in the next section.

3. SPECT machine-specific parameter estimation

Correction for attenuation and scatter effects can be accommodated by estimating the discrete attenuated Radon transform $\mathbf{A}$. This requires estimation of a line spread function g_s due to scattered indirect photon counts. This function is not observed directly, and is modeled as a weighted dif-

ference between observed line spread functions g_0 for the ambient medium (air) and g_m for denser media such as soft tissue ($m = 1$) or bone ($m = 2$). Figure 3 gives an illustrative diagram. Weights w_0 and w_s enter in the estimation of g_s as functions of the number of observed photons and the unobserved proportion of these photons due to scatter (the "scatter fraction"). Both of these weights depend on the distances of events to the gamma camera. The line spread functions g_0 and g_s are modeled under suitable parametric assumptions (eg. double exponential, Gaussian). The line spread functions do not need to be assumed members of any parametric family of curves, however. The standard deviations σ_0 and σ_s are modeled as linear functions of distance. These functions are used to obtain interpolated values of line spread functions at all distances. The scatter fraction has been shown to be as high as 30–40% of the total at certain sites, for instance in regions of 8–10cm internal depth (Penny, et al., 1990; Geman et al., 1991). It is this high fraction of scattered photons that seems to account for much of the internal blurring of SPECT images, although much is still unknown about this property in general biomedical contexts.

Figure 3.



Estimation for uniform attenuation and scatter requires performing two phantom experiments, one through the ambient medium alone (air) and one through medium $m = 1$ alone (which we chose as water). Let B denote the number of detector bins "off" of the center as the random variable of interest, D the distance through the attenuating medium, and d ($= D +$ constant) the total distance from the from the point source to the gamma camera. Geman, Manbeck and McClure (1991) model the line spread functions as

$$g_1(b|d, D) = w_0(D)\, g_0(b|d) + w_s(D)\, g_s(b|D)$$

with

$$g_0(b|d) \sim \mathcal{N}(b, \sigma_0^2(d))$$
$$g_s(b|D) \sim \mathcal{N}(b, \sigma_s^2(D))$$

and

$$w_0(D) = e^{-cD}, c \text{ known}$$
$$w_s(D) = \text{ 'scatter fraction'}$$
$$= \gamma_2 e^{-\gamma_1 D} - e^{-cD},\ \gamma_1, \gamma_2 \text{ unknown.}$$

The constant c depends on the attenuating medium as well as on the tracer used, and can be obtained from known, available sources.

Thus the line spread functions are assumed to be Gaussian ridges with depth-dependent variances. The estimated depth-dependent standard deviations are set equal to their expectations, in standard method of moments fashion, and assumed to be linear functions of distance, i.e.

$$\begin{aligned} \mathrm{E}(\hat{\sigma}_0) &= \sigma_0 = \alpha_0 + \beta_0 d \\ \mathrm{E}(\hat{\sigma}_s) &= \sigma_s = \alpha_s + \beta_s D. \end{aligned} \tag{1}$$

A generalization of this approach, if the problem's complexity required, would be to use the method of estimating functions (Godambe, 1960) or the related generalized estimating equations method (Liang and Zeger, 1986). In our approach, the coefficients in (1) are estimated by the method of ordinary least squares. The task is then to determine which regions in a particular SPECT image correspond to the different media. This can be done either by estimating an additional unknown vector of pixel labels, greatly increasing the dimensionality of the problem, or through labelling each pixel using a map derived from a concurrent MR scan and a working registration of the two images by matching suspected skull boundaries. A more automatic method for this second approach would be to use global ellipses (eg. Alpert et al., 1990) as approximations to skull boundaries, or to represent irregular boundaries by a modified Fourier series (eg. Zahn and Roskies, 1972). In our present case, we labeled pixels in different regions by using crude ellipses, the approximate shape of the hot ring of the scalp.

4. Serial SPECT registration by deformable templates

We have chosen to focus our efforts at finding common and useful frames of references for serial SPECT scans by further development of the pixel-by-pixel method of registration by deformable templates (Amit et al., 1991). This is a local method by which one obtains a deformation map that connects each pixel in one picture into its mate in another picture through minimization of a global goodness-of-fit criterion, while maintaining smoothness constraints in some cases.

Denoting each pixel location by coordinates $\mathbf{x}$, the method of deformable templates assumes that a SPECT image I_t for a particular patient at a particular time t over

Table 1. Results of the phantom experiments. Estimates are reported in centimeters.

	Thallium-201		99mTcHMPAO	
Standard deviation	intercept (α)	slope (β)	intercept (α)	slope (β)
air (σ_0)	1.49	.005	1.46	.033
scatter (σ_s)	2.53	.037	2.33	.260
scatter fraction (γ_1, γ_2, c) in cm^{-1}	(.095, 1.15, .194)		(.012, 1.0, .150)	

a domain S is a deformation of an earlier image $I_{t'}$, the template, taken for this same patient. One determines the deformation map of $I_{t'}$ into I_t by finding values of coefficients $\xi_1, \ldots, \xi_P$ for which the integrated squared distance between images,

$$\int_{x \in S} \left| I_t(\mathbf{x}) - I_{t'}\left(\mathbf{x} + \sum_{p=1}^{P} \xi_p \varphi_p(\mathbf{x})\right)\right|^2 d\mathbf{x}\,, t' < t, \quad (2)$$

is a minimum. In (2), the functions $\varphi_1, \ldots, \varphi_P$ are orthonormal basis functions such the Fourier basis or the "wavelet" basis (eg. Mallat, 1987). Minimization of (2) is done by gradient descent. As discussed by Amit et al. (1991), the prior distribution on the set of possible mappings from $I_{t'}$ into I_t is taken as multivariate Gaussian and concentrated near the identity map, where $\sum_{p=1}^{P} \xi_p \varphi_p(\mathbf{x}) = 0$. If desired, one may include a regularization term in (2) that penalizes non-smooth mappings. Note that no landmarks are required by this method, making it much less operator-assisted and subjective, and more automated and objectively verifable than many existing registration methods. It is not yet known to what extent the deformable template method can be complemented by landmark-based methods.

5. Some initial results

Figure 4 shows transverse Thallium-201 SPECT slices for a single patient at two different times, both post-surgery, spaced about two months apart. Reconstruction was by commerically available filtered back-projection with Chang attenuation correction. The images are arranged in three rows and two columns. The rows proceed from about ear level toward the top of the head, and are at roughly the same level across columns. The first column is for the first scan, the second column for the subsequent scan. The hot ring in each is due to the uptake of Thallium-201 in the scalp. The hot spots in areas interior to the ring indicate active tumor. Although one may be able, by the eye, to infer that the tumor has grown from one occasion to the next, quantification of such suspected changes in tumor size (eg. edge, area and/or volume), as well as quantification of suspected changes in tumor shape, would be affected strongly by artifacts induced by the filtered back-projection reconstructions, and not highly reliable.

5.1. Sharpening

SPECT machine-specific parameter estimates from the phantom experiments are shown in Table 1. Note the higher variability and scatter fraction in the weaker Thallium-201 scans.

Figure 5 shows the result of applying the Bayesian image reconstruction model described in §2 with an **A** matrix that accommodates corrections for attenuation and scatter, obtained from the phantom experiment results shown in Table 1. Reconstruction artifacts appear greatly reduced and areas of tumor activity more localized.

5.2. Registration

Figure 6 shows an application of the deformable template method described in §3 to some filtered back-projection reconstructions (not included in Figure 4). It would have been preferable to apply this method to the sharper reconstructions, but this was not possible by the time of this writing. The upper lefthand frame of Figure 6 is the "template" $I_{t'}$. The lower righthand frame is the observed deformation I_t later in time. The upper righthand frame is the pixel-by-pixel difference between the observed images. The lower lefthand frame is the estimated deformation of $I_{t'}$ into I_t. The map to the right gives a vector for each pixel indicating the direction and distance each has moved from the template image to its deformation. Note a general outward movement from a suspected tumor center. However, artifacts in the filtered back-projection reconstructions seem to preclude reliable clinical interpretation of this estimated deformation map.

6. Summary

We have found in our initial experiments that for our porblem the use of several imaging modalities is essential in order to obtain results and image interpretations that are clinically reliable, We have found in addition that corrections for rigid-body motions (translation, rotation, scale) when comparing different scans are also mandatory. When a goal is to obtain reliable, verifiable and repeatable SPECT image summary measures, we question the use of filtered back-projection reconstructions of the low-energy Thallium-201 scans for pediatric brain tumors. Corrections for non-uniform photon attenuation and scatter are eseential. We have demonstrated that objective, Bayesian image restoration methods can yield results that are relatively artefact-free. More work need to be done with the application of the deformable template method in our context, in particular with the sharpened iamges. Objective and verifiable pixel-by-pixel characterizations of tumor changes over time do appear feasible, however. External, historical atlases may help in normal tissue typing and exclusion tasks. One of the next steps in our research will be to try out some semi-automatic edge extraction methods on the sharpened SPECT images, such as the graduated non-convexity algorithm proposed by Blake and Zisserman (1987), which is programmed and available.

References

Alpert, N. M., Bradshaw, J. F., Kennedy, D. and Correia, J. A. (1990). The principal axes transformation—A method for image registration. *Journal of Nuclear Medicine*, 31, 1717–1722.

Amit, Y., Grenander, U., and Piccioni, M. (1991). Structural image restoration through deformable templates. To appear, *Journal of the American Statistical Association.*

Bajcsy, R. and Kovačič, S. (1989). Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46, 1–21.

Besag, J. (1986). On the statistical analysis of dirty pictures (with discussion). *Journal of the Royal Statistical Society, Series B*, 48, 259–302.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society, Series B*, 36, 192–236.

Blake, A. and Zisserman, A. (1987). *Visual Reconstruction.* Cambridge, MA: MIT Press.

Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intel'igence*, 11, 567–585.

Chang, L. T (1978). A method for attenuation correction in radionuclide computed tomography. *IEEE Transactions on Nuclear Science*, NS-25, 638–643.

Chow, Y., Grenander, U. and Keenan, D. M (1988). HANDS: A pattern theoretic study of biological shape. Technical report, Division of Applied Mathematics, Brown University.

Diggle, P. J., Lange, N. and Beneš, F. M. (1991). Analysis of variance for replicated spatial point patterns in clinical neuroanatomy. To appear, *Journal of the American Statistical Association* (September).

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85, 398–409.

Geman, S. and McClure, D. E. (1987). Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 52, 5–21.

Geman, S. and McClure, D. E. (1985). Bayesian image analysis: An application to single photon emission tomography. In *Proceedings of the Statistical Computing Section, American Statistical Association*, 12–18.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741.

Geman, S. Manbeck, K. M. and McClure, D. E. (1991). A comprehensive statistical model for single photon emission tomography. To appear, *Markov Random Fields: Theory and Application* (R. Chellappa and A. Jain, eds.)

Godambe, V. P. (1960). An optimum property of regular maximum likelihood estimation. *Annals of Mathematical Statistics*, 31, 1208–1211.

Kaplan, W. D., Takvorian, T., Morris, J. H. et al. (1987). Thallium–201 tumor imaging: A comparative study with pathologic correlation. *Journal of Nuclear Medicine*, 28, 47–52.

Laird, N., Lange, N. and Stram, D. (1987). Maximum likelihood computations with repeated measures: Application of the EM algorithm. *Journal of the American Statistical Association*, 82, 97–105.

Lange, N. and Ryan, L. (1989). Assessing normality in random effects models. *Annals of Statistics*, 17, 624–642.

Lange, N. and Laird, N. M. (1989). The effect of covariance structure on variance estimation in balanced growth-curve models with random parameters. *Journal of the American Statistical Association* 84, 241-247.

Lange, N., Carlin, B. P. and Gelfand, A. E. (1991). Hierarchical Bayes models for the progression of HIV infection using $CD4^+$ counts. Submitted.

Liang, K-Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika* 73, 13–22.

Mallat, S. G. (1987). A theory for multiresolution signal decomposition: the wavelet representation. Preprint, GRASP Lab., Department of Computer and Information Science, University of Pennsylvania.

Manbeck, K. M. (1990). Bayesian statistical methods applied to emission tomography with physical phantom and patient data. Unpublished Ph.D. dissertation, Division of Applied Mathematics, Brown University.

Mowforth, P. H. and Jin, Z. (1989). Model based tissue differentiation in MR brain images. Unpublished manuscript, The Turing Institute, Glasgow.

O'Tuama, L. A., Janicek, M. J., Barnes, P. D. et al. (1991). 201-Tl/99mTcHMPAO SPECT imaging of treated childhood brain tumors. To appear, *Pediatric Neurology*.

Owen, A. B. (1986). Discussion of "Statistics, images and pattern recognition" by B. D. Ripley. *Canadian Journal of Statistics*, 14, 106–110.

Pelizzari, C. A. Chen, G. T. Y., Spelbring, D. R., Weichselbaum, R. R. and Chen, C-T. (1989). Accurate three-dimensional registration of CT, PET, and/or MR images of the brain. *Journal of Computer Assisted Tomography*, 13(1), 20–26.

Penny, B. C. King, M. A. and Knesaurek, K. (1990). A projector, back-projector pair which accounts for the two-dimensional depth and distance dependent blurring in SPECT. *IEEE Transactions in Nuclear Science*, 37, 681–686.

Vardi, Y., Shepp, L. A., and Kaufmann, L. (1985). A statistical model for positron emission tomography (with discussion). *Journal of the American Statistical Association*, 80, 8–37.

Zahn, C. T. and Roskies, R. Z. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on Computing*, C-21, 269–281.

Figure 4. Filtered back-projection reconstructions of transverse Thallium-201 SPECT slices for a single pediatric patient. Rows proceed from about midsection of the brain toward the top of the head. The first column is at the first scan, the second column at the second scan about two months later. The rings are due to tracer uptake in the scalp. Uptake areas interior to the rings indicate active tumor.

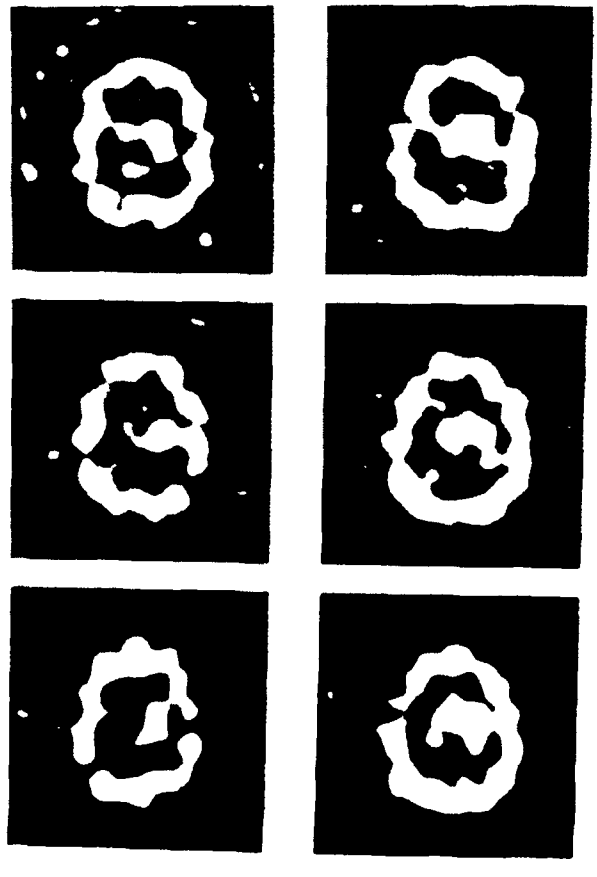


Figure 5. Several filtered back-projection reconstructions (above), with their Bayesian, Markov random field reconstructions with corrections for attenuation and scatter from the phantom experiments (below).

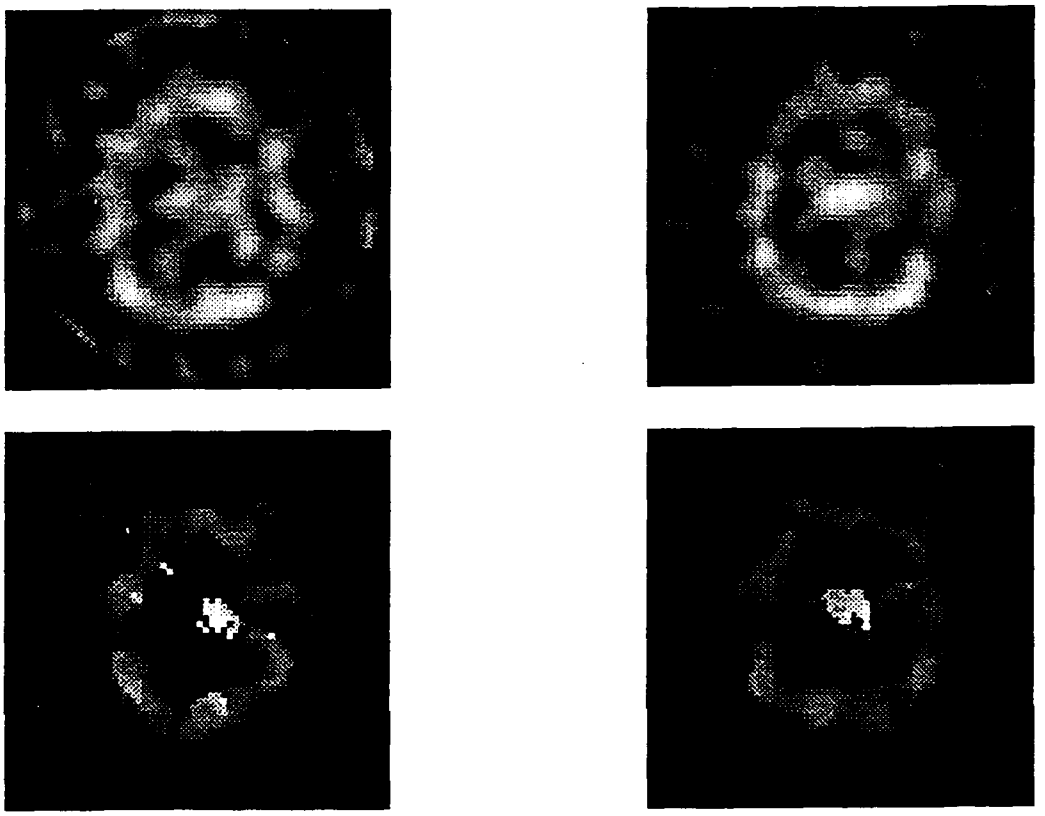
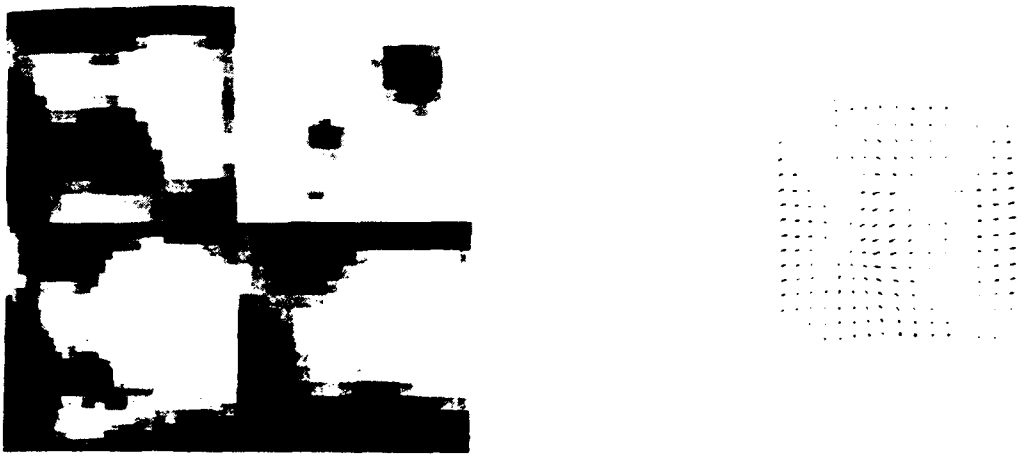


Figure 6. Results of an initial pixel-by-pixel registration of a selected regions of filtered back-projection scans by the deformable template method. *Frames to the left:* upper lefthand: the "template" $I_{t'}$; lower righthand frame: the observed deformation I_t later in time; upper righthand; the pixel-wise difference between the observed images; lower lefthand frame: the estimated deformation of $I_{t'}$ into I_t. The estimated pixel-by-pixel map is shown on the right.



Statistical Shape Models in Image Analysis

K.V. Mardia, J.T. Kent and A.N. Walder
Department of Statistics
University of Leeds
Leeds LS2 9JT
United Kingdom

Abstract

We discuss several models for shapes in the plane based on the distributions of landmarks about an underlying template. The motivation for these models includes Markov random fields and thin plate splines. These models are used as priors in a Bayesian framework to reconstruct a shape from a digital image. An example is given based on the human hand.

1 Introduction

In this paper we shall discuss methods to pick out a shape from a two-dimensional digital image. The shape is assumed to be a deformation of some underlying shape or 'template', and the image is also subject to observational noise. We represent points in the plane as complex numbers. We shall focus attention on the case where the shape can be described as a simply connected domain $D \subset \mathcal{C}$ whose boundary consists of piecewise linear path connecting vertices $z_0, z_1, ..., z_n \in \mathcal{C}$ with $z_0 = z_n$. Let $V = \{z_j\}$, termed the 'outline' of the shape, denote the set of vertices. Similarly, let $V_0 = \{\mu_j\}$ say, denote the outline of the underlying template.

The deformation from V_0 to V consists of two types of transformations. The first type of transformation consists of global linear changes such as (a) location, (b) scale, (c) rotation and possibly (d) a more general linear transformation of the plane. The second type of transformation consists of local changes to the outline. In this paper we shall discuss various probability models for the local change to the outline (including the location change). Thus we will get a probability distribution $P(V)$ on the outline of our shape, centred at the underlying template V_0. Some possible models are given in Sections 2-3.

For other aspects of the deformation, such as scale and rotation changes we shall use ad hoc fitting procedures. An example involving the reconstruction of a hand is given in Section 4. Thus our approach to modelling shapes differs from other approaches (*eg.* Goodall, 1991, Bookstein, 1986, Kendall, 1984, Kent, 1991, Mardia & Dryden, 1989) in which location, scale and rotation effects are incorporated directly into the models.

The observed digital image includes information about the given shape, together with observational errors. One possible model is

$$y_\ell = \nu_1 + \epsilon_\ell \qquad \text{if } \ell \in \mathrm{D}$$

$$y_\ell = \nu_2 + \epsilon_\ell \qquad \text{if } \ell \notin \mathrm{D} \tag{1.1}$$

where $y_\ell \in \Re$ denotes the 'grey-level' in the ℓ^{th} pixel and $\ell = (\ell_1, \ell_2)$ labels the pixels in an $L_1 \times L_2$ grid. In the simplest version of the model we suppose the ϵ_ℓ are independent $N(0, \sigma_\epsilon^2)$ random variables. The mean levels ν_1 and ν_2 indicate the difference between the shape and the background. Thus, given V the model for $\mathbf{y} = \{y_\ell\}$ has pdf

$$P(\mathbf{y}|V) \propto exp\{-\frac{1}{2\sigma_\epsilon^2}[\sum_{\ell \in D}(y_\ell - \nu_1)^2 + \sum_{\ell \notin D}(y_\ell - \nu_2)^2]\}. \tag{1.2}$$

If we let $P(V)$ denote the prior probability density of V under a deformable template model then by Bayes theorem

$$P(\mathbf{y}|V)P(V) \tag{1.3}$$

is proportional to the posterior density of V given the data. We seek an estimate of V to maximise (1.3) This estimate is known as the MAP or 'maximum a posteriori' estimate.

The main focus in this paper is on suitable models for $P(V)$ which we explore in Sections 2-3. To some extent our paper is a review of models proposed by previous authors, but we also bring out some unifying themes behind the models together with some new results.

2 Complex normal models on outlines

Suppose $(Re\ z_1, Im\ z_1, \ldots, Re\ z_n, Im\ z_n)' = \mathbf{r}$ say, follows a 2n-dimensional (real) normal distribution with mean $(Re\ \mu_1, Im\ \mu_1, \ldots, Re\ \mu_n, Im\ \mu_n)$ and $2n \times 2n$ covariance matrix $\mathbf{\Omega}$ say. Typically $\mathbf{\Omega}$ will be small so that the distribution of the observed set of landmarks $V = \{z_j\}$ will not be too far from the template $V_0 = \{\mu_j\}$. For simplicity we shall suppose that $\mathbf{\Omega}$ possesses complex symmetry. That is if we write $\mathbf{\Omega} = (\mathbf{\Omega}_{jk})$ in terms of 2×2 blocks $\mathbf{\Omega}_{jk}$, $j, k = 1, \ldots, n$, then

$$\mathbf{\Omega}_{jk} = \alpha_{jk} \begin{pmatrix} cos\theta_{jk} & -sin\theta_{jk} \\ sin\theta_{jk} & cos\theta_{jk} \end{pmatrix}$$

for some $\alpha_{jk} = \alpha_{kj} \geq 0$ and angle $\theta_{jk} = -\theta_{kj} \in [0, 2\pi)$. In particular $\theta_{jj} = 0$. We can also represent $\mathbf{\Omega}$ as an $n \times n$ complex matrix $\mathbf{\Sigma}$ with $\sigma_{jk} = \alpha_{jk} exp(i\theta_{jk})$. Then $\mathbf{r}'\mathbf{\Omega}\mathbf{r} = \mathbf{z}^*\mathbf{\Sigma}\mathbf{z}$ where $\mathbf{r}'$ denotes the transpose of $\mathbf{r}$ and $\mathbf{z}^* = \bar{\mathbf{z}}'$ denotes the transpose of the complex conjugate of $\mathbf{z}$.

Complex symmetry is often too restictive an assumption to lead to good models for outline data (see *eg.* the figures in Goodall,1991; Dryden & Mardia,1991). However, since it may not be essential to specify the prior distribution $P(V)$ very precisely, the assumptions of complex symmetry may be adequate. In any case the models here can be generalised to the non-complex-symmetric case at the expense of extra notation and additional parameters.

The simplest general model for the vertices $\mathbf{z} = (z_1, \ldots, z_n)'$ about $\mu = (\mu_1, \ldots, \mu_n)'$ is a multivariate complex normal model

$$f(\mathbf{z}) \propto exp\{-\frac{1}{2}(\mathbf{z} - \mu)^*\mathbf{A}(\mathbf{z} - \mu)\}.$$

where the inverse covariance matrix $\mathbf{A}$ is Hermitian. It is simplest to suppose that $\mathbf{A}$ is positive semi-definite of rank n-1, with $\mathbf{A1} = \mathbf{0}$ so that the distribution of $\mathbf{z}$ is improper. Here $\mathbf{1} = (1, 1, \ldots, 1)'$ and $\mathbf{0} = (0, 0, \ldots, 0)'$. Thus $f(\mathbf{z}) = f(\mathbf{z} + \alpha\mathbf{1})$ for any $\alpha \in \mathcal{C}$. The reason for this choice is that we are not usually interested in location differences when judging the similarity of a given outline $\mathbf{z}$ to the template μ.

Without further restriction the matrix $\mathbf{A}$ contains too many parameters to represent a useful model. Therefore it is of interest to look at some special cases.

2.1 The vertex CAR model

Following Besag (1974) the simplest model for the vertices is a first-order conditional autoregressive (CAR) model. Equivalently we require $\mathbf{A}$ to be cyclic tridiagonal. The conditional distribution of z_j given the rest of the points $\{z_k : k \neq j\}$ is complex normal with first two moments

$$E[z_j|rest] = \alpha_j z_{j-1} + \beta_j z_{j+1}$$

$$var[z_j|rest] = \tau_j^2 \qquad (2.1)$$

say, where $\alpha_j, \beta_j \in \mathcal{C}$, $\alpha_j/\tau_j^2 = \beta_{j-1}/\tau_{j-1}^2$, and $\alpha_j + \beta_j = 1$, j=1,...,n. In terms of the elements of $\mathbf{A}$,

$$a_{jj} = 1/\tau_j^2,\ a_{j,j-1} = -\alpha_j/\tau_j^2,\ a_{j,j+1} = -\beta_j/\tau_j^2. \quad (2.2)$$

Here and elsewhere we interpret the subscripts mod n. Remember that the parameters must be chosen so that $\mathbf{A}$ is positive semi-definite of rank n-1.

The simplest version of this model is obtained when $\tau_j^2 = \tau^2$ say, does not depend on j and $\alpha_j = \beta_j = 1/2$ for all j.

2.2 The CAR transformation model on edges

Let $e_j = z_j - z_{j-1}$, $\eta_j = \mu_j - \mu_{j-1}$, j=1,...,n, denote the edges between successive vertices in the random outlines and the template, respectively. Note that $\sum e_j = \sum \eta_j = 0$. Write

$$e_j = (1 + t_j)\eta_j,\ t_j \in \mathcal{C}. \qquad (2.3)$$

Then t_j measures the extent to which e_j differs from template edge η_j. Chow et al (1988) proposed a conditional cyclic-stationary first-order CAR model for $t_1, \ldots, t_n$, conditioning on $\sum t_j\eta_j = 0$.

In its unconditional form the CAR model for the $\{t_j\}$ can be written in the form

$$E[t_j|rest] = -(\bar{\delta}/\beta)t_{j-1} - (\delta/\beta)t_{j+1},$$

$$var[t_j|rest] = 1/\beta, \qquad (2.4)$$

where $\beta > 0$ and $\delta \in \mathcal{C}$. Thus the (unconditional) pdf of $t_1, \ldots, t_n$ is proportional to

$$exp\{-\frac{1}{2}[\beta\sum|t_j|^2 + \delta\sum\bar{t}_j t_{j+1} + \bar{\delta}\sum t_j\bar{t}_{j+1}]\}$$

$$= exp\{-\frac{1}{2}[\beta\sum|t_j|^2 + 2Re(\delta\sum\bar{t}_j t_{j+1})]\}. \qquad (2.5)$$

The sums here range over $j = 1, \ldots, n$ and subscripts are to be interpreted mod n. A sufficient condition to ensure that the covariance matrix of the $\{t_j\}$ is positive definite is $|\delta|/\beta < 1/2$. After conditioning, the distribution of

$(t_1, \ldots, t_n)$ is no longer a CAR, though it is still complex normal.

A multivariate complex normal distribution on $t_1, \ldots, t_n$ induces a multivariate normal distribution on the edges $e_1, \ldots, e_n$. Further if we allow the location of the vertices $z_1, \ldots, z_n$ (as measured by the centroid, say) to have an improper uniform distribution over $\mathcal{C}$ (note the location of the outline is not determined by the edges), then we can transform the above distribution on edges to give an improper multivariate complex normal distribution on the vertices $z_1, \ldots, z_n$.

Write $\omega_j = z_j - \mu_j$. After a little algebra it follows that the distribution of $(z_1, \ldots, z_n)$ is an improper second-order CAR with

$$\begin{aligned} E[\omega_j|rest] = \tau_j^2\{&\beta(|\eta_j|^{-2}\omega_{j-1} + |\eta_{j+1}|^{-2}\omega_{j+1}) \\ &-\delta\eta_j\bar{\eta}_{j-1}|\eta_j|^{-2}|\eta_{j-1}|^{-2}(\omega_{j-1} - \omega_{j-2}) \\ &+\bar{\delta}\eta_{j+1}\bar{\eta}_{j+2}|\eta_{j+1}|^{-2}|\eta_{j+2}|^{-2}(\omega_{j+2} - \omega_{j+1}) \\ &-|\eta_j|^{-2}|\eta_{j+1}|^{-2}(\bar{\delta}\eta_j\bar{\eta}_{j+1}\omega_{j+1} + \delta\eta_{j+1}\bar{\eta}_j\omega_{j-1})\}, \end{aligned} \quad (2.6)$$

$$var[\omega_j|rest] = \tau_j^2,$$

where

$$\begin{aligned} \tau_j^2 = \{&\beta(|\eta_j|^{-2} + |\eta_{j+1}|^{-2}) \\ &-|\eta_j|^{-2}|\eta_{j+1}|^{-2}(\delta\bar{\eta}_j\eta_{j+1} + \bar{\delta}\eta_j\bar{\eta}_{j+1})\}^{-1}. \end{aligned} \quad (2.7)$$

An important special case occurs when the template vertices form a regular polygon, *i.e.* $\mu_j = exp(2\pi ij/n)$. In this case

$$\begin{aligned} E[\omega_j|rest] &= \tau^2|a-1|^2\{(\beta - 2\delta a)\omega_{j-1} \\ &\quad +(\beta - 2\bar{\delta}\bar{a})\omega_{j+1} + \delta a\omega_{j-2} + \bar{\delta}\bar{a}\omega_{j+2}\}, \end{aligned} \quad (2.8)$$

$$var[\omega_j|rest] = \tau^2,$$

where

$$\tau^2 = \frac{1}{2}|a-1|^2(\beta - Re\ \delta a)^{-1}, \text{ and a} = \exp(2\pi\text{i}/\text{n}). \quad (2.9)$$

2.3 A Covariance Model

The above models are useful when landmarks can be consistently identified on the template and the observed outline. However, in some examples, *eg.* an outline of a biological cell, there are no identifiable features and the n landmarks might be defined to be equally-spaced (in terms of arc length) around the outline of the object. In this case it is reasonable to take the template to be a regular n-sided polygon, with $\mu_j = exp(2\pi ij/n)$, and to model the variety of possible shapes using a circulant Toeplitz covariance matrix, as proposed by Miller *et al* (1991). Defining $t_1, \ldots, t_n$ as in (2.3), they model $(t_1, \ldots, t_n)$ as a multivariate complex normal distribution with circulant Toeplitz covariance matrix $\mathbf{B}$, say, conditional on $\sum t_j e_j = 0$. That is, $\mathbf{B} = (b_{jk})$ has entries $b_{jk} = \alpha_{j-k}$, say, where $\alpha_{j-k} = \bar{\alpha}_{k-j}$. Here as elsewhere subscripts are to be interpreted mod n. The eigenvectors of $\mathbf{B}$ are

$$\mathbf{g}_k = \frac{1}{\sqrt{n}}[exp(-2\pi ijk/n), j = 1, \ldots, n]'$$

for $j = 1, \ldots, n$ with eigenvalues

$$\lambda_k = \sum_{j=1}^{n} \alpha_j exp(-2\pi ijk/n),$$

$k = 1, \ldots, n$. The eigenvalues, assumed to be non-negative, are not necessarily in any monotone order.

Let $\mathbf{G}$ be the $(n \times n)$ unitary matrix with columns $\mathbf{g}_k$, and set

$$\mathbf{s} = \mathbf{G}^*\mathbf{t}$$

to be the vector of principal components. The constraint $\sum t_j e_j = 0$ takes an appealing form in terms of principal components,

$$\begin{aligned} \sum t_j(e^{2\pi ij/n} - e^{2\pi i(j-1)/n}) &= [1 - e^{-2\pi i/n}]\sum t_j e^{2\pi ij/n} \\ &= \sqrt{n}[1 - e^{-2\pi i/n}]s_1 = 0, \end{aligned}$$

that is, $s_1 = 0$. Miller *et al* (1991) suggest estimating the parameters in $\mathbf{B}$ by using a training sample of m outlines. Equivalently, after rotating the principal components for each outline, one can estimate the eigenvalue λ_k by $(2m)^{-1}$ times the sum of squared absolute values of the k^{th} principal component in the training sample. Further, since each outline in the training sample will satisfy the constraint $u_1 = 0$, we will always estimate $\lambda_1 = 0$.

Miller *et al* (1991) also suggest a modification to this model in which the real and imaginary parts of $(t_1, \ldots, t_n)$ are modelled independently using separate circulant Toeplitz matrices (with real entries). However this modification lacks the appealing rotational invariance of the original model. Note that the CAR in (2.8) and (2.9) is a special case of this model.

3 Continuous deformable template models - thin plate splines

Another way to model the transformation between the template and the realised outline is to fit a deformation

of $\mathcal{C}$, that is a continuous transformation $z \mapsto w(z)$, from $\mathcal{C}$ to $\mathcal{C}$. The most common such model is the thin-plate spline (Bookstein,1989). The purpose of this section is to explore some of the algebraic aspects of this method.

For this section let $z_j = (x_j + iy_j), j = 1, \ldots, n$ denote the landmarks in the template (denoted by μ_j before) and $w_j = u_j + iv_j$, the transformed landmarks (denoted by z_j before). The output of the thin-plate spline algorithm is a function $w : \mathcal{C} \mapsto \mathcal{C}$, $w(z) = u(z) + iv(z)$, satisfying $w(z_j) = w_j, j = 1, \ldots, n$.

One way to calculate the thin-plate spline is through kriging, which we now briefly describe. The functions $u(z)$ and $v(z)$ are fitted separately as follows. Consider the function

$$\sigma(z) = |z|^2 log|z|^2 + c_1 + c_2x^2 + c_3y^2 + c_4xy \qquad (3.1)$$

where $z = x+iy$ and c_1, c_2, c_3, c_4 are arbitrary constants. (We shall see below that the choice of c_1, c_2, c_3, c_4 has no effect on the final solution).

Let $z_0 = x_0 + iy_0$ be a new point at which we wish to define $u(z_0)$. (In this section subscripts are not to be interpreted mod n; z_0 should *not* be identified with z_n). The kriging approach says to take $u(z) = \alpha'\mathbf{u}$ where α (depending on z_0) is chosen to minimise

$$\beta'\mathbf{A}\beta \text{ subject to } \mathbf{S}\beta = \mathbf{0} \qquad (3.2)$$

where $\beta = (-1, \alpha')'$ is an $(n+1)$−vector, $\mathbf{A}$ is an $(n+1) \times (n+1)$ matrix with entries

$$a_{jk} = \sigma(z_j - z_k) \qquad (3.3)$$

and $\mathbf{S}$ is a $3 \times (n+1)$ matrix,

$$\mathbf{S} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_0 & x_1 & \ldots & x_n \\ y_0 & y_1 & \ldots & y_n \end{pmatrix}. \qquad (3.4)$$

The matrix $\mathbf{A}$ is conditionally positive definite; that is $\beta'\mathbf{A}\beta > 0$ if $\beta \neq \mathbf{0}$ and $\mathbf{S}\beta = \mathbf{0}$ (Matheron,1972). Further it is easily checked that if $\mathbf{S}\beta = \mathbf{0}$ then $\beta'\mathbf{A}\beta$ does not depend on the values of c_1, c_2, c_3, c_4 above.

The motivation for the criteria (3.2) comes from the theory of first-order intrinsic random fields. There exists a real-valued intrinsic random field $\{X(z) : z \in \mathcal{C}\}$ such that whenever $\mathbf{S}\beta = \mathbf{0}$, the increment $\beta'[X(z_0), \ldots, X(z_n)]'$ has mean 0 and variance $\beta'\mathbf{A}\beta$. Further, if $\beta = (-1, \alpha')'$, then

$$var\{\beta'[X(z_0), \ldots, X(z_n)]'\}$$
$$= E\{X(z_0) - \alpha'[X(z_1), \ldots, X(z_n)]'\}^2$$

represents the prediction mean squared error of the random field at the new site z_0 in terms of a linear combination of its values at the existing sites $z_1, \ldots, z_n$.

Partition

$$\mathbf{A} = \begin{pmatrix} \sigma(0) & \delta' \\ \delta & \mathbf{\Sigma} \end{pmatrix}, \ \mathbf{S} = \begin{pmatrix} 1 & \vdots & \\ x_0 & \vdots & \mathbf{T} \\ y_0 & \vdots & \end{pmatrix}, \qquad (3.5)$$

where the matrix $\mathbf{\Sigma}$ depends only on the data $z_1, \ldots, z_n$, but the vector δ $(n \times 1)$ also depends on the coordinates of the new point z_0.

With a suitable choice of c_1, c_2, c_3, c_4 we can ensure that $\mathbf{\Sigma}$ is positive definite. Then using Lagrange multipliers it is straightforward to show that the choice of α minimising (3.2) is

$$\alpha = \mathbf{\Sigma}^{-1}\delta - \mathbf{\Sigma}^{-1}\mathbf{T}'(\mathbf{T}\mathbf{\Sigma}^{-1}\mathbf{T}')^{-1}\mathbf{T}\mathbf{\Sigma}^{-1}\delta$$
$$-\mathbf{\Sigma}^{-1}\mathbf{T}'(\mathbf{T}\mathbf{\Sigma}^{-1}\mathbf{T}')^{-1}[1, x_0, y_0]' \qquad (3.6)$$

so that

$$u(z_0) = \mathbf{u}'\mathbf{B}\delta - \ell' \begin{pmatrix} 1 \\ x_0 \\ y_0 \end{pmatrix}, \qquad (3.7)$$

say. Here $\ell = (\mathbf{T}\mathbf{\Sigma}^{-1}\mathbf{T}')^{-1}\mathbf{T}\mathbf{\Sigma}^{-1}\mathbf{u}$ is the generalised least square regression coefficient of u_j on $(1, x_j, y_j)$, $j = 1, \ldots, n$ and $\ell'(1, x_0, y_0)'$ is the generalised least squares predictor at the new point z_0. It can also be checked that the value of ℓ does not depend on c_1, c_2, c_3, c_4 above.

Also, in (3.7),

$$\mathbf{B} = \mathbf{\Sigma}^{-1} - \mathbf{\Sigma}^{-1}\mathbf{T}'(\mathbf{T}\mathbf{\Sigma}^{-1}\mathbf{T}')^{-1}\mathbf{T}\mathbf{\Sigma}^{-1}. \qquad (3.8)$$

If we let $\mathbf{P} = \mathbf{T}'(\mathbf{T}\mathbf{T}')^{-1}\mathbf{T}$ denote the orthogonal projection matrix in $\Re^n$ onto the columns of $\mathbf{T}'$ (so $\mathbf{P}\mathbf{T}' = \mathbf{T}'$), then it is not difficult to show that

$$\mathbf{B} = [(\mathbf{I} - \mathbf{P})\mathbf{\Sigma}(\mathbf{I} - \mathbf{P})]^- \qquad (3.9)$$

where $[\ ]^-$ denotes the Moore-Penrose generalized inverse. Further $\mathbf{B}^- = (\mathbf{I} - \mathbf{P})\mathbf{\Sigma}(\mathbf{I} - \mathbf{P})$. Note that the eigenvectors of $\mathbf{B}^-$ (corresponding to the non-zero eigenvalues) are all orthogonal to the columns of $\mathbf{T}'$. Hence the matrix $\mathbf{B}^-$ (and therefore $\mathbf{B}$) does not depend on the arbitrary choice of c_1, c_2, c_3, c_4.

The quantity $\mathbf{u}'\mathbf{B}\mathbf{u}$ is identified (see, for example, Wahba, 1990, p33) as being proportional to the bending energy of the transformation $z \mapsto u(z)$. It is also easily checked that $\mathbf{B} = \mathbf{B}\mathbf{\Sigma}\mathbf{B}$ and $\mathbf{B} = \mathbf{B}(\mathbf{I} - \mathbf{P})$. The thin-plate spline for $v(z)$ proceeds similarly.

Hence, given an underlying template of landmarks $z_1, \ldots, z_n$, it is natural to model the deformed landmarks $w_1, \ldots, w_n$ $(w_j = u_j + iv_j)$ using a complex normal distribution based on the bending energy,

$$P(\{w_j\}) \propto exp\{-\frac{1}{2\tau^2}[\mathbf{u}'\mathbf{B}\mathbf{u} + \mathbf{v}'\mathbf{B}\mathbf{v}]\}$$

$$\text{or, } P(\{w_j\}) \propto \exp\{-\frac{1}{2\tau^2}\mathbf{w}^*\mathbf{B}\mathbf{w}\} \qquad (3.10)$$

where τ^2 is a scale parameter. This density is improper since $\mathbf{B}$ has rank $n-3$ and further all linear transformations of $w_1, \ldots, w_n$ have the same density.

It would be interesting to apply this model in the analysis of images. Other than the scale constant τ^2 it contains no parameters to choose, once $z_1, \ldots, z_n$ are given. One choice of $z_1, \ldots, z_n$ is to take these as vertices of a regular polygon. Then $\mathbf{B}$ simplifies somewhat since Σ^{-1} is circulant Toeplitz as in the model of section 2.3 above. Further a similar construction can be carried out in dimensions other than 2.

4 Hand Reconstruction

We now consider an example of shape reconstruction for the human hand using the model (1.1) for observation noise and the edge model of section 2.2 for the prior distribution of the shape. The hand in the image is a real human hand and the template is formed from the average of 8 real hands. The data were provided by Dan Keenan. Our example is motivated by Chow *et al* (1988).

The shape model of section 2.2 contains two parameters $\delta \in C$ and $\beta > 0$. In our experiments we have limited consideration to δ real. We have reparameterised δ and β in terms of λ and σ^2 where

$$\lambda = \{\beta - (\beta^2 - 4\delta^2)^{1/2}\}/2\delta, \; \sigma^2 = (\beta^2 - 4\delta^2)^{-1/2}, \quad (4.1)$$

because they have more intuitive interpretations as the usual first-order autocorrelation and the marginal variance respectively, in the analogous discrete-time AR1 process from time series.

Our reconstruction procedure can be conveniently split into 3 stages.

1. First we want to find the appropriate location and orientation of the hand in the image, using a variant of thresholding. Our approach has been to use the alternating mean thresholding and median filtering (AMT-MF) approach of Mardia and Hainsworth (1988) to obtain a binary image. Setting $\hat{y}_\ell = 1$ inside the largest connected component and $\hat{y}_\ell = 0$ elsewhere gives an initial reconstruction. Here $\ell = (\ell_1, \ell_2)$ labels the pixels of the image.

2. Given a similar binary image $\{x_\ell\}$ for the interior of the template hand, and treating ℓ as a (2×1) column vector $\mathbf{l}$, construct an affine map $\mathbf{A}\mathbf{l} + \mathbf{b}$ so that the first two moments of $\{\mathbf{A}\mathbf{l} + \mathbf{b} : x_\ell = 1\}$ match the first two moments of $\{\mathbf{l} : \hat{y}_\ell = 1\}$. Third moments are used to resolve any orientation ambiguities. Further, small rotations of the template are examined to improve the fit, using the matching coefficient

$$\varphi = \sum x_\ell \hat{y}_\ell / \{\sum (x_\ell + \hat{y}_\ell - x_\ell \hat{y}_\ell)\}. \qquad (4.2)$$

3. We now make use of the shape model of section 2.2 in an approximate ICM algorithm (see Besag,1986). We cycle through the vertices z_j one at a time and using a grid search consider updates of z_j to maximise the posterior density (1.3). These cycles are iterated until convergence. In our example 4 cycles usually sufficed, reducing the grid size from 9×9 pixels down to 3×3 pixels as we progressed through the cycles.

Several features in our reconstruction algorithm are worth emphasising.

(a) The initial reconstruction (Stages 1 and 2) has a very important effect on the quality of the final reconstruction. (b) The number and location of the vertices is important. At the very least, to represent a hand we require the tips of the fingers and the lowest points between them and points at the wrist. However, our experiments indicate that these alone are not nearly enough, and typically we take a template with 51 vertices as in Figure 1(a). In total there are 256 points on the template, and the intermediate points are updated by interpolation. It is also important not to have too many vertices. Because our updating algorithm changes only one vertex at a time, it can get stuck in an unsuitable local optimum. Simulated annealing offers another way to cope with this difficulty, but at increased computational cost. (c) We have also imposed a 'hard-core' restriction to prevent vertices getting too close together, *eg.* bunching up near the tips of the fingers. In our experiments a minimum distance of 3 pixels between vertices was found useful. Bunching is generally a problem only when the noise is high.

Figure 1 shows the results of our algorithm on a 256×256 image. In (b) we have the true image, to which $N(0, \sigma_\epsilon^2)$ noise, $\sigma_\epsilon^2 = 4$, has been added, (c). Here $\nu_1 = 1$, $\nu_2 = 0$. Naive thresholding at $(\nu_1 + \nu_2)/2$ would give an error rate of 40 %. In (d) we have the effect of applying AMT-MF and (e) gives its largest components. In (f) we have the final reconstruction after 4 iterations of Stage 3, with parameters $\lambda = 0.5$, $\sigma = 0.2$. The pixel by pixel error rate is under 2 %.

Stage 1 does not assume any knowledge of ν_1 and ν_2. For comparison we tried Stage 3 assuming ν_1 and ν_2 known (yielding a matching coefficient of $\varphi = 0.83$ and displayed in Figure 1(f)) and with ν_1 and ν_2 estimated from Stage 1 and used in Stage 3 (yielding $\varphi = 0.81$).

Thus knowing ν_1 and ν_2 leads to only a slight improvement in the reconstruction. The ratio $\sigma_\epsilon^2/\sigma^2$ is treated as known and has been chosen by trial and error. Figure 2 shows our initial global matching afer Stage 2.

5 Other Work

For a review of other methods of deformable templates, see Lipson *et al* (1990). Amit *et al* (1991) describe pixel-based approach to fitting a deformation of $\mathcal{C}$. A description of the difficulties involved in three-dimensional problems is given by Grenander and Keenan (1989). Models for curvilinear shapes such as letters of the alphabet are discussed by Manbeck *et al* (1991). They use a second-order SAR, but note that the CAR model in (2.6) and (2.7), this time with boundary conditions, again provides a useful model.

Face recognition is an interesting application area for shape identification (cf. Bruce,1988, Craw & Tock,1991). Here there are nesting constraints. For example, pupils are nested within eyes, teeth within lips, eyes and lips within the head, etc.

To sum up, the area of shape reconstruction poses many interesting statistical problems.

6 Acknowledgements

This work was supported by a grant from the SERC/MOD. We are grateful to both Tim Hainsworth and John Haddon for their helpful comments, to Jackie Gough for the initial work, to Michael Miller for a preprint of his paper for discussion at the conference, and to Dan Keenan for the hand data.

7 References

Amit, Y., Grenander, U. & Piccioni, M. (1991) Structural image restoration through deformable templates. Div. of Applied Maths, Brown Univ.

Besag, J.E. (1974) Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. R. Statist. Soc. B* **36** 192-236.

Besag, J.E. (1986) On the statistical analysis of dirty pictures (with discussion). *J. R. Statist. Soc. B* **48** 259-302.

Bookstein, F.L. (1986) Size and shape spaces for landmark data in two dimensions (with discussion). *Statistical Science* **1** 181-242.

Bookstein, F.L. (1989) Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Machine* **11** 567-585.

Bruce, V. (1988) *Recognising Faces.* London, Lawrence Erlbaum Associates.

Chow, Y., Grenander, U. & Keenan, D.M. (1988) *HANDS: A pattern theoretic study of biological shapes.* Div. of Applied Maths, Brown Univ.

Craw, I. & Tock,D. (1991) The computer understanding of faces. *Processing images of faces.* Ed. V. Bruce & M. Burton. Ablex.

Dryden, I.L. & Mardia, K.V. (1991) General shape distributions in the plane. *Adv. Appl. Prob.* To appear.

Goodall, C. (1991) Procrustes methods in the statistical analysis of shape. *J. R. Statist. Soc. B* **53** 285-339.

Grenander, U. & Keenan, D.M. (1989) Towards automated image understanding. Special issue, Ed. K. V. Mardia. *J. Appl. Stats.* **16** 207-221.

Kendall, D.G. (1984) Shape-manifolds, procrustean metrics and complex projective spaces. *Bull. Lond. Math. Soc.* **16** 81-121.

Kent, J.T. (1991) The complex Bingham distribution and shape analysis. Dept. of Statistics, Leeds Univ.

Lipson, P., Yuille, A.L., O'Keeffe, D., Cavanaugh, J., Taaffe, J. & Rosenthal, D. (1990) Deformable templates for feature extraction from medical images. *Computer Vision - EECV 90.* Ed. O. Faugeras, Springer-Verlag, Berlin.

Manbeck, K., Elion, J. & Geman, S. (1991) Machine recognition of human coronary arteries. Div. of Applied Maths, Brown Univ.

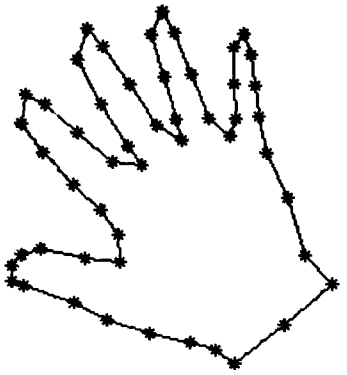
Mardia, K.V. & Dryden, I.L. (1989) Shape distributions for landmark data. *Adv. Appl. Prob.* **21** 742-755.

Mardia, K.V. & Hainsworth, T.J. (1988) A spatial thresholding method for image segmentation. *IEEE Trans. Pattern Anal. Machine* **6** 919-926.

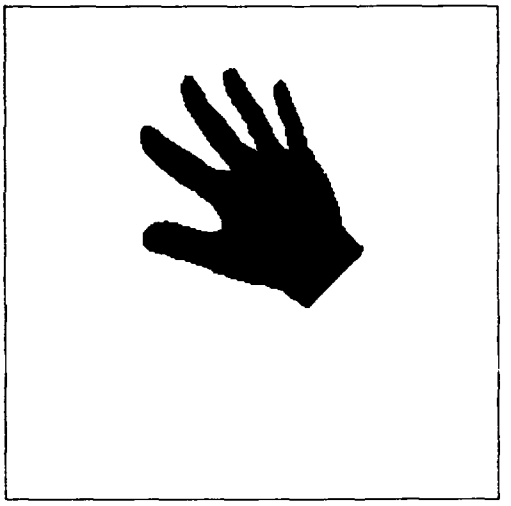
Matheron, G. (1972) *The theory of regionalised variables and its applications.* Les Cahiers du Morphologie Mathematique, Fasc. No. 5, Fountainbleau.

Miller, M., Maffitt, D., Shrauner, J., Roysam, B. & Grenander, U. (1991) Automated segmentation of biological shapes in electron microscopic autoradiography. *Interface '91 Proceedings.*

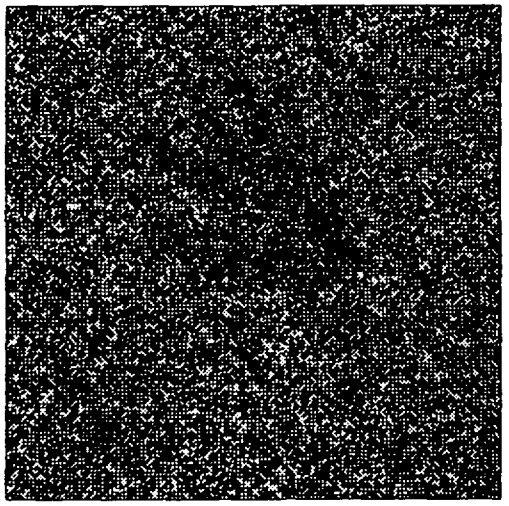
Wahba, G. (1990) *Spline models for observational data.* S.I.A.M., Philadelphia.



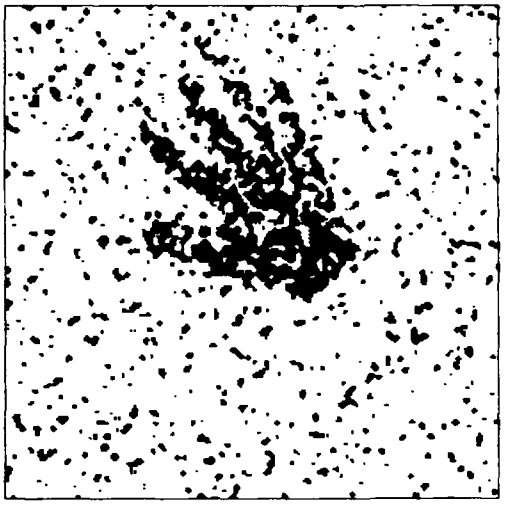
1A) HAND TEMPLATE, 51 VERTICES



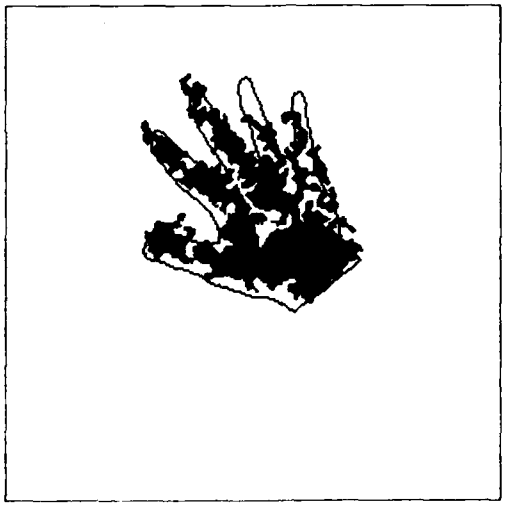
1B) ORIGINAL HAND



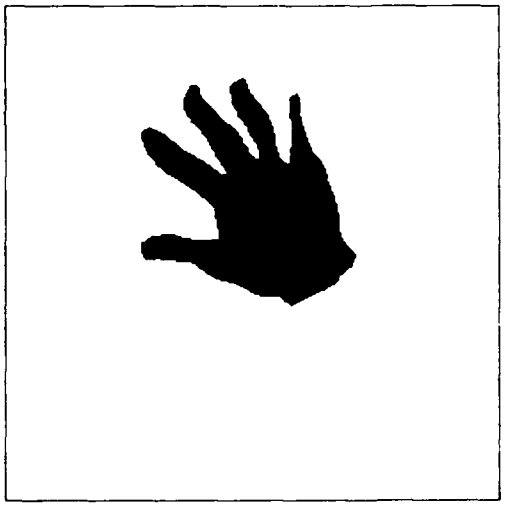
1C) WITH ADDED NOISE



1D) AFTER AMT-MF



1E) LARGEST COMPONENT AND INITIAL TEMPLATE



1F) FINAL RECONSTRUCTION

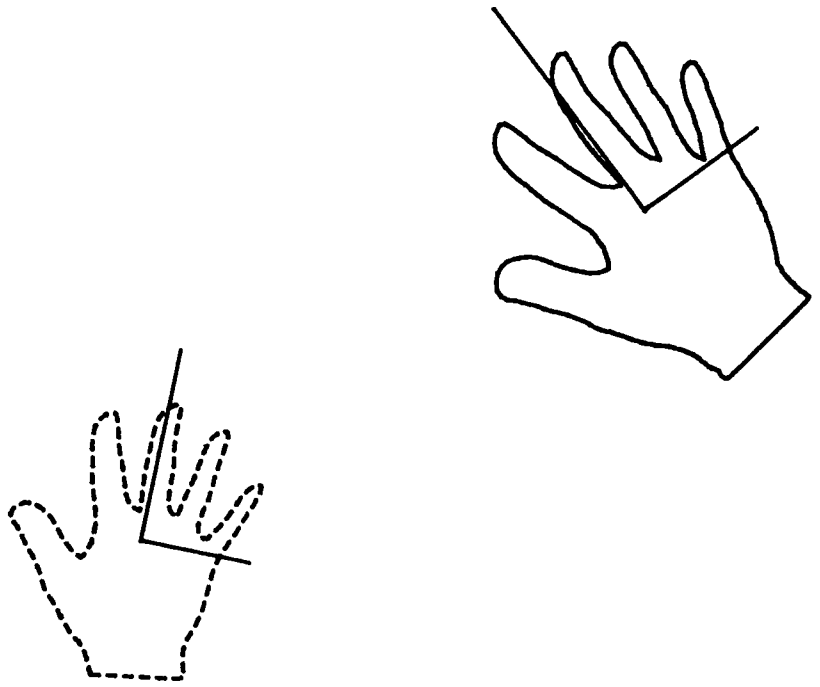


Fig. 2a INDIVIDUAL PRINCIPAL AXES

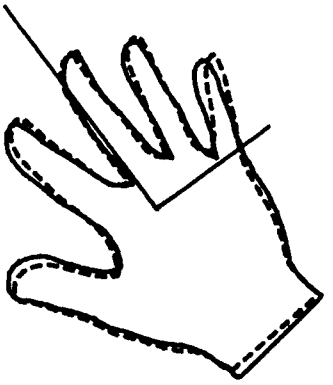


Fig. 2b TEMPLATE SUPERIMPOSITION

Discussion: Multivariate Statistics and Visualization of Labelled Point Data

Fred L. Bookstein
University of Michigan
Ann Arbor, Michigan 48109

In this brief Discussion I would like to weave together the key words and phrases from the title of this session: "multivariate statistics," "labelled point data," and "visualization." Their interplay underlies a useful duality of approaches to many problems of data analysis in a domain of growing practical importance: statistics of image data.

Data from images

The problems I have in mind originate in data that are already visualized. The source of information might be a medical image, perhaps, indicating a physical property (some sort of interaction with radiation) within each of a grid of little volumes inside a region of tissue. Or it might be a geological survey, showing physical or electromagnetic properties near each of a vastly sparser grid of points on the surface of the earth (or in a solid chunk of its interior); or a weather map, indicating physical properties of the air (composition, temperature, suspensates, velocity) near a still sparser grid of points in the three-dimensional atmosphere. In a great variety of scientific contexts, our concern is to investigate aspects of this sort of data many sets at a time: a heap of synoptic weather maps, a span of eons of continental drift, or a sample of biological or biomedical histories. That is, we wish a **synthetic image** concentrating certain features of particular interest drawn from the context of rather dilute information that is each original brain scan, or survey, or weather map. To emphasize this task of comparison rather than the pursuit of arbitrary detail is to ask a different sort of question than that the original visualization was designed to answer. The goal now is to retrieve not what is unique to each instance but what is common to all, what is most variable among them, what typically covaries with exogenous causes or effects, etc. Tools are needed for carrying out some steps in this search in a manner requiring the least attention, the least interaction, or the greatest degree of automation.

Let us be a bit more formal. The data under discussion are, in general, vectors at each point of a domain organized on Euclidean principles—multicolored pixels or voxels or air samples out in the woods. Think of these as a spectrum of scalars on a surface drawn or interpolated "above" each point of the domain. Such pictures can be very beautiful, but we shall ignore that distraction. We pursue the alternative visualizations to arrive at scientific understanding, not necessarily further pretty pictures. The Cartesian product of picture plane or grid or space by the length of the vector of observation is occasionally augmented by another Cartesian factor corresponding to replication (e.g., over time); ordinarily, but not always, this factor can be folded into the length of the vector content.

Vertical and horizontal

A conventional multivariate analysis will usually deal with aspects of this surface in a manner that ignores the prior geometric ordering. The usual multivariate formalism proffers only one quadratic form, representing the variance-covariance matrix; perhaps there is also a custom-designed "error covariance matrix" incorporating information about adjacencies. Otherwise the geometric origin of the index set of variables is nowhere in evidence. Let us call that kind of statistics "vertical." There is always additional information, then, in the *horizontal* part of this imagined figure—the information about where the labelled locations of the ground plane actually lie, and how their locations covary with height(s) of the surface(s) above them. This horizontal part is what we primates are used to processing. It is borne on the nonlinear world of the retina, rich in alternate

visualizations of pattern and color, of depth and of motion, of prey and of predators, of swinging from limb to limb or leg to leg. The conventional multivariate approach ignores the evolutionary history of the organism that has brought forth statisticians. The single visualization of abstract linear vector spaces hardly deserves the name of "visualization" at all. Whenever data is originally visual, and especially if it is originally gridded, the linear machinery *must* be supplemented, if not wholly supplanted, by a semantics of pushes and pulls, of motion and deformation.

Matters are easiest if we restrict ourselves, in accord with the title of the present session, to *labelled* points in this horizontal dimension, points that have specific names corresponding from instance to instance of the image. Note that it is the points of the ground plane (or ground space, in the 3-d applications) that are labelled, not the points of the imaginary "surfaces" of data floating above them. These labels can be fixed n-uples, like map coordinates, or they can move about on a subordinate map of their own, like the bridge of your nose as it locates your eyeglasses on your profile. Labelled points, it turns out, support a feature space quite a bit more promising than that which is accessible in the general run of multivariate problems. For instance, sets of labelled points have their own metrics, usually hugely transmogrified versions of ordinary interpoint distances or Cartesian coordinates, that complement the usual covariance-based metric of multivariate observations. The labelled points can move about in their Euclidean domain at the same time that images change above them, leading to decompositions of the variance "at" a point that are very interesting both scientifically and statistically. One can ask, for instance, whether regions of the cardiac wall that show abnormal changes of curvature, as indicated by the relative motion of the arterial bifurcations nearby, are the same sectors as those showing anomalies of texture in an ultrasound scan.

The three lectures in this session all deal with the relations between "vertical" and "horizontal" aspects of this sort of data, relations very conveniently filtered through the low dimensionality and immense graphical power of labelled points. In the first lecture, Paul Sampson and colleagues show a visualization of the relation between the two distances, horizontal and vertical (geometrical and statistical), for the same set of labelled points which, being geographical sites, are in fact invariable in position. This relation between two distances is visualized *as if* it were a deformation of the map, the "rubber sheet" that is our most familiar imagery for changing distance measures. In this way Sampson reduces "vertical" covariation to "horizontal" visualization. In this synthetic horizontal structure, the principal components that would be lengthy vectors of coefficients point by point in the "vertical" analysis become, instead, extended curves at 90°—the biorthogonal grid of the horizontal analysis. That is, we have turned the relation between a vertical covariance matrix and a horizontal geometric distance matrix into a single visual entity, a horizontal symmetric tensor field, graphed using a pair of directions at every point.

The other two lectures, Lange's and Mardia's, may be thought of as treating strategies for understanding the interplay of "vertical" and "horizontal" analyses of the same "three-dimensional" topographic data. For instance, a horizontal analysis may be best if one wants to use the geometry of the labelled image rather as one uses a covariate in a classic experimental design. In this case it is as if the shape of the configuration of labelled points—the very basis of the vector space underlying the data—is effectively nuisance variation. "Controlling" this variation increases the precision with which other effects can be addressed. That is, one analyzes *vertically*—examining the gradients of the picture, for instance, or its correlations with physical or biological processes—after "unwarping" horizontally to a less blurry feature space in which processes more nearly "stay put" to have their averaged picture taken (Bookstein, 1991*b*). In multivariate language, we are projecting out a complicated nonlinear feature co-space. The experience of generations of anatomists has shown that this maneuver improves the power of subsequent multivariate tactics, such as discrimination or analysis of covariance. When averaging pictures of brain activity over brains of different shape, for instance, the landmarks serve as guides to the correspondence of regions prior to averaging. It is the landmarks, not the squares of the grid of a PET reconstruction, that represent the true coordinate system for valid biometric analyses.

Horizontal analyses

In other applications, this "horizontal"

variation is not noise or nuisance, but itself a signal in its own right. The labelled points support a very powerful low-dimensional feature space, in effect a tangent space to Kendall's shape space (Bookstein, 1991; Goodall, 1991) in the vicinity of the mean configuration of labelled points. With the aid of a convenient basis for the elements of this shape space, this information may be concentrated into linear features of its own. When the variables of this block are paired with less delicately crafted descriptors of the original "vertical" scalar or vector content, there results a sort of hierarchical multiple-regression approach for prediction of other images, such as later images of the same system, and for the joint evocation of shape and content as a bispectral signal in a detection or classification problem, such as locating tumors or quantifying their recession under treatment. For brain shape, for instance, the statistics of this "horizontal" space suggest some unwarpings that might be unusually effective at unblurring the subsequent vertical average (Bookstein, 1991*b*).

It turns out that visualization of vectors in the feature space of labelled point shape—that is, shape changes in sets of landmarks—is at least as easy as visualization of changes in surfaces above the planes or volumes tagged by those points. The best visualizations are suggestive of the process explanations, the "bulges" and "shears" and "warps," that are automatically familiar to any sentient organism that ever navigated a binocular landscape. The combination of features of labelled point shape with features of the image "at the average shape"—the careful separation of vertical from horizontal variation in these mixed feature spaces, and the careful, specialized visualization of the horizontal—is, in my view, the most powerful generator available for good analyses of biometrical images.

Mixed analyses

The freedom to combine a geometric metric with the customary statistical one is unfamiliar to most applied statisticians. An analogy from physical science may be useful: this is precisely the same freedom as is granted us by Newtonian mechanics—the existence of absolute space, and absolute time, and hence an absolute scale of relative velocities in meters per second, independent of all the other quantitative laws of physics. (It is this decoupling that is contravened by Einstein's special theory of relativity; but Einstein was not a statistician in the sense we are using that word here.) Geometrically, this construction is called a "Galilean metric." Space is measured in centimeters, and time in seconds, and there is no absolute constant of conversion between them—no "speed-of-light"—but only diverse objects and their velocities, each of which is an empirical matter. In the analogous context of images over labelled point data, there is a collection of distance measures for landmark configurations and another collection of distance measures for multivariate distributions; and the relation of these two sets of measures is purely an empirical matter, as encoded, for instance, in a singular-value decomposition. The peculiar advantage of labelled point data is the unexpected simplicity of its own statistical structure. Many transformations that appear hopelessly nonlinear in terms of the multivariate space oriented "vertically" turn out to be linear, or nearly so, in aspects of the same space viewed and measured horizontally. There are, then, many more practicable and interesting directions of projection of these composite spaces than would be available in an ordinary problem having the same net count of degrees of freedom.

Let us consider, for example, the difference between two kinds of analysis of relations among pictures: "motion" and "deformation." For the case of "motion," consider a one-dimensional picture, pixels in a line. Our task is to detect an extended point moving uniformly along this grid. Under the (physically reasonable) assumption of linear motion, any such detection is a linear projection—the averaging of values $p(x\text{-}\nu t)$—in a direction of the Cartesian product space (image pixels by replications) taking account of the speed ν of the motion. Hence motion of a point can be detected by a one-dimensional suite (varying ν) of linear operators applied to a higher-dimensional representation.

But consider, now, the problem of detecting *deformation*, like the reflection of your face in a flawed mirror, or the growth of your child's face over time. The corresponding transformations of feature space are *not* linear in the extent of deformation. As landmarks move over distances at greater than subpixel scale, the linearity of geometry in the underlying ground plane is converted into sharp turns in the linear space of vectors over pixels in which the conventional multivariate statistics is mounted. The "same"

tissue signal lies over weighted averages of pixels (p_i,p_{i+1}), then (p_{i+1},p_{i+2}), etc. Each of these segments makes an angle of 135° with its predecessor and successor, and angles of 90° with all the other such segments in the linear space of pixel-by-pixel content. Then as landmarks move over pixels, the resulting series of transformations is far from a linear extension: in multivariate space, it makes a wrenching turn every time a pixel boundary is crossed. Smoothing algorithms can lower the variance associated with these turns, but they cannot evade the underlying geometric infelicity. Yet the averaging of biological images is made vastly more powerful when these nonlinear transformations are executed first (Bookstein, 1991*b*). In practice, these techniques combine among themselves for analyses of motion over a deforming scene, such as when the flexing of a joint deforms surrounding tissues, or in the contraction of a heart that is bouncing on its tether within the chest wall.

The variety of shape metrics

The useful metrics for shape itself are perhaps unfamiliar. They include the Procrustes metric of minimal rms Euclidean distance (Goodall, 1991), the deficient metric of localized shape difference restricted to the space of residuals from affine transformations (Bookstein, 1991), the hyperbolic log-anisotropy metric for uniform shears (ibid.), and others mixing shape information with information about size. The available composites for combining information from the labelled points with information from the "surface" of the image are then far more intricate than those of Hamiltonian mechanics, with its geometrization of Newton's laws. The composite metrics apposite to the understanding of sets of images can incorporate correlations of "height" and its spatial derivatives with shape and its alternate metrics in endless combinations (Bookstein, 1991*a*). Consider, for instance, the problem of detecting growth in a brain tumor. This is the correlation of one visual texture to the interior of a disk under a barrel distortion, and the correlation of another field of *motion* to the exterior of the same disk, all as constrained (with considerable real physical nonlinearity!) by the bony margin of the braincase. The resulting "metric" has no easy illustration other than the very picture of the mixed analysis we would thereby be operationalizing—warping of the interior and the exterior of a labelled disk (the tumor "boundary") separately, followed by vertical comparisons of tumor texture, dissections of the motion of arterial bifurcations, and so on.

To this diversity of metrics corresponds an equal diversity of notions of orthogonal projection. The number of different ways in which features can be measured or, alternatively, projected out out of these composite spaces, is thus fairly rich. One's choice depends on the specific sort of pattern being sought in the data analysis, which is to say, on the process governing the composite image (weather, Alzheimer's disease, continental drift). We can seek to describe the variation (in the labelling plane) of the location of a "point" feature (vanishing of a derivative) or instead the location of an "edge" (vanishing of a second derivative); or we may attempt instead to minimize the variation of location of these features so as to ease the study of something else about the picture (for example, the texture of ventricular borders in Binswanger's disease). One might extend the correspondence of labelled points to curves connecting them; there results a tessellation of the plane into corresponding regions suited for regional averages, coefficients of variation, etc. We can study wave-like phenomena either as the vertical motion of vectors at fixed points or as the horizontal motion of nodes at extremal points, whichever corresponds better to the dynamics of the underlying morphogenetic process. We can attempt to measure the deviation of a spatial surface in between landmarks, in order to study its regional fractal dimension or other aspects of geometrical texture; or we can attempt to flatten this variation of height onto a map so as to study autocorrelation of grey levels or thickness of surface layers in the true Gaussian surface metric. Either of these types of registration may be used to generate sample means for purely descriptive purposes or, alternatively, may be turned to the investigation of group differences or covariation with other aspects of the picture, with causes, or with effects. We can correlate values, or gradients, or the Hessian of the scalar load over a region with a Cartesian coordinate or with a tensor representing some aspect of the relation of the labelling configuration to the mean, such as the Jacobian of the implied deformation. And so on, through many other possibilities, whether in one dimension or in a higher space.

Concluding remark

In this brief compass I can no more than hint at the power for image analysis and scientific insight of the new methods that exploit labelled point data to enrich the conventional multivariate metaphor. In this combination of real (physical, binocular) geometry with the abstract geometry of linear models lies the key to most problems of pattern detection and display across the sciences that begin with real fields: images over maps, bodies, or space, in one instantiation or several, in one color or multispectrally, at one instant or many in linear or cyclic time. The key to the combination of the vertical and the horizontal metrics is their careful separation to begin with: separation of change of image content from repositioning of the carrier pixels. The separation proceeds best with the aid of an unique intermediate statistical structure, the nonstandard multivariate technology of shape space for labelled point configurations. In its peculiar finite-dimensional elegance, this space affords a basis for linear features of the arbitrarily nonlinear transformations that we see, and explain, in the real world. Many of the processes accounting for variation in these images can be modeled as nearly linear in these transformations, and many other questions are made quite a bit less murky after that linearizable part of image variation is partialled out of the image.

Acknowledgement

Preparation of these comments was underwritten in part by NIH grants NS-26529 and GM-37251 to Fred L. Bookstein.

Literature cited

Bookstein, Fred L. 1991. *Morphometric Tools for Landmark Data.* Cambridge University Press, 1991, in press.

Bookstein, Fred L. 1991*a*. Four metrics for image variation. In D. Ortendahl and J. Llacer, eds., *Proceedings of the XI International Conference on Information Processing in Medical Imaging.* Progress in Clinical and Biological Research, vol. 363. New York: Wiley-Liss, Inc., 1991 pp. 227–240.

Bookstein, Fred L. 1991*b*. Thin-plate splines and the atlas problem for biomedical images. To appear in *Proceedings of the XII International Conference on Information Processing in Medical Imaging*, eds. A. Colchester and D. Hawkes. Lecture Notes in Computer Science, Springer-Verlag.

Goodall, C. R. 1991. Procrustes methods in the statistical analysis of shape (with discussion and rejoinder). *Journal of the Royal Statistical Society* B53:285–339.

Exploring Posterior Distributions Using Markov Chains

Luke Tierney*
School of Statistics
University of Minnesota
Minneapolis, MN 55455

Abstract

Several Markov chain-based methods are available for sampling from a posterior distribution. Two important examples are the Gibbs sampler and the Metropolis algorithm. In addition, several strategies are available for constructing hybrid algorithms. This paper outlines some of the strategies that are available, and discusses some theoretical and practical issues in the use of these strategies. In addition, some preliminary efforts to use Markov chains to control dynamic graphics for exploring higher-dimensional posterior distributions are outlined.

1 Introduction

Suppose we are given a posterior distribution π on a quantity θ with values in a space E. Usually E will be a subset of $\mathbb{R}^k$ and π will have a density with respect to a measure μ,

$$\pi(dx) = \pi(x)\mu(dx).$$

For simplicity, π will be used to denote both the distribution and the density. We may be interested in computing a particular numerical characteristic of π, or more generally in developing an understanding of what information π contains about θ.

Several methods for computing characteristics of posterior distributions are now available. These include asymptotic approximations, numerical integration, and sampling or Monte Carlo methods. Sampling methods for examining posterior distributions provide ways of generating samples with the property that the empirical distribution of the sample, or an appropriately weighted empirical distribution, approximate the posterior distribution. Using such samples, it is easy to estimate characteristics such as the mean or standard deviation of a function of θ. Marginal distributions can be estimated using smoothing or, in some cases, variance reduction methods. In addition, for equally weighted samples methods for viewing point clouds, such as rotating plots and Grand Tours, can be used to examine the joint uncertainty about three or more components or features of θ.

A number of different sampling methods are available. In rare cases it is possible to sample directly from the posterior distribution and thus obtain an *i.i.d.* sample from π. In most problems this is not possible. Either the sample has to be dependent, or the distribution used to generate the sample has to be different from π. A method that uses independent samples from a distribution similar to π is importance sampling. The sample is then weighted to make up for the difference between π and the distribution used to generate the sample. Over the past decade, most work on sampling methods for exploring posterior distributions has centered on importance sampling (Geweke, 1989; Stewart, 1979; van Dijk *et al.*, 1978; Zellner and Rossi, 1984; among others). An alternative approach that avoids the need for weights is to use a dependent sample, such as the sample path of a Markov chain.

2 Markov Chain Methods

Markov chain methods generate a sample path from a Markov chain that has π as its stationary distribution. Recent work of Gelfand and Smith (1990) on the Gibbs sampling algorithm has renewed interest in Markov chain methods for exploring posterior distributions. Gelfand and Smith extend the Gibbs sampling algorithm of Geman and Geman (1984), originally developed for Bayesian image reconstruction, to continuous distributions and show how the algorithm can be used in a wide variety of problems.

Markov chain methods have a long history in Mathematical physics dating back to the algorithm of Metropolis *et al.* (1953). The Metropolis algorithm is in fact a general class of algorithms that includes versions of the discrete Gibbs sampler as special cases.

*Research supported in part by grant DMS-9005858 from the National Science Foundation

2.1 The Metropolis Algorithm

Metropolis *et al.* (1953) originally proposed the algorithm now known as the Metropolis algorithm as a method of sampling from the equilibrium distribution of an interacting particle system. The algorithm, which is described in Hammersley and Handscomb (1964, Section 9.3) and Ripley (1987, Section 4.7), was extended by Hastings (1970) and explored further by Peskun (1973).

To define Hastings version of the algorithm, let Q be a Markov transition kernel with

$$Q(x, dy) = q(x, y)\mu(dy).$$

Let $E^+ = \{x : \pi(x) > 0\}$, and assume $Q(x, E^+) = 1$ for $x \notin E^+$. Then define

$$\alpha(x, y) = \min\left\{\frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}, 1\right\}$$

for $\pi(x)q(x, y) > 0$. Otherwise, $\alpha(x, y) = 1$. If the Markov chain is currently at $X_n = x$, then the algorithm generates a candidate $Y = y$ for the next state from $Q(x, \cdot)$. With probability $\alpha(x, y)$ this candidate is accepted and the chain moves to $X_{n+1} = y$. Otherwise, the candidate is rejected and the chain remains at $X_{n+1} = x$.

Since

$$\pi(x)q(x, y)\alpha(x, y) = \pi(y)q(y, x)\alpha(y, x),$$

a Metropolis chain with initial distribution π is reversible. Therefore π is an invariant distribution for the chain. Some additional conditions on π and Q are needed to insure that π is also a limiting distribution; these conditions are discussed in Section 3 below. Since the acceptance probability only depends on π through the ratio $\pi(y)/\pi(x)$, the density π only needs to be specified up to a constant of proportionality.

If $q(x, y) = q(y, x)$, *i.e.* q is symmetric, then the acceptance probability $\alpha(x, y)$ simplifies to

$$\alpha(x, y) = \min\left\{\frac{\pi(y)}{\pi(x)}, 1\right\}$$

This is the original form of the algorithm proposed by Metropolis *et al.* (1953). Other forms of the rejection probability are possible, but the form given here can be shown to be optimal within a wide class of possible alternative forms (Peskun, 1973).

The Metropolis algorithm is actually a class of algorithms. Each different choice of the kernel Q for generating candidate steps produces a different version of the algorithm. Several classes of kernels appear to be particularly useful for examining posterior distributions.

2.1.1 Random Walk Chains

For $E = \mathbb{R}^k$ and f a density on E, set $Y = x + Z$, with Z drawn independently from f. Then

$$q(x, y) = f(y - x).$$

Thus the kernel Q driving the Metrolopis chain is a random walk. Natural choices of f are normal, uniform, and t distributions. Split-t distributions (Geweke, 1989) may also be useful. The scale matrix for f can be taken as a constant c times the inverse information at the posterior mode. Good choices for the step size constant c are still an open problem, but $c = 1$ and $c = 1/2$ seem to work reasonably well in a number of examples.

If f is symmetric about the origin, *i.e.* if $f(z) = f(-z)$, then q is symmetric and the simpler form of the acceptance probability $\alpha(x, y)$ can be used.

2.1.2 Independence Chains

Suppose f is a density on E, and we generate candidates Y independently from the single density f. Then

$$q(x, y) = f(y).$$

The chain of candidates driving this Metropolis chain is an *i.i.d.* sequence from the density f. The acceptance probability for an independence chain can be written as

$$\alpha(x, y) = \min\left\{\frac{w(y)}{w(x)}, 1\right\}$$

for $w(x) = \pi(x)/f(x)$. The function w is the weight function that would be used in importance sampling when the sample is generated from the density f.

There are a number of similarities between an independence chain and the corresponding importance sampling process. While an independence chain does not require explicit use of the weights, it will rarely accept candidates with low weights. On the other hand, a candidate with high weight will almost always be accepted. Furthermore, when the chain reaches a point x with high weight $w(x)$, it will usually remain there for several iterations, thus building up weight on x within the sample path by repetition. Another similarity to importance sampling is that the sample sequence is closer to an *i.i.d.* sequence from π the closer the weight function w is to a constant.

Because of these similarities to importance sampling, it is reasonable to conjecture that guidelines developed for choosing importance sampling densities also apply to choosing densities for driving independence chains. In particular, it is advisable to choose a density with thicker tails than π and thus a bounded weight function

w. Families like the split-t that produce good importance sampling densities are likely to be good choices for independence chains.

2.1.3 Rejection Sampling Chains

An interesting special case of an independence chain occurs when the density f is sampled using rejection sampling. In attempting to use rejection sampling to sample directly from π, we use a density h and a constant c such that, hopefully, $\pi(x) < ch(x)$. If we repeat the process of sampling Z from h and then U uniformly from $[0, ch(Z)]$, until $U < \pi(Z)$, then the final value of Z has density

$$f(x) \propto \pi(x) \wedge ch(x).$$

If we do indeed have $\pi(x) \leq ch(x)$, then f is proportional to π and we obtain an *i.i.d.* sample from π. But it is very difficult to insure that c is large enough for ch to dominate π without choosing c excessively large, leading to an inefficient algorithm with many rejections. And even then without extensive analysis of the tails of h and π we cannot be certain that ch does dominate π.

Fortunately, using this rejection scheme to drive an independence Metropolis chain provides a simple remedy. If we do have $\pi(x) \leq ch(x)$ for all x, then the weight function w is a constant, no candidates are rejected, and the rejection process produces an *i.i.d.* sequence from π that is simply passed through the Metropolis algorithm unchanged. But if ch does not dominate π for some x, then, when the chain reaches such an x, the Metropolis algorithm will occasionally reject candidate steps in order to build up mass on this x to make up for the deficiency in the envelope ch. This introduces some dependence, but insures that the equilibrium distribution of the sample path is π even if the envelope is deficient.

2.2 Combining Strategies

The Gibbs sampler and the Metropolis algorithms described above provide a number of Markov chain strategies. In addition to choosing any one of these strategies and using it in its pure form, it is possible to form hybrid strategies.

Suppose $P_1, \ldots, P_m$ are Markov kernels with invariant distribution π. Two simple ways of combining these kernels is as a mixture or a cycle. In a mixture, probabilities $\alpha_1, \ldots, \alpha_m$ are specified, and at each step one of the kernels is selected according to these probabilities. In a cycle, each kernel is used in turn, and when the last one is used the cycle is restarted.

Both strategies can be used in several ways. For example, a Gibbs sampler can be combined with occasional steps from an independence chain in a mixture or a cycle to "restart" the Gibbs sampler and thus reduce correlations while preserving the equilibrium distribution. As another example, suppose θ can be split into two components (θ_1, θ_2), and direct sampling from $\theta_1|\theta_2$ is possible but direct sampling from $\theta_2|\theta_1$ is not possible. Such a situation is considered by Zeger and Karim (1991). Then "Gibbs steps" for $\theta_1|\theta_2$ can be combined with Metropolis steps for $\theta_2|\theta_1$ in a mixture or a cycle.

3 Some Theoretical Results

Whatever approach is used to produce a Markov chain with invariant distribution π, before the chain can be used with confidence to generate samples for examining π certain theoretical questions need to be addressed. Answers to some of these questions can be obtained using some recent developments in general state space Markov chain theory as described, for example, in Nummelin (1984). This section outlines this approach. A more complete discussion is given in Tierney (1991).

3.1 Convergence

The first question to be addressed is whether the invariant distribution π is also the equilibrium distribution for the chain, *i.e.* whether the distribution of the chain after n iterations converges to π. In discrete state space Markov chain theory, two conditions are needed: irreducibility and aperiodicity. The same is true in general state space theory. Periodicity for general state spaces can be defined in much the same way as for discrete spaces. The concept of irreducibility is a little more complicated, since individual states are usually not hit with positive probability. It is therefore necessary to speak of irreducibility with respect to a measure. In the present context, a natural choice for this measure is π itself. We will therefore say that a Markov chain is π-irreducible if for every set A with $\pi(A) > 0$ the probability of the chain ever entering A is positive for every starting point x of the chain.

Irreducibility and aperiodicity need to be verified for each Markov chain. Some useful sufficient conditions are available for certain Metropolis chains. For example, a random walk chain is π-irreducible and aperiodic if the increment density is positive on a neighborhood of the origin and the density π is positive on all of $\mathbb{R}^k$. An independence chain is π-irreducible and aperiodic if the candidate generation density f is positive whenever the density π is positive.

If a chain with invariant distribution π-irreducible and aperiodic, then it can be shown that the chain must be

tive recurrent and that for π-almost all x,

$$\| P^n(x,\cdot) - \pi(\cdot) \| \to 0$$

re $\| \cdot \|$ denotes the total variation distance and P^n he distribution after n steps of the chain started at

the chain is *Harris recurrent*, then this convergence urs for all x. The definition of Harris recurrence is ıewhat technical, but a simple sufficient condition is ilable that is satisfied by all π-irreducible Metropolis ins and essentially all π-irreducible Gibbs samplers. ı π-irreducible aperiodic Markov chain with invari- distribution π is called *ergodic* if it is aperiodic and itive Harris recurrent.

2 Rates of Convergence

ce we know that the distribution of a chain converges π, the next question is to determine the rate of conver- ıce. The theory presented in Nummelin (1984) pro- ıes several classifications for rates of convergence of odic chains:

egree 2: If a chain is ergodic of degree 2, then

$$n \| P^n(x,\cdot) - \pi(\cdot) \| \to 0$$

for π-almost all x.

eometric: An ergodic chain is geometrically ergodic if $\| P^n(x,\cdot) - \pi(\cdot) \| \leq M(x) r^n$ for some $r < 1$ and some function M with $\int M d\pi < \infty$.

iiform: An ergodic chain is called uniformly ergodic if $\| P^n(x,\cdot) - \pi(\cdot) \| \leq M r^n$ for some $r < 1$ and some constant M.

Uniform ergodicity is the strongest of these forms of ıvergence and it is the easiest form to work with. A essary and sufficient condition for a chain with kernel o be uniformly ergodic is that there exist a probabil- ν, a constant $\beta > 0$ and an integer $n \geq 1$ such that $A) \leq P^n(x, A)$ for all A and x. Using this condition, it ossible to derive a variety of sufficient conditions for iform ergodicity. For example, if $\mu(E) < \infty$ and the ısities q and π are bounded and bounded away from o, then the corresponding Metropolis kernel is uni- mly ergodic. As another example, an independence tropolis kernel is uniformly ergodic if the weight func- n $w(x)$ is bounded.

This condition can also be used to derive conditions uniform ergodicity of hybrid kernels in terms of con- ions on the component kernels. For mixtures the con- ion is particularly simple: if P is uniformly ergodic, then any mixture using P with positive probability is uniformly ergodic. For cycles a slightly more compli- cated condition appears to be needed: if P is used in a cycle and there exists a probability ν and a constant $\beta > 0$ such that $\nu(A) \leq P(x, A)$ for all A and x, then the cycle is uniformly ergodic. This condition is satisfied if P is an independence kernel with a bounded weight func- tion. Combining such a kernel in a mixture or a cycle with any other kernel, such as a Gibbs kernel, therefore insures that the hybrid chain is uniformly ergodic. This provides theoretical support for using occasional inde- pendence "restart" steps together with a Gibbs sampler to improve the properties of the sampler.

3.3 Limiting Behavior of Averages

In Markov chain methods, sample path averages are used to estimate expectations under the distribution π. A law of large numbers and a central limit theorem insure that these estimates converge at reasonable rates. The law of large numbers follows from the ergodic theorem and needs no conditions other than existence of the expecta- tion under π:

Law of Large Numbers. *If P is ergodic with invari- ant distribution π, and $\pi|f| < \infty$, then for any initial distribution*

$$\bar{f}_n = \frac{1}{n} \sum_{i=1}^{n} f(X_i) \to \pi f = \int f(x) \pi(dx)$$

almost surely.

A central limit theorem does appear to require some conditions on the rate of convergence of the chain:

Central Limit Theorem. *If P is ergodic of degree 2 with $\pi P = \pi$, and f is bounded, then for any initial distribution the distribution of*

$$\sqrt{n}(\bar{f}_n - \pi f)$$

converges weakly to a normal distribution with mean zero and variance $\sigma^2(f)$.

Weaker but more complicated sufficient conditions are available. Expressions for the asymptotic variance $\sigma^2(f)$ are available for finite E (Peskun, 1973; Kemeny and Snell, 1976). Other expressions involving certain hitting times are available for general state spaces (Nummelin, 1984). These expressions do not appear to be useful for computing the asymptotic variance.

4 Using a Markov Chain

Once a Markov chain strategy with satisfactory theoretical properties has been selected, it can be used to estimate numerical characteristics or to provide graphical views of features of the posterior distribution.

4.1 Numerical Uses

Using Markov chains for calculating numerical characteristics of a posterior distribution is in principle straight forward: expectations with respect to π can be approximated by sample path averages. There are, however, a number of issues that need to be considered before running a chain.

4.1.1 Choosing a Sampling Plan

The first issue concerns the choice of a sampling plan. There are two extreme approaches. Several authors have proposed that Markov chains should be used to generate n independent realizations from the posterior by using n separate runs, each of length m, and retaining the final states from each chain. The run length m is to be chosen large enough to insure that the chain has reached equilibrium. An alternate approach is to use a single long run, or perhaps a small number of long runs. Experience and theoretical assessments in the simulation literature appear to favor the use of long runs (Bratley *et al.*, 1987, Section 3.1.1; Kelton and Law, 1984). The major drawback of using short runs is that it is virtually impossible to tell when a run is long enough based on such runs. Even using long runs, determining how much of the initial series is affected by the starting state is very difficult, but some literature on the subject is available (Ripley, 1987, Section 6.1). A second drawback of short runs is that it makes inefficient use of the data: only n out of a total of nm data points are used. With a single run of length nm it is possible to use all the data, after possibly discarding a small initial fraction.

A complication that does arise from the dependence in using a single series is that variances of estimates are harder to assess. Again the simulation literature offers several alternatives, such as the use of batch means and time series analysis (Bratley *et al.*, 1987, Chapter 3; Ripley, 1987, Chapter 6). For some purposes it may nevertheless be useful to have an approximate independent sample from the posterior. Using long runs this can be achieved by retaining every r-th point of a sample path. The number r of points to skip in order to produce approximate independence can usually be chosen much smaller than the number m of steps needed to reach approximate equilibrium, since small amounts of correlation are usually much less serious than biases in estimates of means.

4.1.2 Determining the Run Length

Another consideration is to determine the total sample size or run length required for accurate estimates. For an *i.i.d.* sample of size n, the standard deviation of the sample mean of a function $f(\theta)$ is $\sigma/\sqrt{n}$, where σ is the posterior standard deviation of $f(\theta)$. If a preliminary estimate of σ is available, perhaps from an asymptotic analysis, then this can be used to estimate the sample size that would be required in *i.i.d.* sampling. In dependent sampling, observations are generally positively correlated and a larger simple size will be required. If the series is modeled as a first order autoregressive process, then the standard deviation of the sample mean is

$$\frac{\sigma}{\sqrt{n}}\sqrt{\frac{1+\rho}{1-\rho}}$$

where again σ is the posterior standard deviation of $f(\theta)$ and ρ is the autocorrelation of the series $f(X_n)$. A rough estimate of ρ can thus be used to adjust the sample size for dependence in the series.

Instead of determining a fixed sample size in advance, it is also possible to use sequential or batch sequential rules for determining when to stop sampling. Since prior information on the values of the posterior mean and standard deviation is often available form initial analyses, Bayesian sequential methods are a natural choice. Batching can be used to insure that an assumption of normality for batched means is reasonable.

One sequential approach that should be avoided is to plot successive sample means and stop sampling when the means appear to have converged. Since sample means change by increments on the order of $O(n^{-1})$ but errors are of order $O(n^{-1/2})$, this approach will produce sample sizes that are too small. The presence of positive correlations in Markov chain series makes these series appear to have converged even earlier, even though the correlations imply that errors are larger and thus larger sample sizes are required than with *i.i.d.* sampling.

4.1.3 Numerical Issues

Some consideration of numerical stability is needed in using any sampling based method. Expressions used to evaluate log posterior densities obtained by translating mathematical formulas into a computer language are often reasonably stable near the posterior mode but not far away from the posterior mode. This can lead to overflows or, on IEEE hardware, results that are NAN's or INF's. One way to avoid these problems is to carefully

y the formulas for evaluating the log posterior den- and modify them to be numerically stable even for ɜme parameter values. The effort required to do this be considerable. An expedient alternative that is of- ɛffective is to truncate the parameter space to a rea- ıble range that contains essentially all the posterior ɔability and for which the posterior density formula ɹmerically stable. This truncation also often insures a Markov chain used to sample from π is uniformly ıdic and thus improves the behavior of the Markov n estimates.

he need to allow truncation is an important consider- n in developing software for implementing sampling ɜd methods. Subroutines must allow for user sup- d range test functions or allow the results returned the log posterior subroutine to indicate a parameter t is outside of the range.

ı numerical issue that is unique to Markov chain ;hods is the possibility that rounding may introduce orbing states. If this happens, results obtained from a rkov chain method may be meaningless. Again trun- ion away from areas of the state space where such nding may occur can be helpful.

.4 Variance Reduction

with any simulation method, variance reduction tech- ues can often significantly reduce the sample sizes re- red for accurate estimates. Standard variance reduc- ɹ methods such as importance sampling, antithetic iates, conditioning, and control variates (Bratley *et* 1987, Chapter 2; Ripley, 1987, Chapter 5) can be d with any Markov chain method.

mportance sampling can be used as a variance reduc- ɹ method by using a Markov chain with equilibrium tribution f instead of π and then weighting sample ılts with appropriate importance weights. Condition- is often useful in Gibbs samplers, since the assump- ıs required for the Gibbs sampler imply that condi- ıal means or densities of one parameter given the rest usually available. Gelfand and Smith (1990) refer to ɜ use of conditioning as Rao-Blackwellization.

Antithetic variation can be introduced into a Markov ın method by using a Metropolis step in which a can- ate step is obtained by reflecting the current state of chain through a point. If the posterior density is ap- ximately symmetric about this point, then the sam- will be also, and the resulting negative correlations l reduce variances of estimates of linear functions of This technique can also be used to take advantage ıpproximate axial symmetries in a posterior distribu- ɹ.

)ne way to introduce control variates into a Markov chain method is to use the sample path with importance weights to calculate estimates of normal approximations and to correct for the errors in these estimates.

4.1.5 Monitoring Sampler Performance

In using Markov chain methods, it is important to monitor the performance of the samplers to insure that they are not exhibiting any unusual behavior. Gelfand and Smith (1990) propose the use of quantile plots to monitor performance. Monitoring sample paths of estimates is also useful for this purpose, as is monitoring autocorrelations of the parameters. Adaptive time series models may also be useful for determining whether a series exhibits any unusual features.

For Metropolis chains it is also important to keep track of the the number of candidates that are rejected. For an independence chain, the proportion of rejections can be related to the total variation distance between the posterior density π and the candidate generation density f.

By monitoring the performance of a sampler, in particular in the early stages, it is possible to experiment with different setting for sampler parameters to obtain samplers that are efficient for a particular problem. More work is needed to find good strategies for making such parameter adjustments.

4.2 Graphical Uses

Numerical summaries, such as posterior means, standard deviations, marginal densities, and correlations, provide insight into the uncertainty about one or perhaps two features of θ at a time. For understanding uncertainty in higher dimensions graphical methods may be more useful than numerical summaries.

4.2.1 Plotting Samples

For three-dimensional quantities, one useful graphical method available on microcomputers and workstations with bitmapped displays is a rotatable three-dimensional scatterplot. By selecting every r-th entry in a Markov chain sample path we can obtain an approximate *i.i.d.* sample from the posterior distribution and display this sample in a rotatable scatterplot. Three-dimensional structures will readily become apparent as the point cloud of the sample is rotated.

Rotatable scatterplots are only useful for examining three dimensions at a time. A method that may be useful for higher dimensions is the Grand Tour. Again an approximate *i.i.d.* sample can be selected and displayed in a Grand Tour. Implementations of the Grand Tour

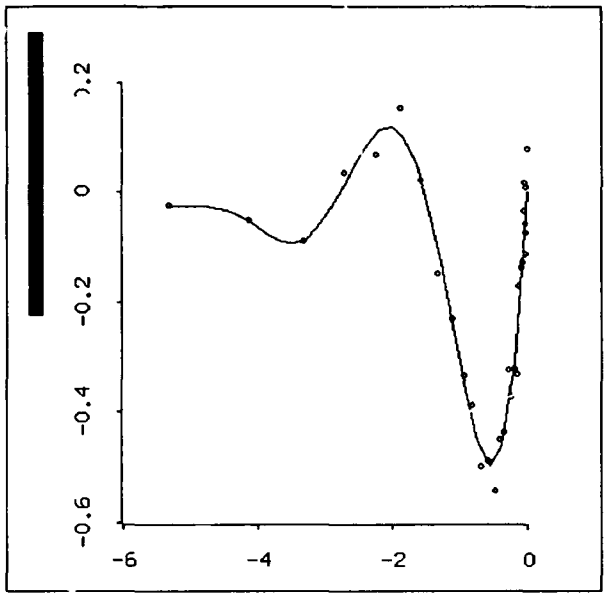


Figure 1: Posterior mean of a response function.

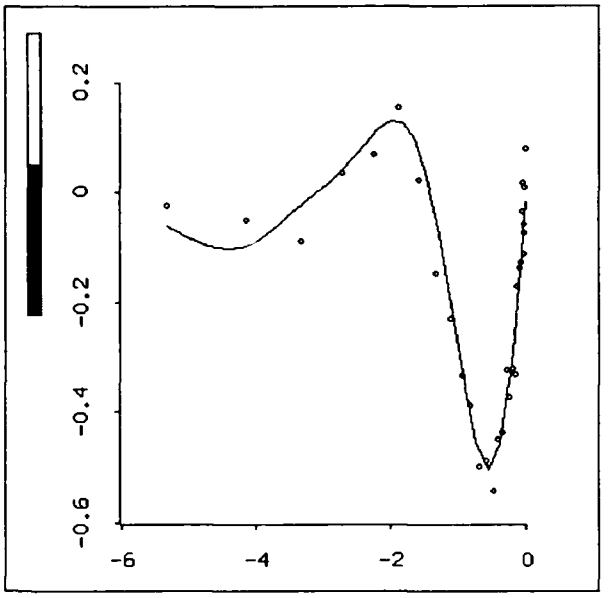


Figure 2: A second response function supported by the posterior distribution.

are only now becoming widely available, so extensive experience with this method is not yet available. Early results suggest that this method is reasonably effective for detecting structures in four to six dimensions.

4.2.2 Controlling Animations

If θ is more than five- or six-dimensional, then it may be difficult enough to understand θ itself, much less uncertainty about θ. If a graphical view of θ is available that is meaningful for particular values of θ, then one way of developing an understanding of the uncertainty about θ is to look at an animated version of the graph in which θ is moved through a variety of values that are plausible under the posterior distribution.

As an example, suppose we have a smooth response function θ of a real variable x in some interval I that is measured with error. Thus we obtain measurements of the form

$$Y = \theta(x) + \epsilon.$$

Our prior opinion on the function θ suggests that this function is smooth, but does not suggest any particular parametric structure.

Several approaches are available for specifying such a prior distribution. Most involve choosing a prior on coefficients in some representation, such as a power series or spline. The coefficients of these representations are not likely to be particularly meaningful. But a plot of the response function θ over the interval I is readily understood. Figure 1 shows a plot of the posterior mean of θ for a particular example. This mean exhibits a number of features, such as a pronounced global minimum and a secondary local minimum. Are these features really present in θ or are they merely artifacts of the posterior mean? One way to answer this question is to look at other functions θ that are supported by the posterior distribution. This can be done by running an animation that shows graphs of different values of θ.

To provide a good understanding of the posterior distribution, an animation needs to visit all areas supported by the posterior. In addition, to allow the user to keep track of the changes in θ as it moves through the posterior distribution, the animation has to move smoothly. These objectives can be achieved using a random walk-driven Metropolis chain with the posterior distribution as its equilibrium distribution. Using the posterior as the equilibrium insures that the chain does eventually approach all possible values of θ but spends most of its time near values that are better supported by the posterior distribution. The correlation in the random walk insures that the chain moves in small steps, thus providing the visual continuity that is necessary for an effective animation. Thus the correlations in the Metropolis chain that are a nuisance for numerical computations are in fact an advantage for this graphical application. Continuity can be further enhanced by interpolating between steps of the random walk.

Figure 2 shows another view of the animation. Viewing the animation for this particular example for a few

minutes quickly reveals that the global minimum is quite well defined but the shape of the left half of the curve is very uncertain.

A useful enhancement for this animation is the bar shown at the left of the two plots. The solid part of the bar represents the probability content in the posterior at or below the level of the current θ, computed using a χ^2 approximation. This gives a quick indication of how plausible the current view is.

Many variations on this animation are possible. For example, using the posterior distribution as the equilibrium of the driving Markov chain is a reasonable starting point but is not essential. At times it may be useful to force the chain to concentrate its motion closer to the mode, or to move farther away from the mode and possibly find interesting features that are farther away. This can be accomplished by using a Markov chain with an equilibrium density that is a power of the posterior density – by "cooling" or "heating" the posterior distribution in the terminology of simulated annealing.

Much additional work is needed to explore ways of merging numerical methods such as the ones described in this paper with new computing hardware that is now becoming more widely available. The animation described here is a first step in that direction.

References

Bratley, P., B. L. Fox, and L. E. Schrage (1987). *A Guide to Simulation*. New York, NY: Springer, second edition.

Gelfand, A. E. and A. F. M. Smith (1990). Sampling based approaches to calculating marginal densities. *J. Amer. Statist. Assoc.* **85** 398–409.

Geman, S. and D. Geman (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **6** 721–741.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* **57** 1317–1339.

Hammersley, J. M. and D. C. Handscomb (1964). *Monte Carlo Methods*. London: Chapman and Hall.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57** 97–109.

Kelton, D. W. and A. M. Law (1984). An analytical evaluation of alternative strategies in steady-state simulation. *Operations Research* **32** 169–184.

Kemeny, J. G. and J. L. Snell (1976). *Finite Markov Chains*. New York, NY: Springer.

Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller (1953). Equations of state calculations by fast computing machines. *J. Chemical Physics* **21** 1087–1091.

Nummelin, E. (1984). *General Irreducible Markov Chains and Non-Negative Operators*. Cambridge: Cambridge University Press.

Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika* **60** 607–612.

Ripley, B. D. (1987). *Stochastic Simulation*. New York, NY: Wiley.

Stewart, L. T. (1979). Multiparameter univariate Bayesian inference. *J. Amer. Statist. Assoc.* **74** 684–693.

Tierney, L. (1991). Markov chains for exploring posterior distributions. Technical Report 560, School of Statistics, University of Minnesota.

van Dijk, H. K., J. P. Hop, and A. S. Louter (1978). An algorithm for the computation of posterior moments and densities using simple importance sampling. *The Statistician* **36** 83–90.

Zeger, S. L. and M. R. Karim (1991). Generalized linear models with random effects: A Gibb's sampling approach. *J. Amer. Statist. Assoc.* **86** 79–86.

Zellner, A. and P. E. Rossi (1984). Bayesian analysis of dichotomous quantal response models. *J. of Econometrics* **25** 365–393.

Efficient Simulation from the Multivariate Normal and Student-t Distributions Subject to Linear Constraints

John Geweke *
Department of Economics
University of Minnesota
Minneapolis, MN 55455
geweke@atlas.socsci.umn.edu

Abstract

The construction and implementation of a Gibbs sampler for efficient simulation from the truncated multivariate normal and Student-t distributions is described. It is shown how the accuracy and convergence of integrals based on the Gibbs sample may be constructed.

KEYWORDS: Bayesian inference; Gibbs sampler; Monte Carlo; multiple integration; truncated normal

1 Introduction

The generation of random samples from a truncated multivariate normal distribution, that is, a multivariate normal distribution subject to multiple linear inequality restrictions, is a recurring problem in the evaluation of integrals by Monte Carlo methods in econometrics and statistics. Sampling from a truncated multivariate Student-t distribution is a closely related problem. The problem is central to Bayesian inference, where a leading example is the normal linear regression model subject to linear inequality restrictions on the coefficients (Geweke, 1986). But it also arises in classical inference, when integrals enter the likelihood function; McFadden (1989) has proposed the use of Monte Carlo integration in one such instance.

Recently several promising solutions of this problem have been investigated. A survey of methods along with several contributions is provided in Hajivassiliou and McFadden (1990). One of these methods, the Gibbs sampler, is especially well suited to the problem. It uses a simple algorithm that generates samples with great computational efficiency, but at the cost of introducing two complications. First, the drawings are not independent, which complicates the evaluation of the accuracy of the approximation using standard methods like those proposed in Geweke (1989). Second, the distribution from which the drawings are made converges to the truncated multivariate normal distribution, but is not identical with it at any stage.

In this paper we contribute to the resolution of these problems. The ability to generate variates from a truncated univariate normal distribution is a central building block in the solution of the more general problem. Section 2 describes an algorithm for the generation of variates from a truncated univariate normal distribution that is substantially more efficient and flexible than the method that has been favored in the past. Drawing from the truncated multivariate normal distribution with the Gibbs sampler is fully developed in Section 3, including the evaluation of the accuracy of the numerical approximation and construction of diagnostics for convergence. These methods are extended to the multivariate Student-t distribution in Section 4.

Throughout the paper some standard but not universal notation is employed. The univariate normal probability density function is $\phi(\cdot)$, the corresponding cumulative distribution function is $\Phi(\cdot)$, and the inverse cumulative distribution function is $\Phi^{-1}(\cdot)$. The uniform distribution on the interval [a, b] is denoted U[a, b]. The univariate truncated normal distribution TN(a, b) is the univariate normal restricted to (a, b): its density is $[\Phi^{-1}(b) - \Phi^{-1}(a)]^{-1}\phi(\cdot)$ on (a, b) and 0 elsewhere; $a = -\infty$ and $b = +\infty$ are permitted special cases.

2 The mixed rejection algorithm for truncated univariate normal sampling

All of the methods described in this paper assume the ability to draw i.i.d. samples from a truncated univariate normal distribution. It is well recognized that rejection sampling from a univariate normal distribution is impractical. Inverse c.d.f. sampling (Devroye, 1986) is a feasible alternative. If $x \sim TN(a, b)$, then $x = \Phi^{-1}(u)$, $u \sim U[\Phi(a), \Phi(b)]$. This method requires the evaluation of one integral for each draw, and if the values of a and b change with the draws, then three evaluations are required. The computation of $\Phi^{-1}(w)$ requires more time as $w \to 0$ or $w \to 1$, and the double precision implementation in the IMSL/STAT library is unable to compute $w = \Phi^{-1}(p)$ if $|w| > 8$. Here, we shall suggest a different algorithm, whose

* Research assistance from Zhenyu Wang and financial support from National Science Foundation Grant SES-8908365 are gratefully acknowledged. The software for the examples may be requested by electronic mail, and will be returned by that medium.

execution times are substantially smaller than inverse c.d.f. sampling, and can draw $x \sim TN(a, b)$ for any $a < b$ so long as $|a| \le 35$ and $|b| \le 35$, when programmed in double precision (64-bit) floating point arithmetic.

The algorithm produces i.i.d. samples from TN(a, b), including the cases $a = -\infty$ and $b = +\infty$. It employs four different kinds of rejection sampling, depending on the values of a and b. In *normal rejection sampling*, x is drawn from N(0, 1) and accepted if $x \in [a, b]$. In *half-normal rejection sampling*, x is drawn from N(0, 1) and $|x|$ is accepted if $x \in [a, b]$ (where $a \ge 0$). In *uniform rejection sampling*, x is drawn from U[a, b], u is drawn independently from U(0,1), and x is accepted if $u \le \phi(x)/\phi(x^*)$, $x^* = \text{argmax}_{[a,b]}[\phi(x)]$.

Exponential rejection sampling is key to the algorithm, and requires description in some detail. The motivating example is $TN(a, \infty)$, where $a > 0$, and possibly $\Phi(a)$ is close to 1. As $a \to \infty$, the $TN(a, \infty)$ distribution comes to resemble the exponential distribution as detailed in Geweke (1986, Appendix A). Suppose z is drawn from an exponential distribution on $[a, \infty)$ with kernel $\exp(-\lambda z)$ for $z \ge a$. Consider fixing λ so as to minimize the probability of rejection. The acceptance probability must be proportional to $\exp(-\frac{1}{2}z^2)/\exp(-\lambda z)$, for $z \in [a, \infty)$. Computing the constants of proportionality, we find acceptance probabilities

$$\exp[-\tfrac{1}{2}(z^2 + a^2)]\exp(-\lambda z) \text{ if } \lambda \le a,$$
$$\exp[-\tfrac{1}{2}(z - \lambda)^2)] \text{ if } \lambda \ge a.$$

The first expression is maximized at $\lambda = a$ for all z. Integrating the second expression with respect to the exponential density $\lambda\exp[-\lambda(z-a)]$ on $[a, \infty)$, we find that the acceptance probability is $\lambda \exp(\lambda a - \frac{1}{2}\lambda^2)(2\pi)^{1/2}[1 - \Phi(a)]$. This is maximized when $\lambda = \frac{1}{2}[a + (a^2 + 4)^{1/2}]$. As $a \to \infty$, $\lambda/a \to 1$, and the acceptance probability converges to unity. Experimentation within the context of the algorithm presently described has shown that the increase in computing time from using the suboptimal but simpler choice $\lambda = a$, is less than the time required to compute $\frac{1}{2}[a + (a^2 + 4)^{1/2}]$. Hence we use $\lambda = a$ in this algorithm.

The algorithm employs four constants (t_i, $i = 1, .., 4$) whose values have been set through experimentation with computation time. The selected value is indicated when the constant is introduced. The sampling procedure depends on the relative configuration of a and b, as follows. Except in case (1), a and b are finite.

(1) On (a, ∞): normal rejection sampling if $a \le t_4$ (= .45); exponential rejection sampling if $a > t_4$.

(2) On (a, b) if $0 \in [a, b]$:
   (a) If $\phi(a) \le t_1$ (= .150) or $\phi(b) \le t_1$: normal rejection sampling;
   (b) If $\phi(a) > t_1$ or $\phi(b) \ge t_1$: uniform rejection sampling.

(3) On (a, b) if $a > 0$:
   (a) If $\phi(a)/\phi(b) \le t_2$ (= 2.18): uniform rejection sampling;
   (b) If $\phi(a)/\phi(b) > t_1$ and $a < t_3$ (= .725): half-normal rejection sampling;
   (c) If $\phi(a)/\phi(b) > t_1$ and $a \ge t_3$: exponential rejection sampling.

The omitted cases $(-\infty, b)$, and (a, b) with $b < 0$, are symmetric to the cases (1) and (3), respectively, and are treated in the same way. Software for the mixed rejection algorithm was tested by comparing the distributions of sampled variates produced, with those produced by inverse c.d.f. sampling. Each was programmed in double precision Fortran-77 using the IMSL/STAT library, on a Sun Sparcstation 4/40 (IPC). Computation times for 10,000 sampled variates are shown in Table 1. Times for the inverse c.d.f. algorithm range from 2.24 to 4.51 seconds, those for the mixed rejection algorithm from 0.67 to 1.28 seconds. On a case-by-case basis the mixed rejection algorithm is from 2.47 to 6.24 times faster than the inverse c.d.f. algorithm.

3 The Gibbs algorithm for truncated multivariate normal sampling

The central problem addressed in this paper is the construction of samples from an n-variate normal distribution subject to linear inequality restrictions,

$$x \sim N(\mu, \Sigma), \quad a \le Dx \le b \qquad (3.1)$$

The matrix D is n x n of rank n, individual elements of a may be $-\infty$, and individual elements of b may be $+\infty$. This accommodates fewer than n linearly independent restrictions. It does not allow more than n linearly independent restrictions, and the method set forth here cannot be extended to these cases, at least in a tidy way. In the applications described in the introduction the truncated multivariate normal distribution arises in the form (3.1). The problem is equivalent to the construction of samples from the n-variate normal distribution subject to linear restrictions,

$$z \sim N(0, T), \quad \alpha \le z \le \beta, \qquad (3.2)$$

where

$$T = D\Sigma D', \quad \alpha = a - D\mu, \quad \beta = b - D\mu,$$

and we then take $x = \mu + D^{-1}z$.

Several approaches to the solution are possible; see Hajivassiliou and McFadden (1990, Appendix B) for a brief survey of these methods, and Hajivassiliou, McFadden, and Ruud (1990) for an application of importance sampling to the special case of orthant restrictions. Naive rejection

Table 1
Comparison of Computation Times
Mixed Rejection and Inverse c.d.f. Algorithms
TN[a, b] Distribution*

a: / b:	-8.0	-5.0	-3.0	-2.0	-1.0	-0.5	0.0	0.5	1.0	2.0	3.0	5.0
-5.0	1.02 4.52											
-3.0	1.04 4.45	1.04 4.45										
-2.0	1.07 4.44	1.10 4.43	1.08 4.49									
-1.0	1.28 3.65	1.26 3.67	1.22 3.66	1.21 3.62								
-0.5	1.19 3.57	1.19 3.69	1.25 3.60	1.26 3.55	.93 2.90							
0.0	1.16 2.91	1.16 2.91	1.15 2.91	1.19 2.89	.75 2.25	.71 2.33						
0.5	.89 3.55	.90 3.54	.90 3.58	.91 3.61	.78 2.92	.73 2.92	.67 2.24					
1.0	.76 3.54	.76 3.52	.78 3.53	.79 3.51	.82 2.89	.79 2.90	.73 2.24	.89 2.90				
2.0	.68 4.15	.69 4.16	.70 4.16	.70 4.14	.97 3.56	1.02 3.52	1.23 2.90	1.21 3.56	1.18 3.63			
3.0	.71 4.24	.69 4.18	.69 4.19	.69 4.18	.90 3.57	1.01 3.65	1.19 2.94	1.18 3.75	1.18 3.71	1.05 4.51		
5.0	.69 4.22	.68 4.15	.69 4.17	.69 4.15	.89 3.52	.98 3.54	1.21 2.92	1.15 3.58	1.23 3.64	1.05 4.43	1.02 4.45	
8.0	.67 4.18	.68 4.15	.69 4.17	.69 4.13	.89 3.53	1.01 3.54	1.17 1.91	1.15 3.56	1.24 3.63	1.04 4.42	1.01 4.45	.97 4.47
a:	-8.0	-5.0	-3.0	-2.0	-1.0	-0.5	0.0	0.5	1.0	2.0	3.0	5.0

* Times are given in seconds, for drawing samples of size 10,000. Computations were performed on a Sun 4/40 (IPC) workstation. Software was written in double precision Fortran-77, and used the IMSL/STAT Edition 10 routines DRNNOF for univariate normal generation, DNORDF for evaluation of the univariate normal c.d.f., and DNORIN for evaluation of the univariate normal inverse c.d.f.

sampling from $N(\mu, \Sigma)$ can be employed directly in (3.1), but is impractical in general since the ratio of rejected to accepted variates is astronomical for many commonly arising problems. More sophisticated procedures must cope with the fact that the marginal distributions of the elements of z, and of x, are not univariate truncated normal. The method set forth here exploits the fact that the distribution of each element of z, *conditional* on all of the other elements of z, is truncated normal. This method has also been described by Hajivassiliou and McFadden (1990), but as outlined in the introduction we pursue several extensions here.

The algorithm employed is the Gibbs sampler, whose systematic application to problems of this form dates from Geman and Geman (1984); see also Gelfand and Smith (1990). The general problem is to sample from a multivariate density f(x) for an n-dimensional random vector x, when no practical algorithm is available for doing so directly. But suppose that the conditional distributions,

$$x_i \mid \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\} \sim f_i(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \quad (i = 1, \ldots, n)$$

are known, and are of a form that synthetic i.i.d. random variables can be generated readily and efficiently from each of

the $f_i(\cdot)$. Let $x^{0\prime} = (x_1^0, \ldots, x_n^0)$ be an arbitrary point in the support of $f(x)$. Generate successive synthetic random variables,

$$x_i^1 \mid \{x_1^1, \ldots, x_{i-1}^1, x_{i+1}^0, \ldots, x_n^0\} \sim f_i(x_1^1, \ldots, x_{i-1}^1, x_{i+1}^0, \ldots, x_n^0) \quad (i = 1, \ldots, n) \tag{3.3}$$

These n *steps* constitute the first *pass* of the Gibbs sampler. The second and successive passes are performed similarly. At the i'th step of the j'th pass,

$$x_i^j \mid \{x_1^j, \ldots, x_{i-1}^j, x_{i+1}^{j-1}, \ldots, x_n^{j-1}\} \sim f_i(x_1^j, \ldots, x_{i-1}^j, x_{i+1}^{j-1}, \ldots, x_n^{j-1}),$$

and the composition of the vector becomes

$$x^{(j,i)\prime} = (x_1^j, \ldots, x_i^j, x_{i+1}^{j-1}, \ldots, x_n^{j-1})'$$

at the end of this step. At the end of the j'th pass the composition of the vector is

$$x^{(j)\prime} = (x_1^j, \ldots, x_n^j)'.$$

Gelfand and Smith (1990) have outlined weak conditions under which $x^{(j)}$ converges in distribution and has limiting distribution give by the density $f(x)$, and the rate of convergence is geometric in the L_1 norm. These conditions pertain to the truncated multivariate normal density in (3.2). The conditional densities $f_i(\cdot)$ for this problem are truncated univariate normal, and the algorithm described in the previous section may be used to generate the required successive synthetic random variables. Suppose that in the non-truncated distribution N(0, T),

$$E[z_i \mid z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n] = \sum_{j \neq i} c_{ij} z_j.$$

Then in the truncated normal distribution of (3.2), the distribution of z_i conditional on $\{z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n\}$ has the construction,

$$z_i = \sum_{j \neq i} c_{ij} z_j + h_i \varepsilon_i,$$

$$\varepsilon_i \sim \mathrm{TN}[(\alpha_i - \sum_{j \neq i} c_{ij} z_j)/h_i, \ (\beta_i - \sum_{j \neq i} c_{ij} z_j)/h_i].$$

Denote the vectors of coefficients in the conditional means,

$$c^i = (c_{i1}, \ldots, c_{i,i-1}, c_{i,i+1}, \ldots, c_{in})' \quad (i = 1, \ldots, n).$$

From the conventional theory for the conditional multivariate normal distribution (Rao, 1965, p. 441) and expressions for the inverse of a partitioned symmetric matrix (Rao, 1965, p. 29),

$$c^i = -(T^{ii})^{-1} T^{i,<i}, \quad h_i^2 = (T^{ii})^{-1},$$

where T^{ii} is the element in row i and column i of T^{-1}, and $T^{i,<i}$ is row i of T^{-1} with T^{ii} deleted. These computations need only be performed once, before sampling begins. An initial value $z^{(0)}$ may be selected by setting $z = 0$ and then successively applying (3.3) for $i = 1, \ldots, n$. At the end of each pass we compute $x^{(j)} = \mu + D^{-1} z^{(j)}$.

Samples from the truncated multivariate normal distribution are typically used to estimate the expected value of a function $g(\cdot)$ of the random vector x,

$$\bar{g} = \int_X g(x) f(x) dx.$$

An assessment of the reliability of this estimate must take into account the facts that in general $\{x^{(j)}\}$ is a serially correlated process, whose unconditional distribution converges to $f(\cdot)$ rather than being identical with $f(\cdot)$. These problems are taken up for the general case in Geweke (1991), where standard spectral analytic techniques are used to produce diagnostics for the convergence of the sampled distributions to $f(\cdot)$ and to provide a numerical standard error for the reliability of the estimated expected value. Using this approach, five statistics from the sample $\{x^{(j)}\}_{j=1}^p$ provide information about the expected value of the function in question.

(1) The simple arithmetic *mean* $\bar{g}_p = p^{-1} \sum_{j=1}^p g(x^{(j)})$ is the most efficient estimate of $\bar{g}$ from a Gibbs sample of size p passes, assuming that departures from convergence in the sample are negligible. The function $g(x)$ is computed at the end of each pass, following the transformation from z to x.

(2) The sampling variance of $\bar{g}_p$ is $S_g(0)/p$, where $S_g(\omega)$ denotes the spectral density of the Gibbs-sampled $g(x)$ process at frequency ω. The *numerical standard error (NSE)* of $\bar{g}_p$ is $[p^{-1} \hat{S}_g(0)]^{1/2}$, where $\hat{S}_g(\omega)$ is a consistent (in p) estimator of $S_g(\omega)$.

(3) The variance of $g(\cdot)$ is estimated in the same way as the mean of $g(\cdot)$. The ratio of this variance to $S_g(0)$ indicates the ratio of the number of i.i.d. draws that would have been required, were such an algorithm available, to the number of passes required with the Gibbs sampler, to produce an estimate of $\bar{g}$ of equivalent reliability. Following Geweke (1989), this ratio is called the *relative numerical efficiency (RNE)* of the Gibbs sampling procedure.

(4) A *convergence diagnostic (CD)* is computed based on subsamples of the sampled $g(x)$; see Geweke (1991, Section 3.2) for details. Under a stationary distribution for $\{x^{(j)}\}_{j=1}^p$ this statistic has a standard normal distribution.

(5) The spectral density provides further details on the characteristics of the process $\{g(x^{(j)})\}_{j=1}^{p}$. If the spectral density is nearly flat, or is lower near $\omega = 0$ than at other frequencies, then the Gibbs sampling process is efficient relative to i.i.d. sampling. But if the spectral density is much higher near $\omega = 0$ than elsewhere, the process is inefficient. Thus, there is a correspondence between the shape of the spectral density and the RNE of the Gibbs sampler.

The Gibbs algorithm was programmed in double precision Fortran-77 using the IMSL/STAT library, on a Sun Sparcstation 4/40 (IPC). The routine was tested by comparing the distribution of truncated normal samples with those generated by a naive accept/reject procedure. To provide some indication of the efficiency of the procedure, we present two examples here.

The first example is a truncated bivariate normal, with parameters chosen so that convergence ought to be especially slow. Both variables have mean zero. The variance of x_1 is 10, while the variance of x_2 is 0.1, and the restrictions are of the form $a_1 \le x_1 + x_2 \le b_1$, $a_2 \le x_1 - x_2 \le b_2$. Consequently the transformed variables z_1 and z_2 have correlation .98. As elaborated in Geweke (1991), this implies that the process $z^{(j)}$, and hence $x^{(j)}$, will exhibit strong positive serial correlation: e.g., if $a_i = -\infty$ and $b_i = +\infty$, then each element of $z^{(j)}$ will follow a first order autoregressive process with parameter .96. Results are presented in Table 2, which shows the five statistics for five different configurations of truncation points (a_i, b_i), and for three choices of the number of passes, p = 400, 2000, or 10,000. In each case p preliminary passes were performed before the functions of interest $g_i(x)$ ($i = 1, \ldots, 4$) were computed and averaged over the next p passes. Computation times varied about 20% depending on the (a_i, b_i) configurations, averaging about .35 seconds for $p = 400$ and 7.1 seconds for p = 10.000.

The results, presented in Table 2, confirm that convergence is slow for the untruncated normal distribution, panel A. (This is presented as a limiting case; obviously Gibbs sampling is not the method of choice for this problem.) Even when p = 10,000, results are unreliable, as indicated by the convergence diagnostics. The problem arises from the strong serial correlation in the processes $\{g(x^{(j)})\}$, which is not fully evident in the estimated spectral densities for the smaller values of p; correspondingly, computed RNE falls as p increases. These results persist in the second case, in which z_2 is truncated at about 1.5 standard deviations above and below (Panel B), but are not so strong. In both cases the convergence diagnostic is an imperfect indicator of unreliable estimates of $\bar{g}$, for there are several cases in which $\bar{g}_p$ is more than three times NSE from 0 (the known true value of $\bar{g}$ in all cases except D) and yet CD is less than 1.5 in absolute value. In the third case z_2 is truncated at about .15 standard deviations above and below (Panel C), and performance is satisfactory for all values of p. The same is true in the fourth case, in which the bivariate normal distribution is truncated to an extreme tail in both dimensions (Panel D), and in the fifth case, in which the the truncation produces a distribution closer to uniform than to bivariate normal (Panel E). Severe truncation diminishes the potential for strong serial correlation in the $x^{(j)}$, and thereby increases the efficiency of the Gibbs sampler.

The second example is constructed to resemble the truncated multivariate normal distribution that might be encountered in Bayesian inference with a multivariate probit model with panel data and serial correlation in equation disturbances for the same sampling unit and different years. Assuming three choices, five years, and a first-order autoregressive process for the disturbance leads to a variance matrix $\Sigma = R \otimes I_3$, $r_{ij} = \rho^{|i-j|}$ in a 15-variate normal with truncation restrictions that require one of x_{3j+1}, x_{3j+2}, and x_{3j+3} to be greater than the other two, for $j = 1, \ldots, 5$. Results are presented in Table 3 for four different values of ρ, ranging from $\rho = .00$ to $\rho = .95$. The number of passes and preliminary passes are the same as those in the previous example, and computation times range from 2.5 seconds for $p = 400$ to about 60 seconds for $p = 10{,}000$. As ρ increases, serial correlation in $z_{(j)}$ and hence $x_{(j)}$ increases, diminishing the efficiency and reliability of the Gibbs sampling algorithm. The convergence diagnostic proves to be a reliable indicator of the reliability of the estimates $\bar{g}_p$. For $\rho = .00$, 400 passes are reliable, despite some modest serial correlation; for $\rho = .50$, 2000 passes are required, and for $\rho = .80$, 10,000 passes are required. For $\rho = .95$, even 10,000 passes do not produce reliable results.

4 The Gibbs algorithm for truncated multivariate Student-t sampling

A closely related problem arising in Bayesian inference is the generation of samples from the multivariate Student-t distribution subject to linear restrictions,

$$x \sim T(\mu, \Sigma; m), \quad a \le Dx \le b.$$

We continue to make the same assumptions about a, b, and D. The genesis of the multivariate Student-t as the ratio of a multidimensional normal to an independent $[\chi^2(m)/m]^{1/2}$ leads immediately to a Gibbs sampling algorithm for $(w, z_1, \ldots, z_n)$ followed by the construction $x = \mu + D^{-1}zw^{-1}$.

At the start of pass j, $w^{(j-1)}$ and $z^{(j-1)}$ are available from the previous pass. In the first step draw $w^{(j)} \sim [\chi^2(m)/m]^{1/2}$ subject to the restrictions

$$\alpha_i w^{(j)} \le z_i^{(j-1)} \le \beta_i w^{(j)} \qquad (i = 1, \ldots, n),$$

using an accept/reject procedure. In steps 2, ... , n+1, draw z from a multivariate normal distribution conditional

on $w^{(j)}$, the pertinent z's, and the restrictions $\alpha_i w^{(j)} \leq z_i^{(j)} \leq \beta_i w^{(j)}$:

$$z_i^{(j)} = \sum_{j=1}^{i-1} c_{ij} z_i^{(j)} + \sum_{j=i+1}^{n} c_{ij} z_i^{(j-1)} + h_i \varepsilon_i,$$

$$\varepsilon_i \sim TN[(\alpha_i w_{(j)} - \sum_{j=1}^{i-1} c_{ij} z_i^{(j)} - \sum_{j=i+1}^{n} c_{ij} z_i^{(j-1)})/h_i,$$

$$(\beta_i w^{(j)} - \sum_{j=1}^{i-1} c_{ij} z_i^{(j)} - \sum_{j=i+1}^{n} c_{ij} z_i^{(j-1)})/h_i)].$$

At the end of the pass, $x^{(j)} = \mu + D^{-1} z^{(j)} w^{(j)}$.

This algorithm was programmed in double precision Fortran-77 using the IMSL/STAT library. The routine was tested by comparing the distribution of truncated Student-t samples with those generated by a naive accept/reject procedure. No appreciable increases in computation time over corresponding problems with the truncated multivariate normal distribution were n-oted. In particular, the accept/reject procedure for $w^{(j)}$ appears quite efficient, even for $m = 2$. No considerations with respect to the efficiency of the Gibbs sampling algorithm, beyond those for the multivariate normal, have been noted.

References

Devroye, L., 1986: *Non-Uniform Random Variate Generation*. New York: Springer-Verlag.

Geman, S., and D.J. Geman, 1984: "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 721-741.

Gelfand, A.E., and A.F.M. Smith, 1990: "Sampling Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association* 85, 398-409.

Geweke, J., 1986: "Exact Inference in the Inequality Constrained Normal Linear Regression Model," *Journal of Applied Econometrics* 1, 127-142.

Geweke, J., 1989: "Bayesian Inference in Econometric Models Using Monte Carlo Integration," *Econometrica* 57, 1317-1339.

Geweke, J., 1991: "Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments," University of Minnesota manuscript. Presented at the Fourth Valencia International Meeting on Bayesian Statistics.

Hajivassiliou, V.A., and D.L. McFadden, 1990: "The Method of Simulated Scores for the Estimation of LDV Models with an Application to External Debt Crises," Colwes Foundation working paper, Yale University.

Hajivassiliou, V.A., D.L. McFadden, and P.A. Ruud, 1990: "Simulation of Multivariate Normal Orthan Probabilities: Methods and Programs," M.I.T. mimeo.

McFadden, D., 1989: "A Method of Simulated Moments for Estimation of Discrete Response Models Without Numerical Integration," *Econometrica* 57.

Rao, C.R., 1965: *Linear Statistical Inference and Its Applications*. New York: Wiley.

Table 2
Properties of the Gibbs Sampler for a Truncated Bivariate Normal Distribution
$\mu_1 = \mu_2 = 0, \quad \sigma_{11} = 0.1, \quad \sigma_{22} = 10, \quad \sigma_{12} = 0$

A: $-\infty < x_1 + x_2 < \infty, \quad -\infty < x_1 - x_2 < \infty$

	$g_1(x) = x_1$			$g_2(x) = x_2$			$g_3(x) = x_1 + x_2$			$g_4(x) = x_1 - x_2$		
p	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000
Mean	-.0093	-.0030	-.0048	.6207	.5295	.4648	.6113	.5265	.4600	-.6301	-.5325	-.4695
NSE	.0172	.0068	.0027	.4515	.2461	.1516	.4522	.2463	.1518	.4514	.2462	.1515
RNE	.932	1.077	1.312	.170	.085	.042	.171	.085	.043	.170	.086	.043
CD	-12.539	.669	-2.319	-12.550	.669	-2.319	-12.539	.715	-2.342	12.453	-.620	2.293
$S_g(0)$	.1164	.0926	.0733	80.31	120.37	229.24	80.57	120.48	229.74	80.28	120.44	228.90
$S_g(\pi/2)$	.0941	.1104	.0999	.3610	.3651	.3002	.4556	.4491	.3911	.4548	.5018	.4092

B: $-\infty < x_1 + x_2 < \infty, \quad -5 < x_1 - x_2 < 5$

	$g_1(x) = x_1$			$g_2(x) = x_2$			$g_3(x) = x_1 + x_2$			$g_4(x) = x_1 - x_2$		
p	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000
Mean	.0071	.0094	-.0016	.9331	.5474	.03667	.9402	.5568	.0351	-.9261	-.5380	-.0383
NSE	.0139	.0065	.0028	.2921	.1737	.1079	.2947	.1749	.1087	.2901	.1728	.1072
RNE	1.105	1.197	1.242	.177	.091	.051	.180	.093	.052	.179	.092	.052
CD	-.808	.927	-2.288	-.111	2.605	-.959	-.177	2.653	-1.041	.044	-2.544	.875
$S_g(0)$	.0761	.0083	.0079	33.607	59.950	116.023	34.220	60.743	117.705	33.148	59.321	114.499
$S_g(\pi/2)$	.0886	.1097	.1019	.3438	.3619	.2894	.4561	.5057	.4164	.4087	.4376	.3663

C: $-\infty < x_1 + x_2 < \infty, \quad -.5 < x_1 - x_2 < .5$

	$g_1(x) = x_1$			$g_2(x) = x_2$			$g_3(x) = x_1 + x_2$			$g_4(x) = x_1 - x_2$		
p	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000
Mean	.0282	-.0018	-.0057	.0445	-.0080	-.0135	.0727	-.0098	-.0192	-.0162	.0061	.0078
NSE	.0150	.0065	.0032	.0272	.0113	.0056	.0405	.0169	.0085	.0170	.0073	.0035
RNE	1.060	1.125	.953	.638	.667	.573	.726	.782	.665	.796	.771	.688
CD	-.934	.251	-.963	-.793	.667	.573	-.902	.782	.665	.265	.341	-.352
$S_g(0)$	.0889	.0833	.1050	.2913	.2530	.3144	.6461	.5677	.7182	.1143	.1049	.1205
$S_g(\pi/2)$	.1012	.1007	.0994	.2126	.1487	.1763	.5304	.4176	.4702	.0097	.0081	.0081

D: $10 < x_1 + x_2 < \infty, \quad 10 < x_1 - x_2 < \infty$

	$g_1(x) = x_1$			$g_2(x) = x_2$			$g_3(x) = x_1 + x_2$			$g_4(x) = x_1 - x_2$		
p	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000
Mean	10.020	10.019	10.020	.0003	.0002	.0001	10.020	10.020	10.020	10.020	10.019	10.020
NSE	.0007	.0003	.0001	.0006	.0003	.0001	.0010	.0004	.0002	.0010	.0005	.0002
RNE	.839	.802	.910	1.089	1.091	1.053	.888	1.039	.941	1.008	.815	1.015
CD	2.246	.781	-.255	.184	-.045	-.001	1.656	.550	-.191	1.855	.577	-.181
$S_g(0)$	.0002	.0002	.0002	.0002	.0002	.0002	.0004	.0004	.0004	.0004	.0004	.0004
$S_g(\pi/2)$	.0002	.0002	.0002	.0002	.0002	.0002	.0004	.0004	.0004	.0004	.0004	.0003

E: $-.5 < x_1 + x_2 < .5, \quad -.5 < x_1 - x_2 < .5$

	$g_1(x) = x_1$			$g_2(x) = x_2$			$g_3(x) = x_1 + x_2$			$g_4(x) = x_1 - x_2$		
p	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000
Mean	-.0041	.0001	-.0011	.0091	.0072	.0032	.0050	.0073	.0020	-.0132	-.0072	-.0043
NSE	.0073	.0035	.0018	.0114	.0052	.0026	.0142	.0066	.0032	.0127	.0059	.0031
RNE	1.392	1.338	1.013	.956	.843	.683	.132	.901	.762	1.221	1.117	1.363
CD	-.971	-.754	.249	.745	.898	-1.526	.132	.276	-1.172	-1.264	-1.229	1.363
$S_g(0)$	.0209	.0240	.0310	.0509	.0538	.0676	.0804	.0875	.1025	.0631	.0680	.0947
$S_g(\pi/2)$	.0351	.0319	.0291	.0511	.0381	.0477	.0893	.0730	.0809	.0831	.0672	.0725

Table 3
Properties of the Gibbs Sampler, Truncated 15-Variate Normal Distribution

$$\mu = 0,\ \Sigma = R \otimes I_3,\ r_{ij} = \rho^{|i-j|}$$

$$x_1 \geq x_2,\ x_1 \geq x_3;\ x_5 \geq x_4,\ x_5 \geq x_6;\ x_7 \geq x_8,\ x_7 \geq x_9;\ x_{12} \geq x_{10},\ x_{12} \geq x_{11};\ x_{14} \geq x_{13},\ x_{14} \geq x_{15}$$

A: ρ = .00

	$g_1(x) = \sum_{i=1}^{5} x_{3i-2}$			$g_2(x) = \sum_{i=1}^{5} x_{3i-1}$			$g_3(x) = \sum_{i=1}^{5} x_{3i}$		
p	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000
Mean	.4336	.3444	.4263	.2821	.4637	.4011	-.8200	-.8176	-.8478
NSE	.0786	.0397	.0181	.0984	.0383	.0176	.0917	.0379	.0186
RNE	1.343	1.033	.954	.845	1.072	1.015	.986	1.169	.964
CD	.750	-.500	-.081	-.135	-1.086	-.310	1.786	.074	-.622
Sg(0)	2.4354	3.1338	3.2663	3.8149	2.9267	3.0835	3.3118	2.8582	3.4432
Sg(π/2)	3.6108	2.9291	3.2524	3.2953	3.0835	3.2071	3.2886	3.9327	3.2096

B: ρ = .50

	$g_1(x) = \sum_{i=1}^{5} x_{3i-2}$			$g_2(x) = \sum_{i=1}^{5} x_{3i-1}$			$g_3(x) = \sum_{i=1}^{5} x_{3i}$		
p	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000
Mean	.4790	.5861	.5312	.4222	.7204	.6776	-1.6333	-1.0562	-1.1207
NSE	.1711	.0847	.0386	.1777	.0806	.0382	.1833	.0827	.0415
RNE	.382	.354	.341	.397	.373	.340	.432	.402	.307
CD	-2.871	.541	1.403	-4.179	1.769	1.359	-2.694	.126	.842
$S_g(0)$	11.536	14.273	14.861	12.441	12.922	14.554	13.244	13.582	17.165
$S_g(\pi/2)$	3.235	3.079	3.176	3.446	3.469	3.132	3.990	3.748	3.554

C: ρ = .80

	$g_1(x) = \sum_{i=1}^{5} x_{3i-2}$			$g_2(x) = \sum_{i=1}^{5} x_{3i-1}$			$g_3(x) = \sum_{i=1}^{5} x_{3i}$		
p	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000
Mean	.7862	.2454	.4784	1.0146	.4719	.7426	-.6402	-1.1973	-.9049
NSE	.2363	.1661	.0830	.2376	.1694	.8232	.2248	.1778	.0858
RNE	.228	.126	.095	.231	.123	.094	.245	.120	.094
CD	-1.709	-2.347	.564	-1.613	-3.090	.101	-2.316	-4.281	.638
$S_g(0)$	22.007	54.795	68.716	22.238	57.024	67.556	19.913	62.802	73.314
$S_g(\pi/2)$	1.652	1.495	1.425	1.448	1.409	1.351	1.781	1.693	1.699

D: ρ = .95

	$g_1(x) = \sum_{i=1}^{5} x_{3i-2}$			$g_2(x) = \sum_{i=1}^{5} x_{3i-1}$			$g_3(x) = \sum_{i=1}^{5} x_{3i}$		
p	400	2,000	10,000	400	2,000	10,000	400	2,000	10,000
Mean	-1.0142	1.0650	-.1249	-.8501	1.2505	.0587	-1.7441	.2082	-.9101
NSE	.2928	.2660	.1298	.2884	.2665	.1295	.2913	.2661	.1311
RNE	.176	.081	.042	.175	.081	.041	.177	.082	.042
CD	4.001	2.166	2.280	3.957	1.988	2.229	3.231	1.603	2.036
$S_g(0)$	33.78	140.06	167.97	32.78	141.11	167.26	33.44	140.71	171.34
$S_g(\pi/2)$	.42	.42	.38	.33	.35	.34	.43	.41	.42

Using Projection Pursuit in Multispectral Image Analysis

G P Nason and Robin Sibson,
School of Mathematical Sciences,
University of Bath, Bath, UK.

ABSTRACT

Principal components analysis is already used in multispectral image analysis to reduce the number of spectral dimensions. We propose to use projection pursuit to find interesting combinations of spectral variates that produce images that enhance contrast differences between differing land-use types. We develop a 3-dimensional moment index based on Jones and Sibson's index for projection into 2-dimensions.

1 Introduction.

Remote sensing is an indispensable tool in many scientific disciplines. It is one of the major tools in monitoring our own environment in a cost-effective way. We will be investigating methods of treating multispectral images, which reduce the number of spectral dimensions, without losing significant information. This information extraction process has been performed in many ways in the past. We develop the necessary techniques to perform projection pursuit, which is to be used in a similar rôle to principal components analysis.

2 The Practical Problem.

The NERC Computer Services kindly supplied us with much *thematic mapper* data. These data sets consist of images collected by a Daedalus thematic mapper, flown in an aeroplane above the area to be remote sensed. The mapper passively senses 12 different spectral channels.

A monoimage of the area is recorded at each spectral frequency. The image that we decided to use was one of the Chew Valley Lake, Somerset, UK. We decided to use this image since it has a good mix of land and water features. Each monoimage consists of 1254x715 pixels, which take discrete values in the range of 0 to 255. We generally operate upon sections of the whole image.

Table 1 details the frequencies that the scanner detects.

Channel	Frequency (μ m)
1	0.42 - 0.45
2	0.45 - 0.52
3	0.52 - 0.60
4	0.605 - 0.625
5	0.63 - 0.69
6	0.695 - 0.75
7	0.76 - 0.90
8	0.91 - 1.05
9	1.55 - 1.75
10	2.08 - 2.35
11	8.50 - 13.00
12	8.50 - 13.00

Table 1: Spectral channels sensed by NERC Daedalus thematic mapper.

2.1 Viewing the image.

One thing we would want to do with this image is look at it. We could view 12 separate monoimages, but it is useful to somehow combine the images to form a colour image. Colour is effective for highlighting differences in land use and type, and directs the eye to various features.

We would generally view the image on a CRT monitor, and would maybe later obtain a hardcopy. Most colour monitors use the red-green-blue (RGB) system of specifying colours (to span the 3D colour space that humans perceive[1]), although this is not the only system that we could use. One way to obtain a quick view of the image is to choose three mapper bands and assign them to one of the RGB colours.

The difficult question is: what mapper frequencies do we use, and which colours do we assign them to? Note also, that there are $P_3^K = \frac{K!}{(K-3)!}$ ways of choosing such assignments (e.g. $P_3^{12} = 1320$). To view all of them, and select good images, is at best non-objective,and at worst, horrendously time-consuming.

2.2 Multivariate methods.

We wish to move onto more incisive techniques of variable reduction. For these techniques, we wish to consider the image as a multivariate data set. To do this we regard spectral channels as variates, and pixels as cases. We will let K represent the number of variates, and N the number of cases (e.g. $K = 12, N = 896610$).

2.3 Why dimension reduction?

To end this section we mention two other reasons why dimension reduction is a useful processing step.

It is very common to run an automatic classifier over an image. Due to the *curse of dimensionality* (see [4]) these algorithms can become confused, and work much better in lower dimensions.

Secondly, the amount of remotely sensed data collected is increasing at an alarming rate, and so knowing what to keep is important.

2.4 Data quality.

From monoimages we have found spectral channels 1 and 7 to be very noisy. Also, channel 12 records at the same frequency as channel 11, except at a different gain level. For these reasons we have discarded channels 1,7 and 12 from the analysis giving an effective set of nine variates.

3 Analysis by Principal Components Analysis.

Principal components analysis is an established multivariate technique already used for dimension reduction in image analysis (where it is also known as *decorrelation*. Full and detailed treatments of principal components analysis can be found in most applied multivariate texts (e.g. [6]).

We compute principal components from the correlation matrix of the image. The correlation matrix usually tells us that channels of similar frequency are highly correlated.

Since humans perceive a 3D colour space, we will usually choose the 3 principal components associated with the 3 largest eigenvalues.

3.1 Results of principal components analysis.

In Table 2 we display a typical set of eigenvalues. From this one can see that the first 3 principal components account for over 90% of the variation inherent in the data (so maybe 3 dimensions are adequate). The first principal component in our example is typically not very far from

$$-(K^{-\frac{1}{2}}, K^{-\frac{1}{2}}, \ldots, K^{-\frac{1}{2}})^T$$

Number	Eigenvalue	% Variance Expl.
1	6.88	76
2	1.50	17
3	0.387	4.3
4	0.130	1.4
5	0.0569	0.63
6	0.0323	0.36
7	0.0138	0.15
8	0.00612	0.068
9	0.00149	0.017

Table 2: Eigenvalues from typical principal components analysis.

In layman's terms, the first principal component appears to be a roughly equal combination of all the original spectral variates. This component has a intuitive interpretation as a brightness variate and so we assign it to the B of the hue-saturation-brightness (HSB) colour model.

The remaining principal components are usually contrasts of certain channels. On an rendered image this has the effect of providing contrast enhancements.

4 Analysis by Projection Pursuit.

4.1 What is projection pursuit?

Exploratory projection pursuit can be used for the same purposes as principal components analysis. We wish to use the cluster-detecting ability of projection pursuit, just as we would with ordinary multivariate data.

We do not wish to describe exploratory projection pursuit in great detail. Interested readers should consult [5] or [2] for more information.

4.2 Projection pursuit into 3 dimensions.

Many projection indices have been proposed in the literature[5] [2][3] None have yet been explicitly developed for projection into 3 dimensions, although for some it is a relatively trivial matter to do so. We also prefer an index that is rotationally invariant with respect to the chosen basis in the projection space.

However, the overriding consideration for us is computational efficiency. All indices in the literature (that we know of) have a computational effort of order N or larger. One index that almost overcomes this barrier is the *moment* index described in [5]. Once a set of summary statistics is computed for a data set the subsequent computation of an optimal projection solution does not depend on N. Since a common method of searching for optimal projections depends on many random starting

positions, this independence of N is very useful, since for these image problems N is usually very large.

4.3 Review of the moment index.

The moment index[5] is derived from the order-1 negative Shannon entropy index. We have extended the moment index to a 3D space and obtain

$$\begin{aligned} & \left[k_{300}^2 + 3k_{210}^2 + 3k_{201}^2 + 3k_{120}^2 + 6k_{111}^2 \right. \\ & \left. + 3k_{102}^2 + k_{030}^2 + 3k_{021}^2 + 3k_{012}^2 + k_{003}^2\right] \\ + & \frac{1}{4}\left[k_{400}^2 + 4k_{310}^2 + 4k_{301}^2 + 6k_{220}^2 + 12k_{211}^2\right. \\ + & 6k_{202}^2 + 4k_{130}^2 + 12k_{121}^2 + 12k_{112}^2 + 4k_{103}^2 \\ + & \left. k_{040}^2 + 4k_{031}^2 + 6k_{022}^2 + 4k_{013}^2 + k_{004}^2\right]. \end{aligned}$$

as our 3D index, where $k_{...}$ are trivariate k-statistics.

This projection index is rotationally invariant with respect to choice of basis in the projection space, and the derivatives with respect to the projection space can be calculated.

4.4 Sphered images.

Sphering[1] [7] is a transformation that transforms the original data set into one that has zero mean and identity variance.

It is very interesting to observe the results of the sphering process applied to the image data. What almost seems like a ghost picture of the "original" results. Certain things remain, for example, edges of fields, certain buildings, indicative of jump changes in intensity which will not be accounted for by linear correlation.

4.5 Results of projection pursuit.

Once we have a 3 dimensional projection solution we still have to decide how we are to apply the solution to the RGB guns of a CRT. Usually the projection solution is transformed back to the unsphered space of variates, and then principal components is applied to the data in this space.

Unlike principal components analysis, projection pursuit finds no brightness component, this is probably due to the action of sphering. Projection pursuit finds linear combinations that it finds interesting.

The moment index has been criticised in the past for rewarding projections which contain outliers. We use this and the image's spatial structure to our advantage to find prominent outlier features, having unique reflectance properties.

[1] also known as the Mahalanobis transformation[6]

Otherwise, projection pursuit finds interesting contrasts of the original variates, which are usually different from those found using principal components. Sometimes, one finds that ground structure is highlighted more effectively with a projection pursuit contrast than a principal components one.

5 Conclusions.

We take the view that projection pursuit should act in a complementary rôle to principal components analysis. It has the potential to find interesting clusters and act as a valuable dimension-reducer.

After practical experience with colour images and their manipulation, we realise how dangerous it is to compare the performance of various methods when the output is a colour image. Sometimes changing the colour assignments in an image can be more revealing than changing a linear combination of channels.

However, for automatic classifiers and storage we must be able to reduce dimension effectively, without losing too much, and projection pursuit will be useful here.

We must investigate the use of other colour models. We have used RGB and HSB models here, there may be others which might fit in more naturally.

We could also try other projection indices, or search for projection spaces one-dimension at a time.

6 Acknowledgements.

This work was performed with the support of a grant from the Science and Engineering Research Council, and G P Nason is supported by a SERC Research Studentship. We thank NERC Computer Services for supplying the thematic mapper data.

References

[1] Richard P. Feynman. *The Feynman lectures on physics*, volume 1. Addison, Reading, Mass., 1963.

[2] Jerome H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397), March 1987. Theory and Methods.

[3] Peter Hall. On polynomial-based projection indices for exploratory projection pursuit. *The Annals of Statistics*, 17(2):589–605, 1989.

[4] P. J. Huber. Projection pursuit (with discussion). *The Annals of Statistics*, 13:435–525, 1985.

[5] M.C. Jones and R. Sibson. What is projection pursuit? (with discussion). *Journal of the Royal Statistical Society: Series A*, 150:1–36, 1987.

[6] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Probability and mathematical statistics. Academic Press, London, 1979.

[7] P.A. Tukey and J.W. Tukey. Preparation; prechosen sequences of views. In V. Barnett, editor, *Interpreting Multivariate Data.*, pages 189–213. Wiley, Chichester, 1981.

Reconstruction of Binary Images

Charles Kooperberg*
Department of Statistics
University of California
Berkeley, CA 94720

1 Introduction

We consider the following problem: a black and white image is observed in digitized form. Unfortunately the 'real' image is not observed: at some stage the image has been distorted with noise. Our objective is to remove as much of the noise as possible, to get approximately the original image back.

In a more mathematical setting, let x be an m by n array, with entries 0 and 1; x is considered to be a realization of a random variable X. We do not observe the image x. Instead we observe y, a noisy version of x that is a realization of the random variable Y, where the distribution of Y depends on x. We want to estimate x on the basis of y.

The problem that we are discussing is a special case of the more general image reconstruction problem, y is a set of records generated by degradation of the true image x. The noisy image y and the original image x may or may not be closely related. Two of the most influential papers discussing these problems are Besag (1986) and Geman and Geman (1984). Since then a large body of literature about image reconstruction has developed. See Besag (1989) and Geman (1991) for a review of this area.

A common assumption is to put a Markov Random Field as a prior on the images. Using a Markov Random Field as prior on the images leads to a global optimization problems to reconstruct the original image. There are several algorithms to deal with this optimization problem, for example, Gibbs sampling, simulated annealing and ICM.

We will not assume a Markov Random Field in this paper. Instead it is assumed that the probabilities of observing a certain pattern in the image are the same everywhere in the image. We will study the independent Bernoulli noise case. For the algorithm which we will discuss the decision about the (i, j) pixel in the original image, is made based upon those pixels in the observed image that are within a window around (i, j). Except for the noise level ϵ all statistics required to make a decision about this pixel, can be gathered, in an empirical Bayes fashion, from the image.

This article is a summary of chapter 1 of Kooperberg (1991).

2 A Bayes Window Estimator for Binary Images

Some definitions and notation:
Let S be a finite subset of $\mathbf{Z}^2$ and let $\mathbf{B}^S$ be the collection of functions on the elements of S that are 0-1 valued.

Let A_{nm} be the set $\{(i, j) : 1 \leq i \leq m, 1 \leq j \leq n\}$. If $S = A_{nm}$ we can think of $\mathbf{B}^S$ as the collection of $n \times m$ arrays with entries 0 and 1. We will write $\mathbf{B}^{nm}$ instead of $\mathbf{B}^{A_{nm}}$. If there is no confusion we will omit the n and m and we will write $\mathbf{B}$ instead of $\mathbf{B}^{nm}$. An element $x \in \mathbf{B}$ can be written as $x = (x_{ij},\ 1 \leq i \leq m,\ 1 \leq j \leq n)$.

Let $(0, 0) \in S$. If (i, j) and S are such that $1 \leq (i+k) \leq m$ and $1 \leq (j + l) \leq n$ for all $(k, l) \in S$, we can define a *window-operator* W_{ij}. Informally, $W_{ij}x$, $x \in \mathbf{B}$ is that part of x that falls within S, when S is positioned such that the origin of S is positioned at (i, j).

Define a *window-operator* $W_{ij} : \mathbf{B}^{nm} \to \mathbf{B}^S$ as:

$$(W_{ij}x)_{kl} = x_{i+k,j+l},\ \ (k, l) \in S.$$

We also need to define the *center-less window-operator* $E_{ij} : \mathbf{B}^{nm} \to \mathbf{B}^{S\setminus\{(0,0)\}}$ as:

$$(E_{ij}x)_{kl} = x_{i+k,j+l},\ \ (k, l) \in S,\ (k, l) \neq (0, 0).$$

Thus a window-operator cuts a piece of shape S from $x \in \mathbf{B}^{nm}$, centered at (i, j); a center-less window-operator cuts out the same piece, except for the center pixel (i, j).

Now let $x \in \mathbf{B}$ be the image that we want to reconstruct. x is a realization of the random variable X. Instead of x we observe a realization of Y, y. The Y_{ij}'s are

*Research supported in part by NASA NCA2-488 and NSF DMS-84-51753. I like to thank my Ph.D. adviser, David Donoho, for many helpful discussions and suggestions.

conditionally independent given that $X = x$:

$$\begin{aligned} P(Y_{ij} = x_{ij} \mid X_{ij} = x_{ij}) &= 1 - \varepsilon \\ P(Y_{ij} = 1 - x_{ij} \mid X_{ij} = x_{ij}) &= \varepsilon, \end{aligned}$$

with $0 < \varepsilon < 0.5$.

Let $x \in \mathbf{B}$, let $T : \mathbf{B} \to \mathbf{B}$ be an estimator of x based upon y. Define

$$\begin{aligned} R(X, T) &= E\left(\sum_{i,j} \mid X_{ij} - T(Y)_{ij} \mid /(nm)\right) \\ &= \sum_{ij} P\left(X_{ij} \neq T(Y)_{ij}\right)/(nm) \end{aligned}$$

to be the *expected misclassification error* of T as estimator of X.

We call T_S a *window-estimator* if $T_S : \mathbf{B} \to \mathbf{B}$, and if $T_S(Y)_{ij}$ is independent of $\{Y_{kl}, (k-i, l-j) \notin S\}$ given $W_{ij}Y$.

Fix a window $S \subset \mathbf{Z}^2$. The following theorem holds:

Theorem: (i) *The window-estimator T_S, S fixed, which minimizes the expected misclassification error $R(X, T_S)$ has the form:*

$$T_S(y)_{ij} = \begin{cases} y_{ij} & \text{if } P\left(Y_{ij} = y_{ij} \mid E_{ij}Y = E_{ij}y\right) \\ & \quad \geq 2\varepsilon(1-\varepsilon), \\ 1 - y_{ij} & \text{otherwise;} \end{cases} \qquad (1)$$

(ii) *The expected misclassification error achieved by this estimator is:*

$$R(X, T_S) = \frac{1}{2} - \frac{1}{2(1-2\varepsilon)} E\left|1 - \frac{2\varepsilon(1-\varepsilon)}{U}\right|,$$

where $U = P\left(Y_{ij} = 1 - y_{ij} \mid E_{ij}Y = E_{ij}y\right)$.

See Kooperberg (1991) for the proof.

What does this mean? It says that one gets the Bayes window estimator by the following procedure:

a) Cover the pixel (i, j) that you want to reconstruct.

b) Compute the probability that this pixel in the *observed* image is white (or black). $(P(Y_{ij} = 1 \mid E_{ij}Y = E_{ij}y)$.

c) If this probability makes you pretty sure (either $P(Y_{ij} = 1 \mid E_{ij}Y = E_{ij}y) \geq 1 - 2\varepsilon(1-\varepsilon)$ or $P(Y_{ij} = 0 \mid E_{ij}Y = E_{ij}y) \geq 1 - 2\varepsilon(1-\varepsilon)$), then this is the Bayes estimate of the pixel in the *original* image x_{ij}.

d) If there is still doubt, remove the cover, and the color that you observed (y_{ij}) will be the estimate for the original color (x_{ij}).

3 An Empirical Bayes Window Estimator for Binary Images

To use the window estimator (1) information about the distribution of $W_{ij}Y$ (or $W_{ij}X$; to compute the probabilities in (b) above) and ε is needed. Although information about the distribution of $W_{ij}Y$ will be seldom availabie, we will assume that ε is (approximately) known.

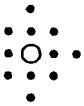
To get information about the distribution of $W_{ij}Y$ we can now take an empirical Bayes approach, and use the data to estimate this distribution. If we assume that the distribution of $W_{ij}Y$ (and thus the distribution of $W_{ij}X$) does not depend on i and j (homogeneity) counting for how many pixels (k, l) $W_{ij}y = W_{kl}y$ gives the empirical estimator

$$\begin{aligned} &\widehat{P}(Y_{ij} = y_{ij} \mid E_{ij}Y = E_{ij}y) \\ &\quad = \frac{\sum_{kl} I\left(W_{ij}y = W_{kl}y\right)}{\sum_{kl} I\left(E_{ij}y = E_{kl}y\right)}, \end{aligned} \qquad (2)$$

where $I(\cdot)$ is the usual indicator function.

There is a problem with this estimator though. Clearly we would like to have a large window to incorporate as much information as possible in the decision. However a large window might lead to very small counts in (2). Even for a relatively small 5×5 window we would be counting the empirical distribution on $2^{24} = 16,777,216$ points. Even to get an average of just 1 observation in each cell we need a picture of 4096 by 4096 points!

One possible modification is to use a large window whenever this is possible, but to use a smaller window if the estimator in (2) would be based on very small counts. For example, we could first use a window of size

13:  , and make a 60% confidence interval for $\widehat{P}(Y_{ij} = y_{ij} \mid E_{ij}Y = E_{ij}y)$ based upon $\sum_{kl} I(W_{ij}y = W_{kl}y)$ and $\sum_{kl} I(E_{ij}y = E_{kl}y)$. If this interval does not cover $2\varepsilon(1-\varepsilon)$ we make a decision, while if it does cover $2\varepsilon(1-\varepsilon)$ we make the decision based upon a window of size 9:  .

Another modification is to assume left-right, top-bottom and/or diagonal symmetry of the prior distribution of $W_{ij}Y$. Each symmetry reduces the number of different patterns by a factor of 2.

These two modifications are used in our examples. Other possible modifications that we do not use include: (i) assume black-white symmetry of the prior distribution on $W_{ij}Y$; or (ii) a procedure in which we do not only count those patterns that are exactly the same, but

also those that are almost the same, i.e. differ only in one or two points. Those that are different would than, conceivably, make a smaller contribution than those that are exactly the same. The reason that we do not use the later idea is that we do not know of an algorithm to implement this rule that would use less than $O((nm)^2)$ time, while without this idea, we can implement the algorithm in $O(nm \log(nm))$ time.

4 Examples

We used the empirical Bayes window reconstruction rule, as described in the previous section on a number of initial examples. Upon examination of the results, we concluded that the estimator was working reasonably well, but that it left too many small spots and was a bit to rough to please our eye. This is actually not surprising: our estimator did not make any assumptions about smoothness, while in practice images do tend to be (somewhat) smooth. We decided to carry out some post-processing to further smooth the picture. We settled on the following operation: change all black(white) pixels, that together with at most 12 other black(white) pixels, are not connected to any other black(white) pixels and are completely surrounded by white(black) pixels.

We applied the algorithm to several other examples. Among them the same examples as were used in Greig, Porteous and Seheult (1989). We added 25% Bernoulli noise to their figure 1. Our reconstruction had 4.8% incorrect estimated pixels after post-processing (9.8% before post-processing). The methods discussed in Greig et.al. (1989) (annealing, ICM and exact MAP) had between 5.2% and 5.4% incorrect estimated errors.

For the figure that was first used as Figure 4 in Besag (1986). This was an $88 times 100$ hand-constructed scene, designed specifically to contain some awkward features. We applied our algorithm with 30% additive Bernoulli noise. There were 6.5% incorrect classified pixels after post-processing. For the other methods Greig et.al. (1989) obtained between 5.4% and 7.0% incorrect classified pixels using several different other reconstruction methods.

On the next page we show two larger examples. Typically, for images with the amount of detail as these figures have, 10% incorrect pixels in Y, the noisy image, are reduced by our reconstruction method to about 1% incorrect pixels in $\widehat{X}$, the reconstructed image. 20% errors in Y is reduced to about 2-3% in $\widehat{X}$; 30% errors in Y is reduced to about 4-8% in $\widehat{X}$; and 40% errors in Y is reduced to about 15-30% in $\widehat{X}$.

5 Discussion

We have introduced a reconstruction rule for binary images. The rule only uses the information within a finite window centered on the point to be reconstructed. The rule is, among all the rules based on that window, the one that minimizes the expected number of incorrectly reconstructed pixels. Surprisingly, the rule can be expressed in the statistics of the observed image only. Therefore, to use the rule, we do not need to know the prior distribution of the images.

If we assume stationarity we can apply the rule in an empirical Bayes fashion. All the necessary parameters can be estimated from the observed image. No training images are required.

Our examples suggest that the method works well for binary images with a small amount of noise if some post-processing is applied. In these cases it removes almost all the noise. In binary images with higher noise levels or more details the method still works quite well. The results are comparable to those achieved by some other methods in the literature. We should point out though that, although some generalizations are possible, our method is not yet applicable to such a wide range of different problems as several of the other methods are applicable to. Further work is needed to explore the possibilities to extend the window based method to other problems.

References

Besag, J. (1986). On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society B* **48** 259-302.

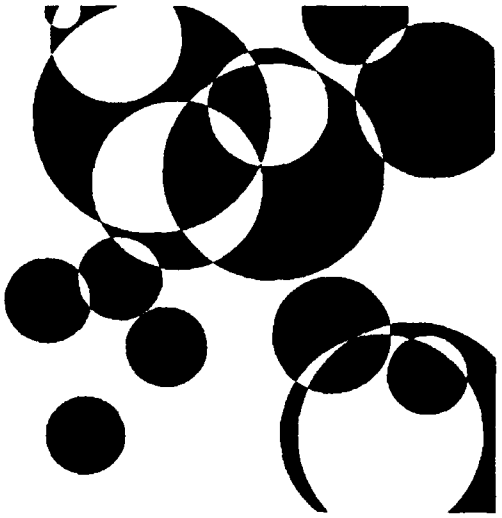
Besag, J. (1989). Towards Bayesian Image Analysis. *Journal of Applied Statistics* **16** 395-407.

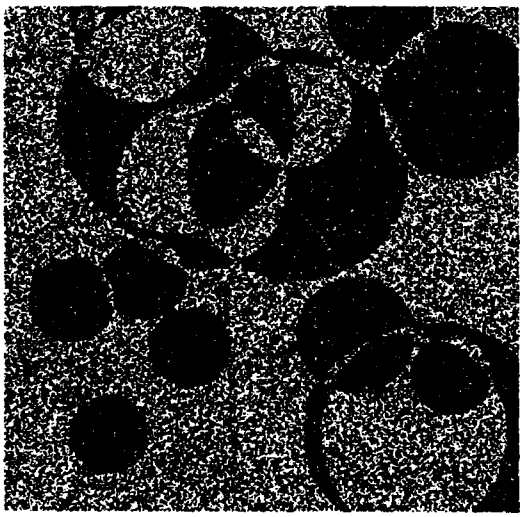
Geman, D. (1991). Random Fields and Inverse Problems in Imaging. *Lecture Notes in Mathematics* Springer, Berlin.

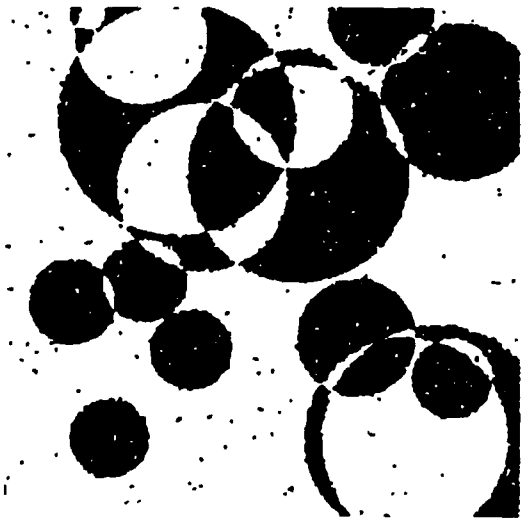
Geman, D. and Geman, S. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6** 721-741.

Greig, D.M., Porteous, B.T. and Seheult, A.H. (1989). Exact Maximum a Posteriori Estimation for Binary Images. *Journal of the Royal Statistical Society B* **51** 271-279.

Kooperberg, C. (1991). *Smoothing Images. Curves and Densities* Ph.D. thesis, University of California at Berkeley.







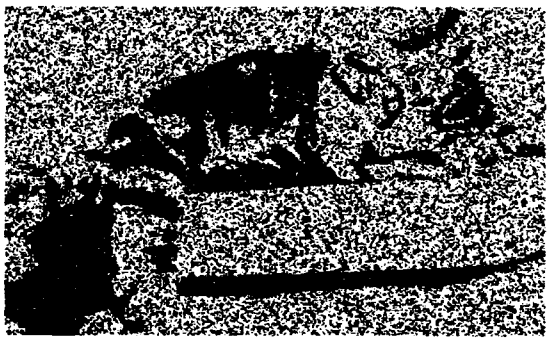


Figure 1 (left): *circles*, top to bottom:
- original (600 × 600 pixels);
- with 30% errors;
- reconstruction, with 2.5% errors.

Figure 2 (right): *kitty*, top to bottom:
- original (630 × 390 pixels);
- with 20% errors;
- reconstruction, with 2% errors.

ESTIMATING BAND-TO-BAND MISREGISTRATIONS IN MULTIVARIATE IMAGERY

by

Mark Berman (1), Andrew A Green (2), Leanne Bischof (1), Steven J Davies (1) and Maurice D Craig (2)

(1) CSIRO Division of Mathematics & Statistics, PO Box 218, Lindfield 2070, NSW, Australia

(2) CSIRO Division of Exploration Geoscience, PO Box 136, North Ryde 2113, NSW, Australia

Abstract

In this paper, we use simulations and some theory to show that many of the standard techniques used to estimate band-to-band misregistrations in multivariate imagery are biased. These biases, although typically small, become important because they are much larger than the standard errors of the estimators obtained for most image pairs, which often consist of sample sizes of the order of 10^5 or 10^6. We develop an empirical method for effectively correcting the biases in some of the methods. Typically, this enables us to estimate misregistrations to within about 1/100th of a pixel.

1 Introduction

In recent years, quite a few papers have been published in the remote sensing and image processing literatures on methods for estimating band-to-band misregistrations in multivariate imagery. Accurate band-to-band registration is important for the multivariate analysis of such imagery, because a basic assumption is that all components of a vector of spectral values refer to the same ground location. This is reflected in specifications for various airborne and satellite multispectral scanners, which typically require that the bands be registered to within 0.1 or 0.2 pixels.

Most of the methods used to estimate band-to-band misregistrations are either cross-covariance-based or Fourier-based. In this paper, we show that, when these methods are applied to remotely sensed imagery, both usually give biased estimates of the misregistrations, the former because of inadequate interpolation procedures and the latter because they do not account for the presence of aliasing in the data. We describe a Fourier-based method which accounts for aliasing and which, for a variety of 512 x 512 image pairs, gives misregistration estimates with standard errors in both horizontal and vertical directions of less than 1/100th pixel! Because of space limitations, only an outline of the work is given here. More extensive descriptions can be found in Berman et al (1990, 1992).

Much of the theory rests on essentially one-

dimensional ideas. Consequently, we deal first, in Section 2, with one dimensional images, that is time series data. This is extended in Section 3 to two dimensional images.

2 One-Dimensional Images or Time Series

Suppose we observe two time series $\{Y_j(t)\}_{t=1}^N$, $j = 1,2$ satisfying the relationships

$$Y_j(t) = \alpha_j + \beta_j S(t + L_j) + \epsilon_j(t), \quad (2.1)$$

t=1,...,N,j=1,2, where $S(t)$ (the "signal") and the $\epsilon_j(t)$ (the "noise") are assumed to be weakly stationary processes that are mutually uncorrelated with $E(\epsilon_j(t)) = 0$. The parameter of interest is $D = L_2 - L_1$. Because pixel values obtained from sensors such as cameras are usually integrals of brightness values over a region corresponding approximately to the pixel, we can further assume that approximately

$$S(t) = \int_{t-1}^{t} X(u)du, \quad (2.2)$$

where $X(u)$ is itself a continuous weakly stationary process.

A naive estimator of D, used widely in remote sensing, is obtained by finding the maximum of the cross-covariance function of the two time series. If $\gamma_s(t) = cov(S(u), S(u + t))$, we see from (2.1) that $cov(Y_1(u), Y_2(u + t)) = \beta_1\beta_2\gamma_s(t + D)$, which (assuming a unique maximum) is maximised when t = -D. However, because the data are not continuously observed, we can estimate $\gamma_s(t)$ directly only for integer t. Hence, if D is non-integer, we need to interpolate our estimates of $\gamma_s(t)$ in the vicinity of its maximum to estimate it. The appropriate interpolator is highly data-dependent. Using simulations, we have found that this often leads to estimates with a bias of about 0.1 of a pixel; see Berman et al(1992, Section 3).

More sophisticated estimation procedures can be based on the Fourier transform. Let

$$F_j(\omega_u) = (2\pi N)^{-\frac{1}{2}} \Sigma_{t=1}^{N} Y_j(t) e^{it\omega_u}, \quad (2.3)$$

$(\omega_u = 2\pi u/N, u = 1, \ldots, [N/2])$ denote the discrete Fourier transform of series j, and let $\hat{\theta}(\omega_u)$ denote the phase difference between the two series at frequency ω_u (note that $\hat{\theta}(-\omega_u) = -\hat{\theta}(\omega_u)$). If either (a) $D = K/2$, where K is an integer, or (b) there is no aliasing of the data, then it can be shown that, in large samples $\hat{\theta}(\omega_u) \sim D + 2\pi m(\omega_u)$, where $m(\omega_u)$ is that integer ensuring that $\hat{\theta}(\omega_u) \in (-\pi, \pi]$. Note that, if $|D| < 1$, which usually is the case with remotely sensed data, $m(\omega_u) = 0$. Hamon and Hannan (1974, Section 2) assert that (provided that there is no aliasing) the asymptotically optimal estimator of D maximizes

$$\Sigma_{0<u<N/2} W(\omega_u) \cos(\hat{\theta}(\omega_u) - D\omega_u), \quad (2.4)$$

where

$$W(\omega_u) = \sigma^2(\omega_u)/(1 - \sigma^2(\omega_u)) \quad (2.5)$$

and $\sigma^2(\omega_u)$ is the coherence between the two series. Since the coherence is usually unknown, it needs to be estimated from the data; see Hamon and Hannan (1974) for details. Hannan and Thomson (1988) consider the behaviour of (2.4) and other asymptotically equivalent estimators under low signal-to-noise scenarios. It is also worth noting (as

Hannan (1975) and Chan et al (1978) have) that, when $|D| < 1$ and the noise is small, maximising (2.4) is approximately equivalent to minimising

$$\Sigma_{0<u<N/2} W(\omega_u)\{\hat{\theta}(\omega_u) - D\omega_u\}^2. \quad (2.6)$$

Of course, the advantage of (2.6) is that it has an explicit solution.

Unfortunately, the presence of edges in images (e.g. rivers, roads, fractures, cell or property boundaries) means that frequencies higher than half the sampling rate are often present, in which case the data are aliased. This manifests itself in biases in $\hat{\theta}(\omega_u)$ for various ω_u. If there is no aliasing and $|D| < 1$, then in large samples, $\hat{\Delta}(\omega_u) \equiv \hat{\theta}(\omega_u)/\omega_u$ (the phase delay) $\sim D$. As an experiment, we generated 512 pairs of time series from real data satisfying (2.1) and a discrete approximation to (2.2). For each pair, $N = 101$ and $D = 0.2$. Further details can be found in Berman et al (1992, Section 3). Fig. 1 shows the means (plus and minus one standard deviation) of the phase delays for the 512 data sets at the [N/2] = 50 positive frequencies given after (2.3). Biases are clearly present at high frequencies and at very low frequencies. The latter are due to the fact that the data are not periodic at the boundaries, and can be largely corrected by "tapering" (Bloomfield, 1976, Section 5.2); see Berman et al (1992, Section 3) for details.

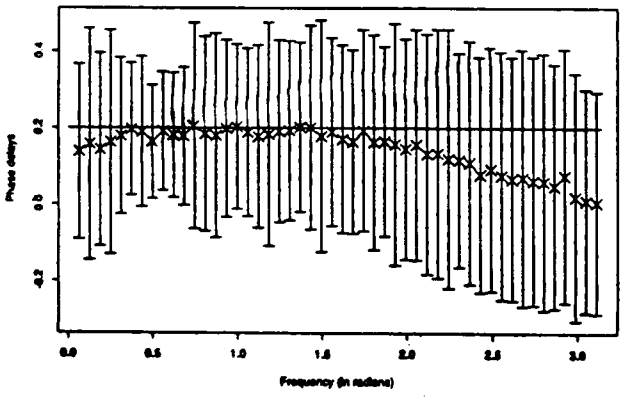


Fig.1

The high frequencies are due to aliasing in the data. It can be shown that, under mild regularity conditions, $\hat{\theta}(\omega)$ will converge, as $N \to \infty$, to $\theta(\omega)$, the phase of the function

$$E(\omega) \equiv (2\pi)^{-1}\Sigma_{l=-\infty}^{\infty}\gamma_s(l+D)e^{-il\omega} \quad (2.7)$$

$$= e^{iD\omega}\Sigma_{l=-\infty}^{\infty}e^{2\pi ilD}f_s(\omega+2\pi l) \quad (2.8)$$

where

$$f_s(\omega) = (2\pi)^{-1}\int_{-\infty}^{\infty}\gamma_s(t)e^{i\omega t} \quad (2.9)$$

denotes the spectrum of the signal. Depending on the nature of $\gamma_s(t)$ and $f_s(\omega)$, it will sometimes be convenient to use (2.7) to compute $\theta(\omega)$ and sometimes (2.8). The "unbiasedness" of the cases (a) D = K/2 and (b) no aliasing of the data (i.e. $f_s(\omega) = 0, |\omega| > \pi,$) follow easily from (2.8). We have computed (2.7) or (2.8) for a range of values of $D \in (-1,1)$ and for a variety of autocorrelation functions. Typically, they asymptote to D as $\omega \to 0$ and converge to 0 as $\omega \to \pi$. A theoretical explanation for this phenomenon is given in a Proposition in Berman et al (1992, Section 4). As an example, Fig. 2 shows the phase delay when $D = 0.2, cov(X(u), X(u+t)) = \rho^{|t|}$, and ρ = .99 (solid line), .5 (dots) and .1 (dashes).

Note how in Fig. 2 (and in Fig. 1 if we taper appropriately) the estimates of the phase delay are, for practical purposes, unbiased below a cutoff frequency (which will be application dependent). When $|D| < 1$, a practical estimate of it can be obtained by taking a weighted mean of the phase delay estimates at frequencies less than the cutoff frequency (assuming we can obtain a good estimate of it), where the weights are inversely proportional to the error variances of the corresponding phase delay estimates.

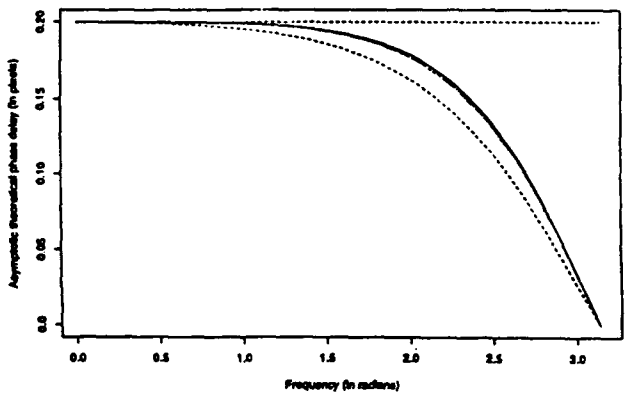


Fig.2

However, we have found that, when $|D| < 1$, a very good empirical approximation to many phase delay curves is given by the formula

$$\Delta(\omega) = D - A(e^{B\omega} - 1). \qquad (2.10)$$

Typically $B > 0$. We estimate the parameters by non-linear weighted least squares, where the weights are again inversely proportional to the error variances; see Berman et al (1992, Section 5) for further details. We can interpret this as an approximate means of finding the cutoff frequency. Further, since we are interested in estimating D, precise estimation of A and B is not important.

When $|D| \geq 1$, we have found it best to estimate the integer part of D first, using a method such as the minimisation of (2.4), then shift one time series the relevant number of integer units with respect to the other, remove the non-overlapping parts of the resulting data sets, and finally estimate the fractional part using (2.10) in conjunction with weighted least squares estimation.

3 Two-Dimensional Images

Much of the one-dimensional theory above is readily extendible to two dimensions, and hence applicable to the misregistration problem. Let D_x and D_y denote the shifts in the x and y directions respectively. Equations (2.1) and (2.2) extend in an obvious way. Again, with the aid of simulated data, we have sometimes been able to demonstrate biases in cross-covariance-based methods of about 0.1 of a pixel. The Fourier theory also extends easily. For $M \times N$ images, the two dimensional Fourier transform of image j is

$$F_j(\omega_u, \chi_v) \propto \Sigma_{s=1}^{M} \Sigma_{t=1}^{N} Y_j(s,t) e^{i(s\omega_u + t\chi_v)}, \qquad (3.1)$$

($\omega_u = 2\pi u/M, \chi_v = 2\pi v/N, u = -M/2, \ldots, M/2 - 1, v = -N/2, \ldots, N/2 - 1$). When aliasing is absent, the phase difference in large samples satisfies

$$\hat{\theta}(\omega_u, \chi_v) \sim \omega_u D_x + \chi_v D_y + 2\pi m(\omega_u, \chi_v), \quad (3.2)$$

where $m(\omega_u, \chi_v)$ is that integer chosen to ensure that $\hat{\theta}(\omega_u, \chi_v) \in (-\pi, \pi]$. If $|D_x| + |D_y| < 1, m(\omega_u, \chi_v) = 0$. For the time being, we shall assume this to be so. When aliasing is present, there are two options. In the first option, we can find a rectangular region around the origin for which (3.2) is a good approximation. This involves finding cutoff frequencies in both the x and y directions. Let

$$\hat{\Delta}_x(\omega_u, \chi_v) \equiv \{\hat{\theta}(\omega_u, -\chi_v) + \hat{\theta}(\omega_u, \chi_v)\}/2\omega_u, \quad (3.3)$$

$u = 1, \ldots, M/2 - 1, v = 1, \ldots, N/2 - 1$. For those frequencies for which (3.2) is a good approximation, it is easily seen that, in large samples, $\hat{\Delta}_x(\omega_u, \chi_v) \sim D_x$. A suitably weighted mean of the $\hat{\Delta}_x(\omega_u, \chi_v)$'s gives an appropriate estimator of D_x . An analogous procedure holds for estimation of D_y. Under mild assumptions, the two estimators are approximately uncorrelated. See Berman et al (1990, Section 3) for further details.

A second, more appealing option, which we now use routinely, is the following. First, we assume separability of the autocovariance function of the signal, i.e. $cov(S(s,t), S(s+u, t+v)) = \gamma_x(u)\gamma_y(v)$, where $S(s,t)$ denotes the signal at (s,t). It follows easily that the limiting phase difference, $\theta(\omega_u, \chi_v)$, will satisfy $\theta(\omega_u, \chi_v) = \theta_x(\omega_u) + \theta_y(\chi_v)$, and hence that, in large samples,

$$\hat{\Delta}_x(\omega_u, \chi_v) \sim \theta_x(\omega_u)/\omega_u. \quad (3.4)$$

Note that the right-hand side of (3.4) is independent of χ_v, and is also the phase delay of a ONE-DIMENSIONAL time series, which we have found is well modelled by (2.10). Our solution therefore is to compute

$$\hat{\Delta}_x(\omega_u) = \Sigma_v \tau_v^{-2} \hat{\Delta}_x(\omega_u, \chi_v)/\Sigma_v \tau_v^{-2}, \quad (3.5)$$

where $\tau_v^2 = Var\{\hat{\Delta}_x(\omega_u, \chi_v)\}$. Then $Var(\hat{\Delta}_x(\omega_u)) = \{\Sigma_v \tau_v^{-2}\}^{-1}$. Typically, τ_v^2 needs to be estimated via the residuals from some local smoothing procedure. Finally, we fit a model of the form (2.10) applied to $\hat{\Delta}_x(\omega_u)$ by non-linear weighted least squares, where the weights are proportional to the inverse of $Var(\hat{\Delta}_x(\omega_u))$. If the various assumptions underlying our model are correct, the residual variance from this fit should be about 1. We should stress however that the assumption of separability of the autocovariance function is not critical to the success of this method. It can be interpreted as an indirect method of finding two-dimensional cutoff frequencies. When $|D_x| + |D_y| \geq 1$, we can estimate D_x and D_y to the nearest integer, using a two-dimensional version of the Hamon-Hannan procedure or by finding where the cross-covariance is maximised, shifting the images the appropriate number of pixels, trimming them and using the above procedure to estimate the fractional parts.

We have applied this method to a simulated image pair, each of size 200 x 200, in which there is no noise and for which $D_x = 0.2$ and $D_y = 0.4$. Details of the construction of these images can be found in Berman et al (1990, Section 3). Our estimates (and their standard errors) are D = .198 (.005), D = .398 (.007). We have also applied the

method to a number of real remotely sensed image pairs, and in most cases obtained comparable results. One example can be found in Berman et al (1990); others will be published elsewhere. In some cases, however, the method breaks down. This occurs when the two images are not highly correlated (in our experience, when the maximum cross-correlation between the two images is less than about 0.7). For remotely sensed imagery, this is typically because the wavelengths at which the two images are recorded are sufficiently far apart that the signals are no longer linearly related and so the two-dimensional version of (2.1) no longer holds. Consequently, care in the use of the method described here is required.

4 References

Berman, M., Green,A.A., Bischof, L., Davies, S.J. and Craig, M. (1990), A comparison of methods for estimating band-to-band misregistrations. Proc. 5th Australasian Remote Sensing Conf., 987-996.

Berman, M., Bischof, L., Craig, M., Davies, S.J. and Green, A.A. (1992), Estimating time delay in the presence of aliasing. Submitted for publication.

Bloomfield, P. (1976) Fourier Analysis of Time Series: An Introduction. New York: Wiley.

Chan, Y.T., Hattin, R.V., and Plant, J.B. (1978) The least squares estimation of time delay and its use in signal detection. IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-26, 217-222.

Hamon, B.V. and Hannan, E.J. (1974) Spectral estimation of time delay for dispersive and non-dispersive systems. Appl. Statist. 23, 134-142.

Hannan, E.J. (1975) Measuring the velocity of a signal. In Perspectives in Probability and Statistics (ed. J.M. Gani). Sheffield: Applied Probabilty Trust.

Hannan, E.J. and Thomson, P.J. (1988) Time delay estimation. J. Time Series Analysis, 9, 21-33.

AUTOMATIC MAGNETIC RESONANCE IMAGING

Katherine B. Ensor
Dept. of Statistics
Rice University
Houston, TX 77251-1892

Joe E. Ensor
Dept. of Patient Studies
Univ. of Texas M. D.
Anderson Cancer Center

Lalith Misra
Div. of Diagnostic Imaging
Univ. of Texas M. D.
Anderson Cancer Center

Abstract: Magnetic resonance imaging (MRI) is currently the most sensitive modality for detecting and differentiating pathophysiologic events. Transverse relaxation times ($T2$) provide quantitative information useful for evaluating a number of diseases (Dumitresco et al, (1986)). In MRI the observed T2 signal is modeled by $m(t) = \lambda(\sum_{j=1}^{k} \delta_j e^{-\alpha_j t})$ where the reciprocal of α_j is the corresponding expected relaxation time. We consider maximum likelihood estimation of the parameters λ, δ_j, α_j, $j = 1, \ldots, k$ under the assumption that the number of excited protons measured follows a Poisson distribution. A computationally simple method for selecting k, the number of exponential components in the model, is proposed.

1. INTRODUCTION.

Estimation of the individual parameters in the sum of exponential components is a long standing statistical problem (Niedzwiecki and Simonoff, (1990)) for which a solution is of fundamental interest in the field of MRI. The multicomponent exponential model associated with the T2 curve is derived in Section 2. Section 3 provides an overview of the fundamental estimation problems encountered with this model. Clearly, estimation of the model parameters is facilitated by knowledge of the correct model. In Section 4 standard model selection procedures are examined and a new method for model selection is proposed. Finally, in Section 5 the methods developed are applied to MRI data from an in vivo study of female breast tissue.

2. THE MODEL.

It is reasonable to assume the initial number of magnetized hydrogen molecules, $X(0)$, in the multi-compartment system follows a Poisson distribution with parameter λ. Further, we assume that the relaxation time of a molecule follows an exponential distribution with parameter α_j for $j = 1, \ldots, k$. Let $X_j(t)$ denote the number of excited molecules at time t of compartment j. Then the conditional joint distribution of $X_1(t), \ldots, X_k(t)$ given $X(0)$ is multinomial with parameters $\lambda, p_1(t), \ldots, p_k(t)$, where $p_j(t) = \delta_j \exp\{-\alpha_j t\}$ and $\sum_{j=1}^{k} \delta_j = 1$. In MRI, the random variable of interest is $Y(t) = a \sum_{j=1}^{k} X_j(k)$, the scaled signal, where a is a real-valued constant. We assume $a = 1$ for the purpose of this paper (consistent with cited authors), however, this parameter deserves future investigation. It then follows, that the marginal distribution of $Y(t)$ is Poisson with mean function

$$m(t) = \lambda \left(\sum_{j=1}^{k} \delta_j e^{-\alpha_j t} \right).$$

3. ESTIMATION OF T2 RELAXTION TIMES.

Given independent observations of $Y(t)$ at times $t_1, \ldots, t_n$ we focus on estimation of the parameters λ, δ_j, α_j for $j = 1, \ldots, k$, with $\alpha_1 > \ldots > \alpha_k$ and $\sum \delta_j = 1$. The expected T2 relaxation times are given by $1/\alpha_j$, $j = 1, \ldots, k$ and are the primary parameters of interest. The maximum likelihood estimates can be obtained by iteratively reweighted least squares with weight function $1/m(t)$ (see Frome, Kutner and Beauchamp (1973), del Pino (1989), Green (1984)).

Sandor et al (1988) derive the m.l.e. for a slightly different model of the decay curve of the transverse relaxation. They assume the observations are from a Poisson random variable with mean function given by $\int_I m(t)dt$ where $I \equiv (t_{j-1}, t_{j+1})$ denotes the time interval. Their formulation assumes the observations represent an accumulated response. Unfortunately the investigation of this model was limited to equally spaced time intervals and cannot be distinguished from a model based on time specific signal intensity. For the unequally spaced data in our example the accumulated model provides a very poor fit.

Although theoretically the above estimates are obtainable; realistically, solving the maximum likelihood equations is very difficult. The problems with fitting the sum of exponential components are well documented in Bates and Watts (1988) and Seber and Wild (1989). One major problem is that of parameter redundancy; in other words, models of different order produce similar results. Based on Reich's (1979) measure for parameter redundancy one cannot reliably estimate the parameters of a biexponential model if the ratio of the decay rates is less than .2. His measure, however, was developed for an additive error model with equally spaced observations. In MRI one expects the mean times of the long and short components to differ by less than a factor of five, so

Reich's measure would imply that estimation of these means is futile. Sandor et al (1988) show that for the Poisson model one can reliably estimate both the short and long expected times when the ratio of the two is as low as 1.1.

4. HOW MANY COMPARTMENTS?

Our objective in this paper is to introduce a noninteractive method for identifying the number of compartments which should be included in the model (again we examine either one or two compartment models). To this end, we have considered two approaches: use of standard model selection procedures for additive error models and development of a graphical method for model selection.

4.1 Standard Approaches for Model Selection.

Assume $Y(t)$ is modeled by $y(t_i) = m(t_i) + \epsilon_i$ where ϵ_i for $i = 1, \ldots, n$ are independent Normal random variables with zero mean and variance σ_i^2. Hurvich and Tsai (1989) propose a corrected form of Akaike's Information Criterion (AIC) (Akaike, 1973) for purposes of model selection in both linear and nonlinear regression when dealing with small sample sizes. For nonlinear regression with nonconstant variance their criterion is

$$\mathrm{AIC}_c = n \ln \hat{\sigma}^2 + n \frac{1 + m/n}{1 - (m+2)/n},$$

where $\hat{\sigma}^2$ is the maximum likelihood estimate of σ^2 and m is the number of parameters in the model. The model selected is the model which minimizes AIC_c. For 16 observations (which is the number of observations in our examples) $\hat{\sigma_1}^2/\hat{\sigma_2}^2 > 1.40$ to select a two-compartment model based on this criterion. For the uncorrected AIC a two-compartment model is indicated if $\hat{\sigma_1}^2/\hat{\sigma_2}^2 > 1.28$. The results of simulations based on both model selection procedures are given in Table 1.

Table 1. Model Selection for Additive Error Model

	%Correct	
$m(t)$	AIC	AIC_c
$300(.5e^{-t/30} + .5e^{-t/130})$	95	95
$300(.5e^{-5/20} + .5e^{-t/150}$	92	88
$500(.8e^{-t/30} + .2e^{-t/130})$	100	100
$450e^{-t/30}$	80	88

The above percentages are out of 25 replications.

In addition to the dependence on an additive error model, a major drawback to the AIC or corrected AIC selection criterion is that both the monoexponential and biexponential models must be fitted to the data. Frequently when the incorrect model is fitted, achieving convergence of the optimization routines is difficult, thus these methods are undesirable. In the next section we propose a method of model selection which does not require fitting the models.

4.2 Suggested Approach for Model Selection.

Consider the biexponential model observed at times ranging from 20 to 300 units. The signal attributed to the "short" component will decay more rapidly than the signal of the "long" component; therefore, the long component should dominate the signal at the larger times. This is the fundamental argument given for obtaining estimates of the expected T2 times by the method of peeling (see Bates and Watts (1988)). If the true mean function $m(t)$ contains more than one exponential term but a monoexponential model is fitted to the function, the monoexponential decay rate is a monotonic decreasing function of time. More specifically, setting

$$e^{-\beta_0 t} = \delta e^{-\alpha_1 t} + (1 - \delta) e^{-\alpha_2 t}$$

and solving for β_0 we obtain

$$\beta_0(t) \equiv \beta_0 = -\frac{\ln(\delta e^{-\alpha_1 t} + (1 - \delta) e^{-\alpha_2 t})}{t}.$$

Figure 1 is a plot of $\beta_0(t)$ for expected T2 times of 30 and 125 with three different mixtures (20% long and 80% short; 50% long and 50% short; 80% long and 20% short); Figure 2 presents $\beta_0(t)$ with expected T2 times of 30 and 65 for the same mixtures. For MRI data we would be interested in the $\beta_0(t)$ curve up to time 300. Clearly, when the short component contributes at least 50% of the signal this curve is distinguishable from the constant curve exhibited by a one compartment model. As expected, when the long component dominates the signal it would be difficult to make a distinction between a one and two compartment model. If we observe these patterns in sample data, then we should be able to distinguish between one and two compartment models.

Furthermore, the authors wish to note that even when the signal is dominated by the long component in a two compartment model and there exists reasonable separation of the two expected relaxation times, care should be taken in using methodologies which attribute the tail data to the long component. As shown in Figure 1, the curve with an 80% decay rate of .008 appears flat by time 250, but the value of $\beta_0(t)$ at this point is .00889 ($\beta_0(400) = .00856$). This 11% increase could understandably cause bias in estimates based on the monoexponential model at the larger times and this is a model which is dominated by the long component.

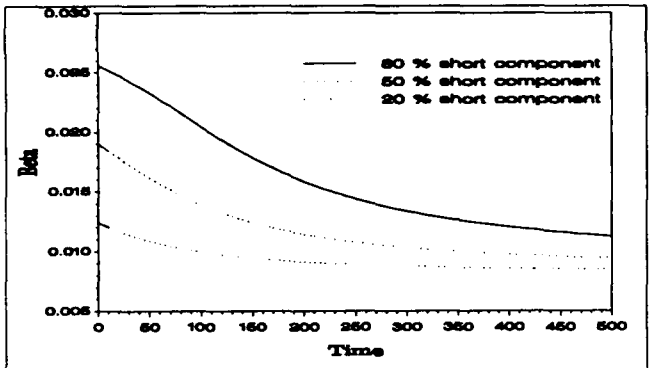


Figure 1. Beta(time) with decay rates of .08 and .008

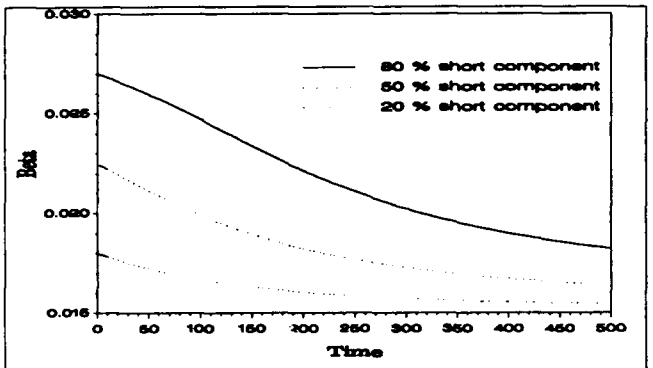


Figure 2. Beta(time) with decay rates of .08 and .015

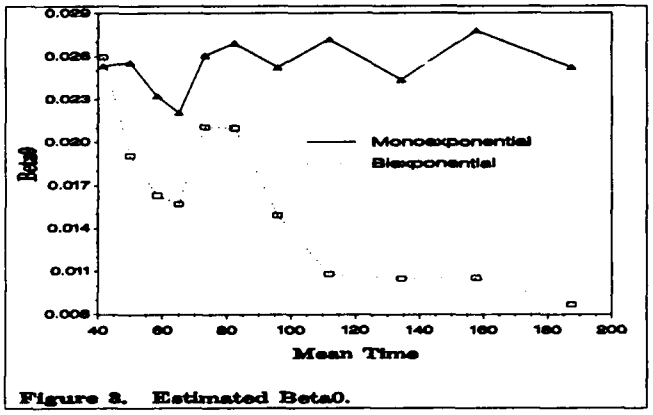


Figure 3. Estimated Beta0.

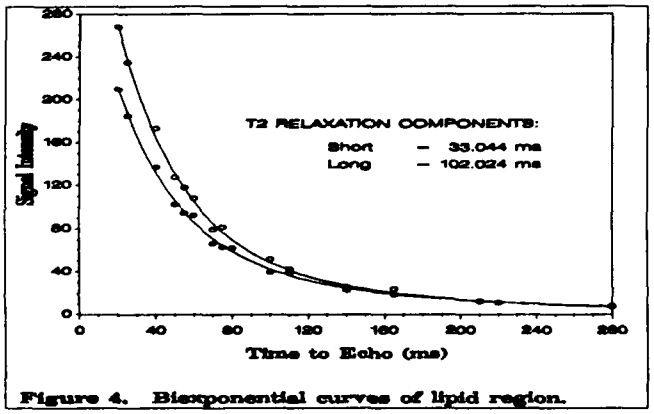


Figure 4. Biexponential curves of lipid region.

As a computationally simple estimate of $\beta_0(t)$ we propose using the slope from a simple linear model fitted to consecutive observations $z_{t_i}, \ldots, z_{t_{i+q}}$ where $z_i = \ln y(t_i)$ for $i = 1, \ldots, n$. In other words, as an estimate of $\beta_0(t)$ at $t_i^* = \frac{1}{(q+1)} \sum_{j=i}^{i+q} t_j$ we suggest

$$\hat{\beta}_i(t_i^*) = \frac{\sum_{j=i}^{i+q}(z_j - \bar{z}_i)(t_j - t_i^*)}{\sum_{j=i}^{i+q}(t_j - t_i^*)^2} \quad i = 1, \ldots, n-q$$

where $\bar{z}_i = \frac{1}{(q+1)} \sum_{j=i}^{i+q} z_j$.

The choice of q will depend on the variation expected in the data. For problems with large variation we do not want an estimate of $\beta_0(t)$ to be based on just two or three observations; however, if q is chosen too large then we are unable to identify the change in the decay rate (e.g. consider the extreme case when $q = n - 1$).

In Figure 3, the above estimates of $\beta_0(t)$ for simulated Poisson data from a two compartment model with mean $m(t) = 500(.8e^{-t/30} + .2e^{-t/130})$ and a one compartment model with mean $m(t) = 1000e^{-t/40}$ are plotted over t_i^*. For these examples the slope is computed from 6 consecutive points (i.e. $q = 5$). Clearly, the two compartment model is distinguished from the one compartment model. Future investigation into the use of tests of randomness to distinguish one and two compartment models is warranted.

An added advantage of our procedure is that the last estimate of $\beta_0(t)$ can be used as an initial estimate of the long expected T2 time for input into the optimization routine used to solve for the maximum likelihood estimates. The final slope estimate corresponds to the estimate of the long component obtained by the method of peeling when the same number of observations are used. As previously stated, this initial guess will be an underestimate of the long expected time since the short component still contributes substantially to the decay rate at time 200 (see Figures 1 and 2). In our example, the estimate of $\beta_0(t)$ at $t_{11}^* = 187.5$ is .008666 which would provide an initial guess of 115.39 for the long expected time of 130.

5. EXAMPLE.

The above methods were applied to MRI data from an in vivo study of the female breast. Our suggested method of order selection indicates that the signal from the ductal regions of the breast is best modeled by the monoexponential decay curve, whereas the data from the lipid region suggests a two compartment model. Figure 4 presents data observed at two sites in the lipid region of one patient and the estimated biexponential decay curves.

REFERENCES

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory*, Ed. B.N. Petrov and F. Csaki, pp. 267-81. Budapest: Akademia Kiado.

Bates, D. M., and Watts, D. G. (1988). *Nonlinear Regression Analysis and its Applications.* Wiley: New York.

del Pino, G. (1989). The unifying role of iterative generalized least squares in statistical algorithms. Statistical Science, **4**:394-408.

Dumitresco, B.E., Armspach, JP., Gounot, D., Grucker, D., Mauss, Y., Steibel, J., Wecker, D., and Chambron, J. (1986). Multi-exponential analysis of T_2 images. Magnetic Resonance Imaging, **4**:445-448.

Frome, E.L., Kutner, M.H., and Beachamp, J.J. (1973). Regression analysis of Poisson-distributed data. *JASA*, **68**:935-940.

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. JRSS-B, **46**:149-192.

Hurvich, C. M., and Tsai, C-L. (1989). Regression and time series model selection in small samples. Biometrika, **76**:297-307.

Niedzwiecki, D. and Simonoff, J. S. (1990). Estimation and inference in pharmacokinetic models. Journal of Pharmacokinetics and Biopharmaceutics, **18**: 361-377.

Reich, J. G. (1979). On parameter redundancy in curve fitting of kinetic data. *In L. Endrenyi (Ed.), Kinetic Data Analysis: Design and Analysis of Enzyme and Pharmacokenetic Experiments*, pp. 39-50. Plenum Press: New York.

Sandor, T., Bleier, A. R., Ruenzel, P.W., Adams, D. F., and Jolesz, F. A. (1988) Application of the maximum likelihood principle to separate exponential terms in T_2 relaxation of nuclear magnetic resonance. Magnetic Resonance Imaging, **6**:27-40.

Seber, G. A. F., and Wild, C. J. (1989). *Nonlinear Regression.* Wiley: New York.

A Practical Approach to Estimating the True Effect of Exposure Despite Imprecise Exposure Classification

James J. Weinkam, D.Sc., Wilfred L. Rosenbaum, M.Sc., and Theodor D. Sterling, Ph.D.
School of Computing Science, Simon Fraser University,
Burnaby, B.C., Canada, V5A 1S6

Introduction

Computation of relative risk of mortality or morbidity associated with exposure to some causative agent proceeds under the implied assumption that the exposure is measured with precision. In reality, accurate exposure information may or may not be obtainable. For instance, many surveys that determine the smoking status of individuals in a population appear to do so with acceptable accuracy. However, occupational exposure to many toxic substances may be much more difficult to ascertain with similar accuracy. It is simply not practical to measure the actual exposure directly on a broad scale or on a continuous basis. Consequently, some form of indirect or surrogate measurement must be employed.

A number of techniques have been developed to assign exposure levels when physical exposure measurements are absent [Checkoway, 1986; Dement et al, 1983]. For the most part these techniques rely on estimated exposure levels related to job categories, to location of workers, and/or to extrapolation from present day measurements of exposure to previous practices of manufacture. Despite the great amount of effort devoted to divising and improving these techniques, the problem remains that inevitably a proportion of individuals classified as exposed by the surrogate actually have little or no real exposure and at the same time a proportion of those classified as not exposed nevertheless have substantial exposure [Gerin et al, 1983, 1985; Siemiatycki et al, 1981; Hoar et al, 1980; Dosemeci et al, 1990; Esmen, 1979; Greenberg et al 1981, 1983]

Only a few studies attempt to estimate the proportion of individuals misclassified. Williams et al [1984] observe that assignment by job title of "Undertaker" to a category of "Exposed to Formaldehyde" would result in approximately 25% misclassification. Schulz et al [1983] report that 30% of individuals who were classified according to seriousness of complications had been misclassified. Millar [1986] reports approximately 12% misclassification of individuals who were classified by self-reported heights and weights. Millar's observation seems especially relevant because certainly it is likely that individuals would know their heights and weights with greater accuracy than their possible exposure to toxic products.

Even where physical exposure data exist classification may be inaccurate because such data are associated with a location, not an individual and workers typically move about their work area [Berode et al, 1980a, 1980b; Boillat et al, 1986; Cope et al, 1979; Sterling, 1964].

The existence of these misclassifications and the fact that commonly used methods of risk analysis essentially treat the data as if there were no misclassification raises two questions:

1. To what extent does the imprecise classification of individuals affect the calculated relative risk (or *Apparent Relative Risk*)?
2. Is it possible to determine from the *Apparent Relative Risk* what the *True Relative Risk* might be under reasonable estimates of the amounts of misclassification?

Copeland et al [1977] and Goldberg [1975] present numerical examples of the effects of misclassification. Barron [1977] gives a formula for the true relative risk in terms of conditional probabilities which are related to the prevalence, sensitivity, and specificity for the population at risk and the decedents. Flegal et al [1986] gives a formula for the true relative risk in terms of the apparent relative risk and the prevalence, sensitivity, and specificity.

Approaches to the analysis of misclassification bias generally parameterize the problem in terms of the exposure prevalence and the sensitivity and specificity of the exposure classification. The bias resulting when a confounding variable is present is discussed in Greenland [1980] and in Greenland and Robbins [1985] primarily through numerical examples. A good overview of the effects of misclassification is given in Kelsey et al [1986].

The problem with the traditional parameterization is that the sensitivity, specificity, and proportion exposed are all unknown and unobservable. Estimating the sensitivity and specificity essentially involves expressing a judgement about the proportion of an unknown subpopulation (those truly exposed or not exposed) which has been correctly classified by the surrogate exposure variable. Instead we adopt an approach similar to that of Green [1983] and parameterize our analysis in terms of the proportion classified as belonging to the higher likelihood of exposure group (an observable quantity) and the proportions of the higher and lower likelihood of exposure groups which really have high or low exposure respectively (the predictive values of the positive and negative classification). While the predictive values are conceptually similar to sensitivity and specificity, as Green [1983] has pointed out, it may be more feasible for an investigator to determine or at least estimate bounds for these unknown quantities.

In the remainder of the report we derive an expression for the true relative risk in terms of the apparent relative risk, the proportion in the high exposure group, and the positive and negative predictive values.

Analysis

We start with the familiar method of computing relative risk. Let b be the background probability of occurence of the disease or cause of death of interest and let t be the true relative risk of exposed to unexposed. Let p be the proportion of the population at risk who are exposed. Then $b(1-p)$ and btp are the numbers of cases among the

unexposed and exposed respectively. (The absolute size of the population at risk, N, always cancels out when the relative risk is computed. Therefore we omit it for simplicity although the tables are labeled and formulas presented as if the factor N appeared in each cell.) These data are normally arranged in a two by two table as shown in Figure 1 and the relative risk is given by the cross product ratio, which is equal to t.

Next we consider the case in which the exposure variable is imprecisely classified. Let b and t be defined as before, and let p be the proportion of the population at risk in the higher likelihood of exposure group and l and h be the (unknown) proportions of correct classification in the lower and higher groups respectively. Now $b(1-p)(l+(1-l)t)$ and $bp(1-h+ht)$ are the numbers of cases in the lower and higher groups respectively. We arrange the data in a two by two table as shown in Figure 2. Taking the cross product ratio we obtain the apparent relative risk

$$a = \frac{(1-h) + h\,t}{l + (1-l)\,t}$$

Solving for t (the actual risk of exposure), we obtain

$$t = \frac{a\,l - (1-h)}{h - a\,(1-l)} = M(a,h,l)$$

where we define $M(a,h,l)$ as the misclassification function which gives the true relative risk that is required to be consistent with given values of a, h, and l. Under the assumption that the higher group really is more likely to be exposed than the lower group, $0 < (1-l) < h < 1$. Consequently, $0 < (1-h) < l < 1$. Therefore, the numerator has a root at $(1-h)/l$ between 0 and 1, while the denominator has a root at $h/(1-l)$ which is greater than 1. Thus the graph of t increases steadily from the point $((1-h)/l,0)$, passes through the point (1,1), and approaches a vertical asymptote at $h/(1-l)$. Figure 3 shows a graph of $M(a,.6,.8)$ i.e., for the situation in which 40% of the high likelihood of exposure group and 20% of the low likelihood of exposure group have been misclassified.

Next suppose that exposure classification is precise but there is a dichotomous confounding factor. Let c be the relative risk of the confounder and s the relative synergistic (or antagonistic) effect between the true exposure and the confounder. In other words, persons exposed to the confounder but not the agent have c times the risk of persons exposed to neither, while persons exposed to both the confounder and the agent have cs times the risk of persons exposed to the agent alone.

Let p be the proportion of the population at risk who are exposed to the agent, u, the proportion exposed to the confounder, and f be the proportion exposed to both the agent and the confounder. Then the proportion exposed to the agent alone is $p-f$, the confounder alone is $u-f$, and to neither is $1-p-u+f$. These data can be arranged in a table as shown in Figure 4. The number of cases arising in each subgroup is shown in Figure 5. We can compute several forms of risk ratio standardized across levels of the confounder. Let a denote the computed SRR. Standardizing to the low exposure population group gives

$$a = \frac{c\,s\,(f-u) + p + u - f - 1}{c\,(f-u) + p + u - f - 1}\,t$$

and solving for t yields

$$t = \frac{c\,(f-u) + p + u - f - 1}{c\,s\,(f-u) + p + u - f - 1}\,a = S_l a$$

Similarly, standardizing to the high exposure population group yields

$$t = \frac{c\,f - f + p}{c\,f\,s - f + p}\,a = S_h a$$

Finally, standardizing to the total population yields

$$t = \frac{c\,u - u + 1}{c\,s\,u - u + 1}\,a = S_p a$$

In summary, the true relative risk of exposure, t, is equal to the apparent relative risk, a times a factor which depends on c, s, f, u, and p. If there is no synergistic effect, i.e., $s=1$, then as is well known the apparent relative risk is identical to the true relative risk for all values of all the other parameters. We call this factor the synergism factor and denote it by S_x where x indicates the referent population.

Finally, we consider the combined effect of misclassification and confounding. The distribution of the population at risk is the same as in the previous case (see Figure 4), except that Not Exposed and Exposed should now read Lower and Higher and p now denotes the proportion in the higher likelihood of exposure group. The number of cases arising in each subgroup is shown in Figure 6.

Standardizing to the entire population and solving for t yields $t = S_p\ M(a,h,l)$. Similarly, standardizing to the group with probable low exposure yields $t = S_l\ M(a,h,l)$. and standardizing to the group with probable high exposure yields $t = S_h\ M(a,h,l)$.

In a practical situation an estimate of a may be obtained in the usual way and a test for heterogeneity can be used to determine whether or not the assumption that $s=1$ is justified. If there is no synergism, then the range of possible values of t can be computed by assigning values or plausible ranges of values to h and l. If there is synergism, then it is not appropriate to attempt to summarize the effect of the agent in terms of a single relative risk. Instead a stratified analysis should be performed.

Discussion

Let us turn to an example. A study compares two groups of individuals, one of them classified as high exposed and the other classified as low exposed (H and L respectively). Assume that an apparent relative risk of 1.8 is computed for the H group as compared to the L group.

Figure 7 shows level curves of the function $M(1.8,h,l)$.

It is not unreasonable to assume that in a typical occupational health study at least 10% and possibly as many as 40% of individuals are incorrectly classified [Williams et al, 1984; Fergusson et al, 1989; Schulz et al , 1983; Millar, 1986]. Corresponding values of the true relative risk can be read off for different assumptions about misclassification. For instance, under the assumption of 10% misclassification in each category, an apparent relative risk of 1.8 would correspond to a true relative risk of approximately 2.1. If it is assumed that as many as 30% of individuals were misclassified in each group, a true relative risk of 6.0 would be needed to result in an apparent relative risk of 1.8.

The region in the lower left portion of Figure 7 may be considered the region of incompetence. It corresponds to situations in which the L group would be a better indicator of high exposure than the H group: in other words, inept choice of surrogate exposure variable. Such instances are unlikely to occur so this region may be ignored.

It is important to note that for certain combinations of misclassifications, an apparent relative risk of 1.8 (or any other value) could not be observed. For instance, if both the H and L groups have 50% misclassification, the apparent relative risk will be 1 regardless of the true risk. Therefore a low apparent relative risk may not be a reflection of absence of hazard but may simply be due to imprecise exposure classification.

For practical purposes, the approach suggested here may be used to set reasonable bounds between which the true relative risk may be assumed to lie given that an investigation has obtained a particular apparent relative risk. The amount of misclassification assumed to be operating may be set either by what appears to be reasonable (i.e., between 10% and 30%) or by relevant existing or obtainable information.

References

Barron B (1977): The effects of misclassification on the estimation of relative risk. Biometrics June:414-418.

Berode M, Guillemin MP, Martin B, Balant L, Fawer R, Droz PO, Madelaine P, Lob M (1980): Evaluation of occupational exposure to metallic mercury and its early renal effects. in: "Mechanism of toxicity and hazard evaluation." Elsevier/North-Halloand Biochemical Press, pp. 371-374.

Berode M, Boillat MA, Guillemin M, Droz PO (1980): Amelioration de surveillance biologique des travailleurs exposes au plomb inorganique. Internal Report, Swiss National Insurance.

Boillat MA, Berode M, Droz PO (1986): Surveillance de personnes exposees au perchloroethylene ou au styrene. Med Soc Prev 31:260-262.

Checkoway H (1986): Methods of treatment of exposure data in occupational epidemiology. Med Lav 77(I):48-73.

Cope RF, Pancamo BP, Rinehart, GL, Ter Haar, GL (1979): Personnel monitoring for tetra-alkyl lead in the workplace. Am Ind Hyg Assoc J 5:372-379.

Copeland KT, Checkoway H, McMichael AJ, Holbrook RH (1977): Bias due to misclassification in the estimation of relative risk. Am J Epidem 105(5):488-495.

Dement JM, Harris RL, Symons MJ, Shy CM (1983): Exposures and mortality among chrysotile asbestos workers. Part 1: Exposure estimates. Am J Ind Med 4:399-420.

Dosemeci M, Stewart PA, Blair A (1990): Three proposals for retrospective, semiquantitative exposure assessments and their comparison with the other assessment methods. Appl Occup Environ Hyg 5(1):52-59.

Esmen N (1979): Retrospective industrial hygiene surveys. Am Ind Hyg Assoc J 40:58.

Fergusson DM, Horwood LJ (1989): A latent class model of smoking experimentation. J Child Psychol Psych Allied Dis 30:761-773.

Flegal KM, Brownie C, Haas JD (1986): The effects of exposure misclassification on estimates of relative risk. Am J Epidem 123(4):736-751.

Gerin M, et al (1983): Translating job histories into histories of occupational exposure for epidemiologic purpose. in: Acheson ED. (ed) Job-Exposure Matrices. Proceedings of a conference held in April 1982 at the University of Southampton, Hobbs, The Printers of Southampton, Southampton. U.K., pp. 78-82.

Gerin M, Siemiatycki J, Kemper H, Begin D (1985): Obtaining occupational exposure histories in epidemiologic case-control studies. J Occup Med 27:420-426.

Goldberg J (1975): The effects of misclassification on the bias in the difference between two proportions and the relative odds in the fourfold table. JASA 70(351):561-567.

Green MS (1983): Use of predictive value to adjust relative risk estimates biased by misclassification of outcome status. Am J Epidem 117(1):98-105.

Greenberg RA, Tamburro CH (1981): Exposure indices for epidemiological surveillance of carcinogenic agents in an industrial chemical environment. J Occup Med 23:353.

Greenberg RA, Tamburro CH (1983): Rank ordered exposure for industrial surveillance. in: Acheson ED. (ed) Job-Exposure Matrices. Proceedings of a conference held in April 1982 at the University of Southampton, Hobbs, The Printers of Southampton, Southampton. U.K., pp. 52-62.

Greenland S (1980): The effect of misclassification in the presence of covariates. Am J Epidem 112(4):564-569.

Greenland S, Robbins JM (1985): Confounding and misclassification. Am J Epidem 122(3):495-506.

Hoar SK, Morrison AS, Cole P, Silverman DT (1980): An occupation and exposure linkage system for the study of occupation carcinogenesis. J Occup Med 22:722-726.

Kelsey JK, Thompson WD, Evans AS (1986): Methods in Observational Epidemiology. Oxford University Press, New York.

Millar WGA (1986): Distribution of body weight and height: Comparison of estimates based on self-reported and observed measures. J Epidem Commun Health 40:319-323.

Schulz KF, Cates WJR, Grimes DA, Selik RM, Tyler CW Jr (1983): Reducing classification errors in cohort studies: The approach and practical application. Stat Med 2:25-31.

Siemiatycki J, Day NE, Fabry J, Cooper JA (1981): Discovering carcinogens in the occupational environment: A novel epidemiologic approach. JNCI 66:217-225.

Sterling TD (1964): Epidemiology of disease associated with lead. Arch Environ Health 8:146-160.

Williams TM, Levine R, Blunden P (1984): Exposure of embalmers to formaldehyde and other chemicals. Am Ind Hyg Assoc J 45:172-176.

	Persons At Risk	Cases
Not Exposed	$(1 - p)$	$b\ (1 - p)$
Exposed	p	$b\ t\ p$

Figure 1. 2 X 2 Table for Precise Classification with no Confounder

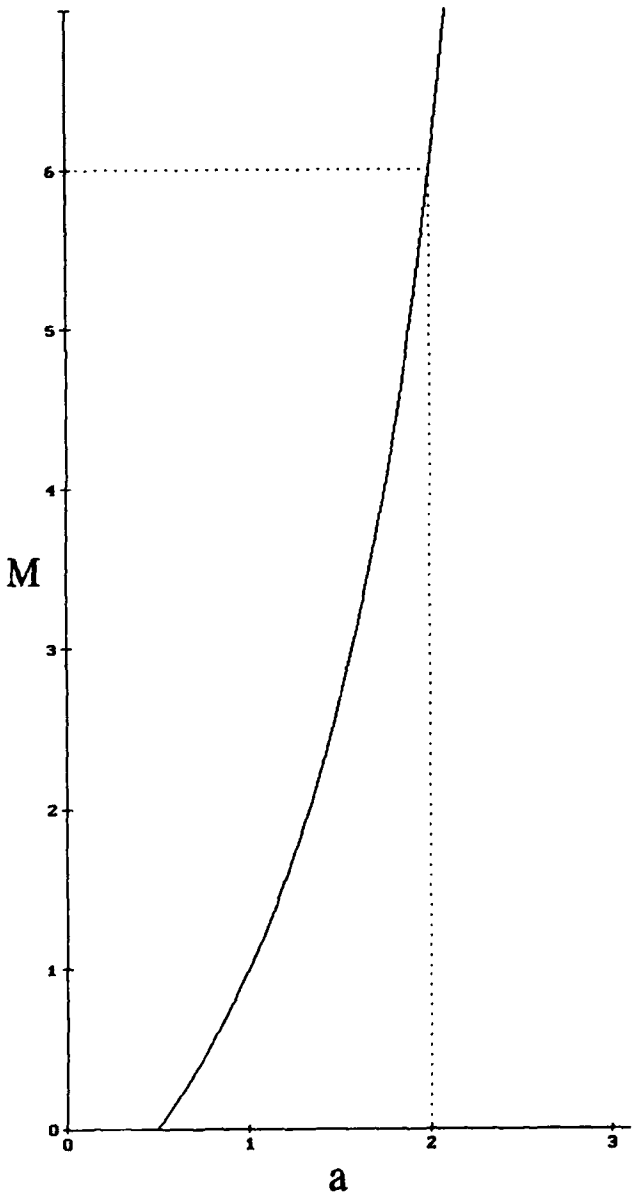


Figure 3. Graph of M(a,.6,.8), the *True Relative Risk* required to be consistent with an *Apparent Relative Risk*, a, if 60 percent of the High Group actually have high exposure (h=.6, i.e., 40 percent are misclassified) and 80 percent of the Low Group actually have low exposure (l=.8, i.e., 20 percent are misclassified).

	Persons At Risk	Cases
Lower	$1 - p$	$b\ (1 - p)\ (l + (1 - l)\ t)$
Higher	p	$b\ p\ (1 - h + h\ t)$

Figure 2. 2 X 2 Table for Imprecise Classification with no Confounder

	Population At Risk: Confounder Absent	Confounder Present
Not Exposed	$1 - p - u + f$	$u - f$
Exposed	$p - f$	f

Figure 4. Joint Distribution of Exposure and Confounding Variables

	Cases: Confounder Absent	Confounder Present
Not Exposed	$b\ (1 - p - u + f)$	$b\ c\ (u - f)$
Exposed	$b\ t\ (p - f)$	$b\ c\ t\ s\ f$

Figure 5. Cases Arising Assuming Precise Classification with Confounding

	Cases: Confounder Absent	Confounder Present
Lower	$b\ (1 - p - u + f)\ (l + (1 - l)\ t)$	$b\ c\ (u - f)\ (l + (1 - l)\ t\ s)$
Higher	$b\ (p - f)\ (1 - h + h\ t)$	$b\ c\ f\ (1 - h + h\ t\ s)$

Figure 6. Cases Arising Assuming Imprecise Classification with Confounding

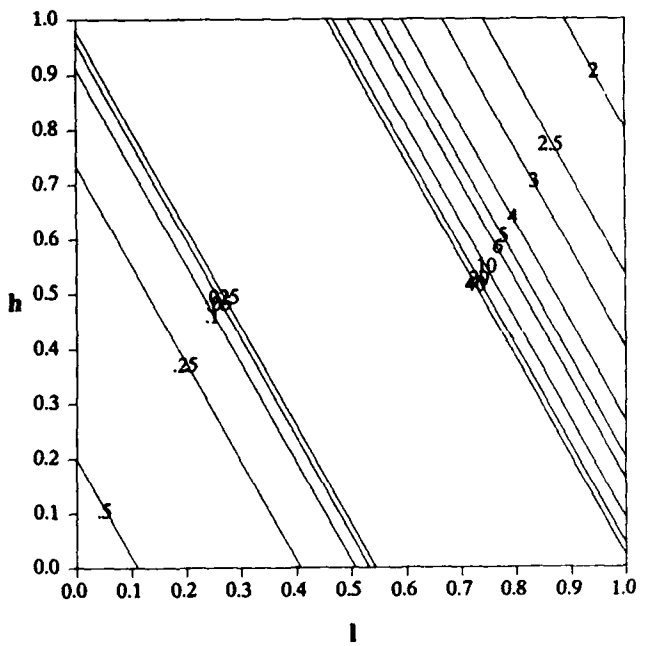


Figure 7. Level curves of M(1.8,h,l), the *True Relative Risk* required to be consistent with an *Apparent Relative Risk* of 1.8.

Relative Cancer Risk of Homemakers

T.D. Sterling*, W.L. Rosenbaum,* D.A. Sterling**, J.J. Weinkam*

*School of Computing Science, Simon Fraser University, Burnaby, British Columbia V5A 1S6
**College of Health Sciences, Old Dominion University Norfolk, Virginia, 23529

Introduction

Morton[1,2] reported significant excess cancer mortality rates among housewives compared to women employed outside the home. Subsequently, Sterling and Weinkam[3] reported that age-specific morbidity ratios for all chronic conditions were significantly larger for housewives than for employed women. These findings raise the question of whether the incidence as well as prevalence of chronic diseases, and in particular cancer, is larger in general among housewives than among women employed in other occupations. Such a difference would have important consequences not only for housewives' chronic disease and cancer morbidity but also for their cancer mortality insofar as morbidity is related to mortality. Verbrugge[4,5] has shown that while morbidity rates from chronic disease are higher for women than men, mortality rates from chronic diseases are higher for men than women. Among women, however, it seems likely that groups that tend to have higher morbidity rates for chronic diseases also may be expected to have higher mortality rates from the same diseases. The availability of large archives of population morbidity data such as those collected by the U.S. National Health Interview Survey (NHIS) makes it possible to compare morbidity of homemakers to that of women employed outside the home. Using public use tapes of the NHIS, we compared morbidity of homemakers with those of employed women for two blocks of time: 1970-1975 and 1982-1987.

Method

The NHIS collects information on a nationwide sample of households as part of the ongoing activity of the National Center for Health Statistics. Each week a sample of households is selected from the civilian, non-institutionalized, U.S. population using a stratified probability sampling technique in such a way that each weekly sample is representative of the target population and the weekly samples are additive over time.[6]

Public use tapes of the NHIS for 1970 to 1975 and 1982 to 1987 were used. Individuals were classified according to race (white, nonwhite), age (by 5-year age groups for ages 20 to 64) and occupation (employed outside the home and homemaker). Homemakers are those who indicated that their usual activity was "keeping house").

For purposes of making prevalence estimates of chronic conditions, a list of diseases was read to each NHIS sample member. The respondent reported his or her experiences with each disease on the list. During 1970 to 1977 one list per year was asked of all sample members. Prevalence estimates for a particular condition may be obtained only in the year in which the condition was probed. For each year 1982 to 1987 each annual sample was divided into one-sixth subsamples. All members of a particular subsample were asked to respond to one of the six lists.[7] Using these data it is possible to compute estimates of national prevalence rates for certain chronic conditions. However, only in 1982 and 1983 did the NHIS probe for the existence of any malignant neoplasm. Thus it is possible to obtain national prevalence estimates for any form of cancer based only on the 1982 and 1983 data (although for selected sites it is possible to use data for 1982 to 1987).

National estimates of the total number of persons and of prevalence rates for various causes were computed for each race-age-employment group, for each year and for all years combined. SPRs (Standardized Prevalence Ratios) were computed in a manner identical to the computation of the familiar Standardized Mortality Ratio[8] using the employed population as the referent. Variances were computed using the appropriate generalized variance function recommended by NCHS[9] and confidence intervals computed under the assumption of a log normal distribution for the SPR estimate.[8]

All analyses were done on weighted data. The National Center for Health Statistics weighting factors compensate for sampling variation within different sampling areas and adjust the data to the race-age-sex distribution of the non-institutionalized U.S. population as determined by the U.S. Current Population Survey.[6]

Results

Homemakers show an increased prevalence over employed women of each chronic condition investigated, with the lone exception of breast cancer. Table 1 gives estimates of SPRs and correpsonding 95% confidence intervals for chronic conditions for 1970 to 1975 and 1982 to 1987. Prevalence ratios larger than 1.0 indicate increased condition prevalence for homemakers compared with employed women. Hypertension, ischemic heart disease, stroke and combined bronchitis, emphysema and asthma (for 1982 to 1987 but not for 1970 to 1975) all exhibit statistically increased prevalence ratios. Each cancer site considered (except for breast cancer) and all cancers combined showed an increased prevalence ratio among homemakers. However, that increase fell short of the customary level of rejecting the null condition with $p \leq .5$, possibly due to the relative scarcity of data for these conditions. Because cancer SPRs for all sites but one are elevated and because the SPRs for 1970-1975 and 1982-1987 are very much alike, the cancer risks may be considered as significantly elevated as well. Our data then support the conclusion that women working at home have a significantly higher prevalence rate of all chronic conditions when compared with women working outside the home.

The excess prevalence of chronic conditions among homemakers relative to employed women may be due to the occupational exposures of homemaking or to a number of confounding factors. Some women may

have to select homemaking because they suffer from a chronic disease that prevents them from seeking and holding down full time employment.

The selection bias for adopting housework as an occupation because of already existing disease may be controlled for to some extent by adjusting each chronic disease for the difference between risks of homemakers and employed women risk for all chronic diseases. Such an adjustment may be simply done by dividing the risk ratios for Cancer, Heart and Other Chronic Diseases of Table 1 by the SPR for all chronic conditions. The result of that adjustment can be seen in Table 2. After this adjustment, the SPRs are still elevated for all conditions except for breast cancer and hypertension for the period 1970-1975 and for breast cancer but not hypertension for the period 1982-1987. Again we note that the SPRs were elevated for 6 out of 9 conditions in 1970-1975 and 9 out of 11 conditions in 1982-1987. The probability of that many increased SPRs arising from the population with similar prevalence of disease ten years apart by chance alone is vanishingly small.

Differences in prevalence rates are unlikely to be due to smoking. While our analysis combines NHIS data for the years 1970 to 1975 and 1982 to 1987, smoking information was obtained only for 1970 and 1987. However, for both years similar percentages of homemakers and otherwise employed women smoked (see Table 3). Weinkam and Sterling have shown a similar lack of difference in smoking prevalence between homemakers and otherwise employed women for the 1979-1980 NHIS.[10]

Discussion

Homemaking or housekeeping has not been and is not now generally considered an occupation. Historically, the keeping of the house and the care of children was considered woman's work, but not in the sense of an occupation. It was an obligation and duty, performed by the wife. Even in our advanced Western societies and in developing countries housework is still not recognized as an occupation entitled to some of the basic consideration of employment such as coverage by Social Insurance (Social Security in the U.S.), or by Workman's Compensation, or by pay (or even by a recognized commercial value).

Yet, housework has all the earmarks of an occupation. It is performed in a workplace, the home. It requires a number of skills, some of them rather intricate. The obligations of the worker can be described (including cooking, cleaning, washing, various types of yardwork, maintenance and repair, use of appliances etc.). Whether or not it is officially recognized, housework has a definite commercial value. The cost of replacing the houseworking spouse with paid help may be considerable, as the cost of care for the handicapped or the elderly proves.

Like all occupations, housework has its hazards. These hazards may be divided into two groups, that of occupationally related accidents, and that of chronic disease following exposure to toxic materials.

Homemaker's Potential Exposure to Toxic Substances

Table 4 (expanding listings by Gleason et al[11]) lists toxic components commonly found in the home. Perhaps the most serious exposure is to modern household cleaners. They are favorite household tools because they relieve the homemaker of considerable physical exertion. However, they expose the user to extremely toxic agents in parts of the dwelling that have usually the poorest air circulation. Another possibly very serious exposure may be to toxic material brought home on hair, skin and clothing by industrial workers who are members of the household. Such exposures have been shown to lead to specific illnesses that are distinctly related to occupational exposures. Cases in point are mesothelioma or beryllioses among family members of individuals employed in occupations where they may bring home asbestos or beryllium. These observations raise the possibility that other diseases of members of a household may not be recognized as being of occupational origin. Finally we include basements because, where there is a background basements are the major avenue for radon gas penetration. Homemakers can accumulate high levels of exposure because of the length of exposure.

The basis for our results is the statistical analysis of the National Household Interview Survey, and not of medically established cases. However, coupled with the observation that homemakers may be exposed substantially to carcinogens at home, very often in unventilated spaces and subjected therefore to repeated relatively large doses, the conclusion that homemakers are at an increased risk from Cancer compared with women in other employment seems plausible. (A more definitive answer will come from a case-control study underway and from a more detailed analysis of household use of toxic material.)

Acknowledgements

The work on this project is supported in part by a grant from B.A.T. Ltd.

References

1. Morton WE, Unga TJ: Cancer Mortality in the Major Cottage Industry. Women and Health 1979; 4:346-354.
2. Morton WE: Further Investigation of Housewife Cancer Mortality Risk. Women and Health 1982; 7:43-51.
3. Sterling TD, Weinkam JJ: The "Healthy Worker Effect" on Morbity Rates. Journal of Occupational Medicine 1985; 27:477-482.
4. Verbrugge LM: Recent Trends in Sex Mortality Differentials in the United States. Women and Health 1981; 5(3):17-37.
5. Verbrugge LM: Recent, Present, and Future Health of American Adults. Annual Review of Public Health 1989; 10:333-361.
6. National Center for Health Statistics, Current Estimates from the Health Interview Survey, Publication (HRA) 74-1054, Series 10, No. 72. Washington, D.C.: U.S. Department of Health, Education and Welfare, 1974.
7. National Center for Health Statistics, Survey Design 1973-1984 and Procedures 1975-1983. DHHS Pub. No. (PHS) 85-1320, Aug, 1985.
8. Rothman, K.J., Modern Epidemiology. Little, Brown and Co., Boston, 1986.
9. Wolter, K.M., Introduction to Variance Esimation, Springer-Verlag, New York, 1985.
10. Weinkam JJ, Sterling TD: Chanages in Smoking Characteristics by Type of Employment from 1970 to 1979/80. American Journal of Industrial Medicine 1987; 11:539-561.
11. Gleason MN, Gosselin RE, Hodge HC, Smith RP: Clinical Toxicology of Commercial Products, 3rd Edition, Baltimore, Williams & Wilkens, 1969.

TABLE 1

Standardized Prevalence Ratios and 95% Confidence Intervals For a Number of Chronic Conditions of Homemakers Standardized to Women Employed Outside the Home

	1970 - 1975		1982 - 1987	
		(LC, UC)		(LC, UC)
Any Cause	1.07	(1.01, 1.13)	1.17	(1.11, 1.23)
Any Chronic Condition	1.11	(1.05, 1.18)	1.21	(1.15, 1.28)
Hypertension	1.31	(1.17, 1.47)	1.44	(1.27, 1.64)
Ischemic Heart Disease	1.63	(1.17, 2.29)	1.63	(1.18, 2.24)
Stroke	3.29	(1.50, 7.21)	3.25	(1.84, 5.73)
Bronchitis/ Emphysema/Asthma	1.10	(0.97, 1.25)	1.23	(1.07, 1.42)
Any Cancer	1.58	(0.88, 2.86)	1.46	(0.76, 2.82)
Breast Cancer	0.99	(0.33, 2.57)	0.95	(0.53, 1.73)
Lung Cancer		N/A	3.41	(0.43, 26.91)
Genital Urinary Cancer	2.24	(0.16, 31.53)	1.84	(0.59, 5.67)
Leukemia	1.46	(0.04, 52.74)	1.92	(0.12, 31.3)
Cancer of Digestive Organs	3.16	(0.41, 25.59)	2.60	(0.70, 9.62)
Respiratory Cancer		N/A	4.27	(0.57, 32.10)

TABLE 2

Standardized Prevalence Ratios and 95% Confidence Intervals For a Number of Chronic Conditions of Homemakers Standardized to Women Employed Outside the Home Adjusted for Differences in Standardized Prevalence Ratios For Any Chronic Condition

	1970 - 1975		1982 - 1987	
		(LC, UC)		(LC, UC)
Any Chronic Condition	1.00		1.00	
Hypertension	0.95	(0.84, 1.09)	1.19	(1.038, 1.37)
Ischemic Heart Disease	1.19	(0.85, 1.68)	1.34	(0.97, 1.86)
Stroke	2.40	(1.09, 5.28)	2.68	(1.52, 4.75)
Bronchitis/ Emphysema/Asthma	1.00	(0.86, 1.14)	1.02	(0.88, 1.19)
Any Cancer	1.16	(0.65, 2.10)	1.04	(0.71, 1.50)
Breast Cancer	0.89	(0.34, 2.32)	0.79	(0.43, 1.43)
Lung Cancer		N/A	2.82	(0.36, 22.27)
Genital Urinary Cancer	1.52	(0.18, 13.70)	1.52	(0.93, 4.69)
Leukemia	1.07	(0.03, 38.50)	1.59	(0.10, 25.91)
Cancer of Digestive Organs	2.14	(0.27, 16.62)	2.15	(0.58, 7.96)
Respiratory Cancer		N/A	3.53	(0.47, 26.56)

TABLE 3

Percent of Current, Former, Ever and Never Smokers Among White Homemakers and Otherwise Employed Women for 1970 and 1987

1970	Current	Former	Ever	Never
Homemakers	29.3	12.6	41.9	58.1
Otherwise Employed	35.0	11.9	46.9	53.1
1987				
Homemakers	31.7	15.8	47.5	49.9
Otherwise Employed	32.3	15.8	48.1	49.7

TABLE 4

Homemaker's Potential Exposure to Toxic Substances from Use of Household Consumer Products, Appliances and Other Activities

Item Use	Possible Components May Include:
Household cleaners:	
Window	Ammonium hydroxide
Spot/Textile	Tetrachloroethylene, Trichloroethylene, Methyl alcohol,Petroleum derived solvents, Methanol, Benzene
Soaps/Detergents	Polyether sulfates, Alcohols, Sulfonates, Alkyl sodium isothianates
Oven	Sodium hydroxide, Potassium hydroxide
Drain/Toilet bowl	Sodium hydroxide, Lye
General cleaning	Ammonium hydroxide, Chlorine, Lye, Sodium hypochlorite, Sodium peroxide
Home Repair/Maintenance	
Paints/Varnish	Toluene Xylene, Methylene Chloride,Heavy metal pigments, Methanol, Ethylene glycol, Benzene
Pesticides	Organophosphates, Carbamates, Pyrethroids, Botanicals (plant derivatives), Biological, Growth regulators
Lawn & Yard Care Items	Various pesticides, herbicides, gasoline, oil, paints, fertilizers
Appliances	
Humidifiers	Off-gassing from water components,chlorinated contaminants, Biological organisms
Gas Range & Heater	NO2, CO, Formaldehyde
Kerosene Heaters	NO2, CO, SO2, General petroleum hydrocarbons
Electrical Equipment	Ozone
Spray Aerosol	Propane, Butane, Nitrous Oxide, Methylene propellents, Chloride, Isobutane, Fluorocarbon 11 and 12
Disinfectants	Sodium hypochlorite, Quarternary ammonium, Phenols, Pine oils
Furniture/Carpets Offgassing	Formaldehyde, General organics, Residues
Basement	Radon daughters (depending on background and ventilation)
Smoking	Environmental Tobacco Smoke
Washing Clothing	Toxic material brought home on clothing by employed member of household (such as asbestos, beryllium etc.)
Hobbies	Depends on the hobby
Beauty/Grooming Aids	Alcohol, sodium hydroxide, Thioglycollates, Talc, Benzethonium

Application of a Random Choice Method to Small Amplitude 2D Shockwaves

Gholam-Ali Zakeri
Mathematics Department
California State University-Northridge
Northridge, Ca 91330

Abstract

Sampling techniques and exact solutions of Riemann Problems are used in a random choice method. This procedure is used to obtain the numerical solutions of a system of conservation laws which describes the dynamics of flow for small amplitude two-dimensional shockwaves. An intrinsic coordinate system is used to formulate the model.

1 Introduction

Accuracy of numerical solutions and efficiency of numerical schemes are major concerns in obtaining numerical solutions. Moreover, the numerical solution at the jump discontinuities called shocks should remain sharp, stable and transports the discontinuities at the correct physical speed. Random variables have been used to control numerical dissipation or to control numerical viscosity. Basically, random variables appear either as a component added to the deterministic equation to study the effect of numerical viscosity or they are used to sample the solution at a randomly chosen point to obtain a numerical solution which preserves some mathematical properties of the solution function. The purpose of this paper is to present a random choice method for computing the numerical solution of two-dimensional small amplitude shockwaves. The numerical random sampling procedure is a shock capturing and a marching time method for solving system of conservation laws. The random sampling procedure consists of approximating the numerical solution by a piecewise constant state at each time step and proceeding to the next time step by solving the corresponding problems formed by the constant on the neighboring spatial intervals. It is well-known that the exact solution of non-linear system of partial differential equations arising in fluid flow problems even with smooth initial data develops shocks (jump discontinuities) in a finite time interval. Thus it is not unnatural to approximate their initial data with constant states.

The sampling procedure is based on approximating the numerical solution of the given problem with a sequence of elementary problems, known as the Riemann problems. These Riemann problems can be thought of as information source about the solution within each spatial mesh interval. More importantly they provide valuable information on wave interaction.

Godunov [1] initiated utilizing the solutions of the Riemann problems as building blocks for the construction of numerical solution of the nonlinear hyperbolic partial differential equations. Godunov replaced the initial data by a piecewise constant states with jump discontinuities at the middle of spatial mesh interval. Then the exact solution of this Riemann problem at the first time step is calculated. To proceed to the next time step replace this exact solution by a new piecewise constant state approximation and solve the corresponding Riemann problem and maintain integral properties of the conserve variable.

Another utilization of Riemann problems in obtaining the solution of conservation laws was initiated by Glimm [2] who followed Godunov as far as obtaining the exact solution of Riemann problem and then the value of the new approximated solution at the new time step is taken to be the exact solution evaluated at a random point on that mesh interval. This solution is conservative on the average, however, has the advantage that near jump the solution is incremented either by the amount of jump or not at all. This forces that an initially sharp discontinuities remains sharp. Chorin [3] developed Glimm's random choice method into a numerical technique. The random choice method by its way of construction propagates shocks without introducing any dissipation and the method is unconditionally stable. However, because of approximating solution at a randomly chosen point a small amount of statistical noise enters into the solution which is acceptable within the accuracy imposed by discretization of model problem.

2 Two-Dimensional Flow Problem

The equations describing the two-dimensional flow of shockwaves with a source term in fluid dynamics for compressible fluid may be written in the form

$$(1) \qquad u_t + f(u)_x + g(u)_y = h(u, x, y, t)$$

where f and g are physical fluxes, h is a source term and the unknown quantity, u is a function of x, y, t. Denoting the front coordinate by α and letting the coordinate β be

the arc length measured from a reference point along the front, then the successive front positions are given by the family of curves, α = constant and the ray positions by the family of curves, β = constant. By using this intrinsic coordinate system α β (see Whitham [8]) where α and β are functions of x, y, t, equation (1) can be written as

$$w_\alpha + F(w)_\beta = G(w, \alpha, \beta) \tag{2}$$

subject to the initial condition given by

$$w(0, \beta) = w_0(\beta) . \tag{3}$$

Equations relating x and y to α and β are given by

$$x_\alpha = (1 + \tfrac{1}{2} m\phi)\cos(\theta) \quad x_\beta = -A\sin(\theta)$$

$$y_\alpha = (1 + \tfrac{1}{2} m\phi)\sin(\theta) \quad y_\beta = A\cos(\theta)$$

Here θ is the angle that each front makes with the positive x-axis, A is the cross-sectional ray-tube area, and m is the acoustic Mach number. For small amplitude two-dimensional shockwaves we have

$$w = \begin{pmatrix} A \\ A\theta \\ m\sqrt{A} \end{pmatrix} \qquad F(w) = \begin{pmatrix} -\theta \\ (m\phi - \theta^2)/2 \\ 0 \end{pmatrix}$$

$$G(w) = \begin{pmatrix} 0 \\ 0 \\ -\dfrac{\phi A m^3}{4CZ} \end{pmatrix} \tag{4}$$

where C is the local sound speed, ϕ is the nonlinearity constant which depends on the media and Z is the area under the initial pulse. For a detail discussion of these equations see Zakeri [4]-[5]. To solve (2)-(3) we use operator splitting method to remove the inhomogeneous term G(w, α, β). That is, first we solve the corresponding one-dimensional homogenous problem,

$$w_\alpha + F(w)_\beta = 0 \tag{5}$$

by sampling procedure and then we use its solution to determine the value of the inhomogeneous term, G(w, α, β). Finally, we solve the corresponding ordinary differential equations (ODEs) given by

$$w_\alpha = G(w, \alpha, \beta) . \tag{6}$$

To solve (6) we use a common ODE solver such as Runge-Kutta or a multi-level method.

3 Numerical Scheme

We develop a numerical scheme to compute the successive shock fronts using geometrical shock dynamics given by (2)-(3). One of the advantages of formulation of a model problem using geometric shock dynamics is its simplicity. To develop a random choice method first we must define a random variable defined over closed interval [-½, ½]. It is absolutely necessary that the successive values of the random variable tend to approximate equi-partitioning the closed interval [-½, ½] (see Glimm [2]). To generate such random variable let us consider a sequence of pseudorandom integers generated by

$$N_{n+1} = N_n + \left[\frac{3-\sqrt{3}}{6} k\right] \pmod{k} \tag{7}$$

where k is an odd positive integer and N_0 is an arbitrary integer less than k. Let us define an equidistributed sequence random variables, σ_n on the interval [-½ , ½] given by

$$\sigma_n = \frac{N_n}{k} - \frac{1}{2} . \tag{8}$$

We introduce front-ray grid defined by mesh lengths $\Delta\alpha$ and $\Delta\beta$. The solution of (2)-(3) is to be calculated both at grid points, i.e., at

$$P(n, j) = (n\Delta\alpha , j\Delta\beta)$$

and at the center of rectangle grid point, i.e., at

$$P(n+\tfrac{1}{2} , j+\tfrac{1}{2}) = ((n+\tfrac{1}{2})\Delta\alpha , (j+\tfrac{1}{2})\Delta\beta)$$

where n and j are integers. We denote the approximate value of w at the grid point by $w_j^n = (n\Delta\alpha , j\Delta\beta)$.

Following the outline given above, let us consider the corresponding local Riemann problem to (2) when front is at $n\Delta\alpha$ along with the piecewise constant initial data given by

$$R_\alpha + F(R)_\beta = 0 \tag{9}$$

$$R(n\Delta\alpha , \beta) = \begin{cases} w_{j-1+\varepsilon}^n & ; \ \beta < J\Delta\beta \\ w_{j+\varepsilon}^n & ; \ \beta \geq J\Delta\beta \end{cases}$$

where

$$J = j - \tfrac{1}{2}(-1)^\varepsilon$$

$$\varepsilon = \tfrac{1}{2}(1 + sgn(\sigma_{n+1}) \cdot 1)$$

i.e. ε = 0 or 1 whenever σ_{n+1} is negative or non-negative respectively. The Riemann problem here is sampled at $(j+\tfrac{1}{2})\Delta\beta$ and at $(j-\tfrac{1}{2})\Delta\beta$.

When σ_{n+1} is non-negative the initial data for Riemann problem formed by using information at grid points P(n,j) and P(n,j+1) and if σ_{n+1} is negative then the initial data is constructed by using information at grid points P(n,j) and P(n,j-1). At point P(n+½, j+½) we define

$$w_{j+1/2}^{n+1/2} = R((j+\sigma_{n+1})\Delta\beta,\ (n+1/2)\Delta\alpha)$$

On each mesh interval we get a local Riemann problem. In order to assure that the waves produced by this sequence of local Riemann problems do not interact we must have

$$\frac{\Delta\beta}{\Delta\alpha} C(1+\tfrac{1}{2}m\phi) < 1. \tag{10}$$

This important requirement is known as Courant-Friedrichs-Lewy (CFL) condition. If inequality (10) holds then we can combine the solutions of the Riemann problems (9) into a single exact solution.

4 Solution of Riemann Problem

The main part of a random choice algorithm is obtaining the solutions of a sequence of local Riemann problems efficiently. The solution of a Riemann problem consists of three elementary waves, a backward shock wave or rarefaction on left, a slip line, and a forward shock or a rarefaction on right. A slip line is a discontinuous solution separating two constant states such that the angle of flows remain the same on both sides of the discontinuity line while Mach number is arbitrary. Slip lines are one family solution between the backward and forward waves, i.e., between rarefactions and shocks. To solve the Riemann problem (9) we follow Lax [6]. Let us consider the following initial data for system of equations in (9)

$$R(n\Delta\alpha, \beta) = \begin{cases} w_1 & ;\ \beta < J\Delta\beta \\ w_2 & ;\ \beta \geq J\Delta\beta \end{cases} \tag{11}$$

where subscripts 1 and 2 refer to values of w just behind of and just ahead of the discontinuity respectively. If these two values are equal then the solution of (9) is a constant state and its value is equal to the value of initial data. However, if these two values are different then the initial jump discontinuity will propagates in the form of a center expansion wave and/or a shock (i.e., jump discontinuity satisfies the entropy condition.) or a contact discontinuity. In order that solution converges to a unique weak solution of (9), it must satisfies the Rankine-Hugoniot jump condition and the Oleinik entropy condition. At the shock, let us define the values of R(α,ß) just behind of and just ahead of the shock by

$$R_1 = \lim_{\beta\to\beta^-} R(\alpha,\beta) \qquad R_2 = \lim_{\beta\to\beta^+} R(\alpha,\beta)$$

The jump conditions for system of equations (4) are given by

$$\frac{d\beta}{d\alpha} = -\frac{\theta_2-\theta_1}{A_2-A_1}$$

$$\frac{d\beta}{d\alpha} = \frac{\phi(m_2-m_1) - (\theta_2^2-\theta_1^2)}{2(A_2\theta_2 - A_1\theta_1)} \tag{12}$$

$$A_2 m_2^2 = A_1 m_1^2.$$

The entropy condition is given by

$$\frac{F(R_2)-F(R)}{R_2-R} \leq \frac{F(R_2)-F(R_1)}{R_2-R_1}$$

for any R between R_2 and R_1. The entropy satisfaction is a major concern for numerical approximation of solutions of nonlinear fluid flow problems. This simply means that the computed solution converges toward the correct physical solution as the mesh sizes of intervals along α and ß approach to zeros. The above inequality can be written as

$$E(R) = F(R_2) + \frac{d\beta}{d\alpha}(R-R_2)$$

satisfying the following inequality

$$(E(R) - F(R))\ (R_1 - R_2) \geq 0$$

where E(R) defines the chord connecting left and right limiting points across the shock. The entropy related to the first component of F is given by

$$\frac{\theta_2-\theta}{A_2-A} \geq \frac{\theta_2-\theta_1}{A_2-A_1}$$

for any A between A_1 and A_2, and Θ between Θ_1 and Θ_2. Similar inequalities hold for other components of F.

4.1 Rarefaction Waves

Rarefaction waves are two families of solutions curves, forward and backward waves. In this section we compute the simple rarefaction waves of system of equations (9) which can be reformulated in the form

$$U_\alpha + H(U)\ U_\beta = 0$$

where H(U) = H(A, Θ, m) is 3 by 3 matrix given by

$$U=\begin{pmatrix} A \\ \theta \\ m \end{pmatrix} \qquad H=\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & \frac{\phi}{2A} \\ 0 & \frac{m}{2A} & 0 \end{pmatrix}$$

The eigenvalues, μ and thier corresponding eigenvectors, e of the matrix H are

$$\mu = 0,\ \pm\frac{\sqrt{\phi m}}{2A};\quad e=\begin{pmatrix} 1 \\ -\mu \\ -2\mu^2 A/\phi \end{pmatrix}$$

For the system of equations in (9) the simple rarefaction waves are the continuous solutions of (9) of the form

$$(13)\qquad U(\alpha,\beta)=\begin{cases} R_1 & if\ \frac{\beta}{\alpha}<\mu(R_1) \\ v(\frac{\beta}{\alpha}) & if\ \frac{\beta}{\alpha}=\mu(v) \\ R_2 & if\ \frac{\beta}{\alpha}>\mu(R_2) \end{cases}$$

where v is an integral curve of the vector field of the corresponding eigenvector connecting the two constant states such that the corresponding eigenvalue, μ is increasing between this two constant states from left to right. Since the matrix H has three distinct eigenvalues, there are three possible rarefaction waves through any given state. These rarefaction waves are the integral curves of the vector field defined by each eigenvector of matrix H, i.e., each eigenvector is tangent at each point of integral curve. Thus for the eigenvector e corresponding to the eigenvalue μ of matrix H, the integral curves are solutions the following system of equations

$$\frac{dA}{1}=\frac{d\theta}{-\mu}=\frac{dm}{-2\mu^2 A/\phi}$$

If $\mu = 0$ then the integral curves of its corresponding eigenvector, e = (1, 0, 0) are curves where θ and m are both constants. Hence a simple rarefaction wave of the form of (13) exists if the left and right values of θ and m across the shock are equal, in addition, μ must be an increasing function of θ and m from left to right across the shock. Therefore there is no A-rarefaction wave.

θ-rarefaction waves. If μ is not zero then the integral curves of its corresponding eigenvector are curves where m^2A is constant. There are two families of curves where θ is either positive or negative along each integral curve.

5 Numerical Experiments

We compared the numerical solutions using the random choice method developed here with those solutions obtained using the method of characteristics. Consider the initial condition

$$A(0,\beta)=1$$
$$m(0,\beta)=0.01$$
$$\theta(0,\beta)=\begin{cases} \frac{\pi}{12}\beta & -1<\beta<1 \\ 0 & otherwise \end{cases}$$

together with equations (2) and (4). The results obtained with both methods were very close to each other. The numerical calculations show that the convergence of the solution toward the exact solution is independent of choice of the odd number k in (7) as long as k is bounded. In addition, the method is numerically stable. All the striking physical features of system of equations (2) and (4) is observed, i.e., as the initial front propagates into the rest state the center of hump becomes flat and this flat region propagates on both directions untill the front becomes a flat surface.

6 Conclusion

The numerical solutions show that the method is stable and correctly describes the important physical feature of the solution of the model problem. The various choices of random number generators do not have any effect on the accuracy of computed solution as long as the random variable tent toward the equipartitioning of the given interval.

Bibliography

[1] S. K. Godunov, "A finite difference method for the numerical computation of discontinuous solutions of fluid dynamics", Math USSR Sb., 47, 271-290, (1959).

[2] J. Glimm, "Solutions in the Large for Nonlinear Hyperbolic System of Equations", Comm. Pure Appl. Math., 18, 697-715, (1965).

[3] A. J. Chorin, "Random Choice Solutions of Hyperbolic systems", J. Comput. Phys., 22, 517-533, (1976).

[4] G. A. Zakeri, "Geometric Structure of 2D Weak Shock Waves", Appl. Math. & Comp., 33, 161-183, (1989).

[5] G. A. Zakeri, "Numerical Results for Wavefront Tracking", J. Wave-Material Interaction, 3, 127-134, (1988).

[6] P. D. Lax, "Hyperbolic systems of Conservation Laws II", Comm. Pure Appl. Math., 10, 537-566, (1957).

[7] O. A. Oleinik, Amer. Math. Soc. Transl. Ser. 2, 33, 285-290 (1963).

[8] G. B. Whitham, Linear and Nonlinear Waves, John Wiley & Sons, (1974).

An Algorithm to Estimate Parameters for a Stochastic Linear Compartmental System

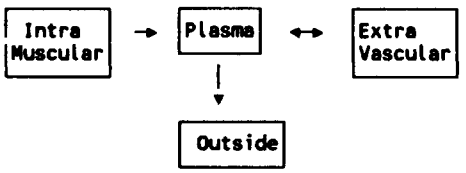
P. M. Simpson
Biostatistics, MCV/VCU
Richmond, VA 23298-0032

D. M. Allen
Statistics, POT, UK
Lexington, KY 40506

ABSTRACT. Linear compartmental systems have compartments with flows to and from the compartments. Of interest, is the estimation of the constants, θ, governing the flows. In the particular system considered, only one compartment, out of several, is observed for n cases over k time points. A stochastic model is used with a maximum likelihood approach taken to the estimation of θ. The algorithm involves iteratively using an estimate of θ to solve differential equations which describe the system, and improve on the estimate of θ by adding a constant multiple, α, of an increment $\delta\theta$. Allen's results are incorporated to obtain required derivatives. Due to non-zero correlations, a modification to Jennrich and Moore's results is made, involving using both the observations and their cross-products, to obtain $\delta\theta$. α is determined with Fletcher's method. A program in Turbo Pascal implements the algorithm.

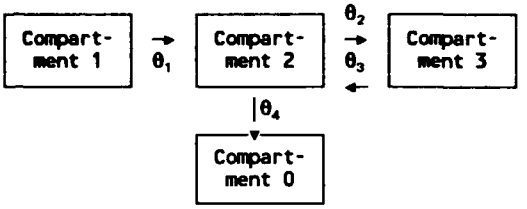
INTRODUCTION. Compartmental systems have long been a useful tool in pharmacokinetics (Wagner (1971)). The body is thought of as a series of compartments, with a drug moving between any of the compartments. For example, Gladtke et al. (1979) p. 36, suppose that the body may be represented as three compartments - plasma, muscle and extravascular. An initial muscle injection is given and the levels of the drug in,the plasma are monitored from time to time. The drugs will flow from muscle to plasma. Additionally, flow will be between the extravascular system and plasma, and from plasma to the outside, as depicted in diagram 1.

Diagram 1:



For simplicity, this paper will be restricted to a three compartmental system of this type, with compartments 1,2,3 and the outside, as compartment 0; flows have a parameter attached to them as in diagram 2, (In the traditional deterministic model, these are the rate parameters).

Diagram 2 :



It is assumed that a bolus injection is given in compartment 1 and only compartment 2 is observed, at times $t_1,t_2,\ldots,t_k$. Let θ_0 be the initial concentration in compartment 1 and C(t) be the concentration at time t.Then

$$\boldsymbol{C}(t)=\begin{bmatrix}C_1(t)\\C_2(t)\\C_3(t)\end{bmatrix},\quad \boldsymbol{C}(0)=\begin{bmatrix}\theta_0\\0\\0\end{bmatrix}.$$

STOCHASTIC MODEL . For the "particle model", as discussed by Purdue (1974a), it is assumed that there are N particles in the system acting independently. Transitions between compartments follow a Markov process, with the transition probability being constant. The resulting system of equations is as follows:

$$dP(t)/dt = P(t)A^T, \qquad .....(1)$$

where $P(t) = (p_{ij}(t))$ is a nonsingular 3x3 matrix with $p_{ij}(t)$, $i,j=1,2,3$, the probability of a particle transferring from compartment "i" to compartment "j" in time t and

$$\boldsymbol{A}=\begin{bmatrix}-\theta_1 & 0 & 0\\ \theta_1 & -(\theta_2+\theta_4) & \theta_3\\ 0 & \theta_2 & -\theta_3\end{bmatrix}=(a_{ij}).$$

$\theta = (\theta_0\ \theta_1\ \theta_2\ \theta_3\ \theta_4)^T$ is to be estimated.

Under these assumptions it can be shown that the exact distribution of C(t), given C(0) is a multinomial (since only $C_1(0)$ is nonzero), but the distribution of $C(t_1)$, given $C(t_2)$, is a convolution of multinomials. Thus, to write down the exact distribution of the concentrations in compartment 2 for $t_1,t_2,\ldots,t_k$, when k is large is impossible since it would consist of many sums whose limits are complex. Using a diffusion approximation, it has been demonstrated that the concentrations over time have a multivariate normal distribution, Lehoczky and Gaver (1977). Hence, the distribution of the concentrations in compartment 2 have a (marginal) multivariate distribution, with mean and variance-covariance matrix the same as that given by the particle model . Simpson (1988) has shown that an estimator derived from the maximum likelihood equations, using this approximate normal distribution, is still a consistent asymptotically normal estimator, if the particle model is the correct one. Thus, an algorithm was written to estimate θ using the approximate normal distribution.

Suppose that X_{ij} is the concentration in compartment 2 for person i, i=1,2,...n at time t_j, j=1,2,...k. $X_i = (X_{i1},X_{i2},\ldots X_{ik})$ are the k observations of ith case. $X_1,X_2,\ldots, X_n$ are independently and identically distributed as a normal variable with mean and variance which are functions of the probability matrix P. By considering a vector comprised of the sufficient statistics of the covariance of X, it can be seen that a normally distributed random variable is a linear exponential and so the algorithm of Jennrich and Moore (1975) ,hence referred to as JMA, can be used to find the approximate maximum likelihood estimator . We take the k_1x1 vector Y, where $k_1 = k(k+3)/2$ and

$$\begin{aligned}\boldsymbol{Y}=\frac{1}{n}\Big[&\sum_{r=1}^{n}X_{r1},\sum_{r=1}^{n}X_{r2},\ldots\ldots,\sum_{r=1}^{n}X_{rk},\\&\sum_{r=1}^{n}X_{r1}^2,\sum_{r=1}^{n}X_{r1}X_{r2},\ldots\sum_{r=1}^{n}X_{r1}X_{rk},\\&\sum_{r=1}^{n}X_{r2}^2,\ldots\ldots,\sum_{r=1}^{n}X_{r2}X_{rk},\\&\ldots\ldots\ldots\ldots,\\&\sum_{r=1}^{n}X_{rk-1}^2,\sum_{r=1}^{n}X_{rk-1}X_{rk},\\&\sum_{r=1}^{n}X^2{}_{rk}\Big]\end{aligned}$$

Using properties of normality, Y has a mean, $\mu(\theta)$, and variance-covariance matrix, $\Sigma(\theta)$, which is a function of the mean and variance-covariance matrix of X, and therefore of P, also.

THE ALGORITHM. Suppose that the k_1x5 matrix of partial derivatives with respect to θ_i, i=0,1,..,4 is denoted by $d\mu/d\theta$. According to the JMA, for a given θ, replace θ by $\theta+\alpha\delta\theta$,where

$$\delta\theta=\left[\left(\frac{d\mu}{d\theta}\right)^T\Sigma^{-1}\frac{d\mu}{d\theta}\right]^{-1}\left(\frac{d\mu}{d\theta}\right)^T\Sigma^{-1}(y-\mu)$$

This requires that
(1) P(.) and its derivatives with respect to θ be obtained
(2) $\delta\theta$ be calculated.
(3) An appropriate α be found.

(1) P(.) and its derivatives:
For this compartmental system, P(.) and its derivatives can be calculated analytically. However, to maintain generality for the program, it was decided to obtain P(.) and its derivatives by applying the method developed by Allen (1987) to columns of P(.).

The steps, for each column of P, are as follows:
(i) Find R so that
- $\mathbf{R^TAR = T}$, a triangular matrix,
- $\mathbf{R^TR = I}$, the identity matrix

(ii) Using R, obtain a triangular system of equations; so, a backward solving technique can be used, with the initial condition P(0) = I satisfied.

For any θ_a, a=1,2,3,4, the equation (1) becomes

$$\frac{d\boldsymbol{V}_a}{dt}(t)=\boldsymbol{T}\boldsymbol{V}_a(t)+\boldsymbol{R}^T\frac{\partial\boldsymbol{A}}{\partial\theta_a}\boldsymbol{R}\;\boldsymbol{R}^T\boldsymbol{X}(t),$$
$$\boldsymbol{R}\boldsymbol{V}_a(t)=\frac{\partial\boldsymbol{X}(t)}{\partial\theta_a}$$

and Allen's method can be used again to solve for $V_a(t)$, with $V_a(0) = 0$, as the initial condition

(2) Calculation of $\delta\theta$:
Using Wilkinson's algorithms, Wilkinson (1971), a Cholesky decomposition is used to find $\Sigma^{1/2}$, where $\Sigma^{1/2}(\Sigma^{1/2})^T=\Sigma$. Using this, the following equation is solved to obtain $\delta\Theta$:

$$\mu_1=\Sigma^{1/2}\frac{d\mu}{d\theta}$$
$$y_1=\Sigma^{1/2}(y-\mu)$$
$$\mu_1^T\mu_1\delta\theta=\mu_1^Ty_1$$

(3) An α:
Fletcher's descent method, Fletcher (p.26, 1980) is used to find α. It uses the first derivatives only. Assuming uniform continuity conditions, it will achieve, at least, a local optimum if an optimum exists and if the starting value is close enough.

SUMMARY. A stochastic approach rather than the traditional deterministic model approach is taken to a particular compartmental model, where only one compartment is observed. An algorithm is developed which uses the JMA to obtain maximum likelihood estimates from a diffusion approximation. The program is written in Turbo Pascal and can be generalised:
(i) to other linear compartmental systems,
(ii) for θ to be functions of time,
Work is being done:
(iii) to incorporate measurement error in the model
(iv) to include people variation

An important step for its general use would be to incorporate this program in a general pharmacokinetic program so that, in a user friendly environment, its estimates could be easily compared to those obtained from other methods.

BIBLIOGRAPHY.

Allen, D.M. (1987). Computation for Compartmental Models. ASA Computer Science and Statistics: Proceedings of the 19th Symposium on the Interface. editor R.Hieburger.

Fletcher, R. (1980). Practiacl Methods of Optimization Vol.I: Unconstrained Optimization. Wiley, New York.

Gladtke, E., and von Hattingberg, H.M. (1979). Pharmacokinetics. Springer-Verlag, New York.

Jennrich, R.I., and Moore, R.H. (1975). Maximum likelihood Estimation by means of Nonlinear Least Squares. ASA Proceedings of the Statistical Computing Section.

Purdue, P. (1979). Stochastic Compartmental Models: A review of the Mathematical Theory with Ecological Applications. Compartmental Analysis of Ecosystem Models, edited by Matis, J.H., Patten, B.C., and White, G.C., International Cooperative Publishing House, Fairland, Maryland.

Simpson, P. (1988). On Stochastic Models for Linear Compartmental Systems. Unpublished dissertation, University of Kentucky, Lexington, Kentucky.

Wagner, J.C.(1971) Biopharmaceutics and Relevant Pharmacokinetics. Hamilton, Illinois.

Wilkinson, J.H. and Reinsch, C. (1971) Linear Algebra. Springer-Verlag, Heidelberg.

Sampling Based Approach to Computing Nonparametric Bayesian Estimators with Doubly Censored Data

Lynn Kuo *

Department of Statistics
University of Connecticut
Storrs, CT 06269

and

Department of Operations Research
Naval Postgraduate School
Monterey, CA 93943.

Abstract

Nonparametric Bayesian estimators with Dirichlet process priors for doubly censored data can be derived from mixtures of Dirichlet distributions. To circumvent the computational difficulties in evaluating these mixtures, this paper describes the Gibbs sampling approach to approximating them. The Gibbs samplers augment the censored data by the number of observations falling into each interval. An example taken from Turnbull (1974) is given to illustrate the approach.

Keywords: Gibbs sampling; Stochastic substitution; Dirichlet process priors; Doubly censored data.

1 Introduction

Nonparametric Bayesian inference for the survival function with right censored data has been studied by Susarla and Van Ryzin (1976), and Ferguson and Phadia (1979). However, we often encounter the situation where some observations are censored from the left and some observations are censored from the right. Turnbull (1974) has cited many papers addressing doubly censored data sets and their frequentists' analyses.

This paper studies a nonparametric Bayesian approach to the data analysis. This approach allows us to incorporate prior belief and frees us from making a restrictive model assumption for the survival function. Specifically, we assume the survival function is taken from a Ferguson's (1973) Dirichlet process, $\mathcal{D}(\alpha)$. The prior parameter, α, may be written $\alpha = MF_0$, where F_0 represents the statistician's prior guess of the distribution function of the times of incident(death) and M represents the degree of concentration of the true distribution function around F_0.

Due to the doubly censored data, it is usually very difficult to obtain an explicit expression for the nonparametric Bayes estimators. Fortunately, it is not necessary to have this closed form to obtain numerical solutions to the problem of computing Bayes estimators. This paper proposes a Gibbs sampling approach to computing them. The approach augments the data by using latent variables that decompose the number of the censored observations into the possible number of observations falling into each interval. This augmentation facilitates us in specifying the conditional densities of the survival functions given the latent variables. A repeated sampling scheme, that uses this conditional density and the conditional density of the latent vari-

*Research supported by NSF grant DMS-90-08021 and Naval Postgraduate School

ables given the distribution function and the data, allows us to approximate the posterior distribution of the survival function.

Although we emphasize the doubly censored data in this paper. The model discussed in the next section is very general. It applies to the data set that includes only completely observed data and right censored data. Nonparametric Bayesian estimators in this situation have been derived by Susarla and Van Ryzin (1976), and Ferguson and Phadia (1979). Kuo (1991) computed these estimates based on the data from Kaplan and Meier (1958) using the Gibbs sampling approach. These estimates compare favorably to the estimates obtained by Susarla and Van Ryzin.

The model also includes the situation that none of the completely observed data are available, i.e. all incidents are either right or left censored. The likelihood reduces to that considered in the quantal bioassay. Gelfand and Kuo (1991) studies the sampling based approach to this problem. In addition to the Dirichlet process prior, they also consider a product of beta prior. They also generalize their results to polytomous response.

Section 2 discusses the model. Section 3 describes the Gibbs sampling approach. An example using the data set in Turnbull (1974) is given in Section 4.

2 The Model

The model is basically the one studied by Turnbull (1974). Turnbull proposes a self-consistent algorithm for computing the generalized maximum likelihood estimators. This paper adds the Dirichlet process prior to the model.

Let $T_1, T_2, \cdots, T_n$ denote the true survival times of n individuals that could be observed precisely if no censoring were present. The T_i are independent and identically distributed with distribution F; that is, $F(t) = Prob(T \leq t)$ for $t \geq 0$. We consider the case that not all T_i are observed precisely. For each i, we assume that there are "windows" of observations L_i and U_i ($L_i \leq U_i$) that are either fixed constants or random variables independent of the $\{T_i\}$. We observe

$$X_i = max\ [\ min(T_i, U_i),\ L_i].$$

Moreover, for each item, we also know whether it is left censored (late entry) with $X_i = L_i$, or right censored (a loss) with $X_i = U_i$, or a precisely observed time (death) with $X_i = T_i$.

Usually, items are examined at discrete times, for example, monthly. We can assume there is a natural discrete time scale $0 < t_1 < t_2 < \cdots < t_m$, and the observed deaths are classified into one of the m intervals $(0, t_1], (t_1, t_2], \cdots, (t_{m-1}, t_m]$. Let δ_i denote the number of observed deaths in the period $(t_{i-1}, t_i]$, μ_i denote the number of late entries at age t_i, and λ_i denote the number of losses at t_i. It is assumed that the late entries μ_i all occur at the end of age period $(t_{i-1}, t_i]$ and the losses λ_i all occur at the beginning of (t_i, t_{i+1}). The data can be summarized by the following tabulation:

Type\age	$(0, t_1]$	$(t_1, t_2]$	$\cdots$	$(t_{m-1}, t_m]$
Deaths	δ_1	δ_2	$\cdots$	δ_m
Late entries ($\leq$)	μ_1	μ_2	$\cdots$	μ_m
Losses ($>$)	λ_1	λ_2	$\cdots$	λ_m

Let $P_j = P(t_j) = 1 - F(t_j)$ denote the survival function evaluated at t_j. The likelihood function is proportional to

$$\prod_{j=1}^{m} (P_{j-1} - P_j)^{\delta_j} (1 - P_j)^{\mu_j} P_j^{\lambda_j}.$$

Let $q_j = P_{j-1} - P_j$ for $j = 1, \cdots, m$ and let $q_{m+1} = P_m$. The Ferguson's process prior assumes that the distribution of the q's is the Dirichlet distribution

$$\pi(\vec{q}) = C \prod_{j=1}^{m+1} (q_j)^{\alpha_j - 1},$$

where

$$\alpha_j = M(F_0(t_j) - F_0(t_{j-1})),$$

for $j = 1, \cdots, m+1$, with $F_0(t_{m+1}) = 1$, and

$$C = \frac{\Gamma(M)}{\prod_{j=1}^{m+1} \Gamma(\alpha_j)}.$$

The posterior distribution of the $\vec{q} = (q_1, q_2, \cdots, q_m; q_{m+1})$ is a mixture of Dirichlet distributions. The results of Antoniak (1974) can be used to derive this mixture. The next section will develop the Gibbs sampling technique to approximating this mixture.

3 Gibbs Sampling

To employ the Gibbs sampling technique, we need to introduce the latent variables that decompose the

numbers of losses and late entries into the numbers of observations belonging to individual intervals. Let $Z_{1j}, Z_{2j}, \cdots, Z_{jj}$ denote the random variables that count the number of observations in μ_j that might fall in the intervals $(0, t_1], (t_1, t_2], \cdots, (t_{j-1}, t_j]$ respectively. Observe $\mu_j = \sum_{l=1}^{j} Z_{lj}$. Moreover, let $Z_{j+1,j}, \cdots, Z_{m+1,j}$ denote the number of observations in λ_j that might fall in the intervals $(t_j, t_{j+1}], \cdots, (t_{m-1}, t_m], (t_m, \infty]$ respectively. Observe $\lambda_j = \sum_{l=j+1}^{m+1} Z_{lj}$.

Our objective is to obtain the posterior distribution of the $\vec{q}$ given the data. To apply the stochastic augmentation idea discussed in Tanner and Wong (1987) and in Gelfand and Smith (1990), we can sample from two densities recursively. The first density is the posterior density of the $\vec{q}$ given the $\vec{Z}$s and the data, which is an updated Dirichlet distribution depending only on the $\vec{Z}$s. The second one is the postrior density of the $\vec{Z}$s given the $\vec{q}$ and the data, which is the density of a product of multinomial distributions.

Suppose at the ith iteration step of the Gibbs sampling, we have the probabilities $\vec{q}^{\,i} = (q_1^i, q_2^i, \cdots, q_{m+1}^i)$,with $\sum_{l=1}^{m+1} q_l^i = 1$, where q_l^i represents an estimate of q_l. Then we can update the Z variables from the multinomial distributions. That is, for each j, $j = 1, \cdots, m$, we sample $Z_{1j}^{i+1}, \cdots, Z_{jj}^{i+1}$ from the multinomial distribution with sample size μ_j and parameters $r_{1j}^i, \cdots, r_{jj}^i$, where $r_{lj}^i = q_l^i / \sum_{l=1}^{j} q_l^i$ for $l = 1, \cdots, j$. Similarly, we sample $Z_{j+1,j}^{i+1}, \cdots, Z_{m+1,j}^{i+1}$ from the multinomial distribution with sample size λ_j and parameters $r_{j+1,j}^i, \cdots, r_{m+1,j}^i$, where $r_{lj}^i = q_l^i / \sum_{l=j+1}^{m+1} q_l^i$ for $l = j+1, \cdots, m+1$.

Having sampled the Z random variables, we can update the q variables by the Dirichlet distribution. Let us compute, for each $l, l = 1, \cdots, m+1$,

$$Y_l^{i+1} = \alpha_l + \delta_l + \sum_{j=1}^{m} Z_{lj}^{i+1}.$$

Then we could sample $(q_1^{i+1}, \cdots, q_m^{i+1}; q_{m+1}^{i+1})$ from the Dirichlet distribution with parameters $(Y_1^{i+1}, \cdots, Y_{m+1}^{i+1})$. Now we use the updated q's to continue sampling until the Ith step.

By starting independent initial choices of the $\vec{q}$s, we can also replicate the iterations ν times. After ν replications each to the Ith iteration, we have $q_{1s}^I, q_{2s}^I, \cdots, q_{m+1,s}^I$ and $Y_{1s}^I, \cdots, Y_{m+1,s}^I$, for $s = 1, \cdots, \nu$. The posterior distribution of q_l for $l = 1, \cdots, m+1$ can be approximated by

$$\hat{F}(q_l | data) = \nu^{-1} \sum_{s=1}^{\nu} Beta(Y_{ls}^I, \sum_{k \neq l}^{m+1} Y_{ks}^I),$$

where $Beta(\alpha, \beta)$ denotes the beta density with parameters α and β. Then the posterior estimate of the q_l can be given by

$$\hat{q}_l = \nu^{-1} \sum_{s=1}^{\nu} \frac{Y_{ls}^I}{\sum_{l=1}^{m+1} Y_{ls}^I}.$$

The posterior standard error ($S.E.$) of $\hat{q}_l$ and the naive posterior confidence interval for the q_l can be computed similarly from the replicated samples.

The numbers I and ν are selected to achieve convergence to smooth estimates. We can fix a number of ν, plot the posterior densities of q_l given the other q's (beta distributions) for two different iteration numbers, for example, 5 units apart. We increase the iteration numbers until the two densities come close to each other. Then we increase the number of replications for the final run. Choice of I determines the convergence of the density estimates to the actural marginal posterior density at an exponential rate (Geman and Geman, 1984; Tanner and Wong, 1987). The order of convergence for the replications is $O(\nu^{-1})$. The standard error of the mean and the confidence intervals from the replications could also help us in selecting the desired ν.

4 Numerical Examples

The data set taken from Turnbull (1974) is summarized in the following:

Type of obs.\ age	$(0, t_1]$	$(t_1, t_2]$	$(t_2, t_3]$	$(t_3, t_4]$
Deaths	12	6	2	3
Late entries	2	4	2	5
Losses	3	2	0	3

The likelihood function is

$$\begin{aligned} L(\vec{q}) \;=\; & q_1^{12} q_1^2 (q_2 + q_3 + q_4 + q_5)^3 \times \\ & q_2^6 (q_1 + q_2)^4 (q_3 + q_4 + q_5)^2 \times \\ & q_3^2 (q_1 + q_2 + q_3)^2 \times \\ & q_4^3 (q_1 + q_2 + q_3 + q_4)^5 q_5^3. \end{aligned}$$

Table 1: Gibbs Approximation to the Bayes Estimates for $\alpha(l) = .00001, l = 1, \cdots, 5$

Statistics \ Cell	1	2	3	4	5
$\hat{q}_l$	.462	.243	.084	.116	.095
S.E.	.001	.002	.001	.001	.001
$\hat{P}_l$	.538	.295	.211	.095	0

Table 2: Gibbs Approximation to the Bayes Estimates for $\alpha(l) = 1, l = 1, \cdots, 5$

Statistics \ Cell	1	2	3	4	5
$\hat{q}_l$	.431	.237	.100	.126	.106
S.E.	.004	.005	.003	.003	.002
$\hat{P}_l$	.569	.332	.232	.106	0

We generate all the Z_{lj} variables as described in Section 3. For example, let $Z_{11} = 2$ and $Z_{21} + Z_{31} + Z_{41} + Z_{51} = 3$. We generate the $Z_{21}, \cdots, Z_{51}$ variables from the multinomial $MN(3, r_{21}, r_{31}, r_{41}; r_{51})$ distribution, etc. If there is only one cell in the multinomial distribution, we just let the corresponding Z variable be the frequency count of that cell. If the number of count is zero for a group of cells, then all its summands are set to zero.

Tables 1-3 exhibit $\hat{q}_l, l = 1, \cdots, 5$, the Bayes estimates approximated by the Gibbs samples with $I = 10$ and $\nu = 1000$. The estimated posterior standard errors (S.E.) constructed from the replicated samples are also given. The naive posterior 95% coverage intervals for q_l can be obtained using $\hat{q}_l \pm 1.96 S.E.$ The last row summarizes the estimates of the q_l in terms of the P_l. These values can be compared with the generalized maximum likelihood estimates computed by Turnbull, which are .538, .295, .210, .095 and 0. Three sets of prior parameters are chosen: (1) $\alpha(l) = .00001$; (2) $\alpha(l) = 1$; and (3) $\alpha(l) = 5$ for $l = 1, \cdots, 5$. The first set mimics a very small prior sample size. The second set selects a uniform prior on the $\{q_l\}$. The last one illustrates strong prior influence which assigns uniform weight 5 over each of the cells. The results confirm our expectations that the Bayes estimates from the case (1) are very close to the estimates produced by Turnbull.

Table 3: Gibbs Approximation to the Bayes Estimates for $\alpha(l) = 5, l = 1, \cdots, 5$

Statistics \ Cell	1	2	3	4	5
$\hat{q}_l$	.354	.226	.137	.148	.135
S.E.	.001	.001	.001	.001	.000
$\hat{P}_l$	.644	.420	.283	.134	0

References

[1] Antoniak, C.E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist.*, **2**, 1152-74.

[2] Ferguson, T.S. (1973). A Bayesian analysis of some nonparametric problems. *Ann. Statist.*, **1**, 209-30.

[3] Ferguson, T.S. and Phadia, E.G. (1979). Bayesian nonparametric estimation based on censored data. *Ann. Statist.*, **7**, 163-86.

[4] Gelfand, A.E. and Kuo, L. (1991). Nonparametric Bayesian bioassay including ordered polytomous response. *Biometrika*, **78**, in press.

[5] Gelfand, A.E. and Smith, A.F.M. (1990). Sampling-based approaches to calculating marginal densities. *J. Am. Statist. Assoc.*, **85**, 398-409.

[6] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis abd Machine Intelligence*, **6**, 721-741.

[7] Kaplan, E.L. and Meier, P. (1958) Nonparametric estimation from incomplete observations. *J. Am. Statist. Assoc.*, **53**, 457-81.

[8] Kuo, L. (1919). Sampling based approach to computing nonparametric Bayesian survival curves with censored data. Preprint.

[9] Susarla, V. and Van Ryzin, J. (1976). Nonparametric estimation of survival curves from incomplete observations. *J. Am. Statist. Assoc.*, **71**, 897-902.

[10] Tanner, M. and Wong, W. (1987) The calculation of posterior distributions by data augmentation (with discussion). *J. Am. Statist. Assoc.*, **81**, 82-86.

[11] Turnbull, B.W. (1974) Nonparametric estimation of a survivorship function with doubly censored data. *J. Am. Statist. Assoc.*, **69** 169-73.

CALCULATING MAXIMUM LIKELIHOOD ESTIMATORS FOR THE GENERALIZED PARETO DISTRIBUTION

Scott D Grimshaw
College of Business and Management
University of Maryland
College Park, MD 20742

Abstract

The Generalized Pareto Distribution (GPD) is a two-parameter family of distributions which can be used to model exceedences over a threshold. Maximum likelihood parameter estimates are preferred since they are asymptotically normal and asymptotically efficient. Numerical methods are required for maximizing the log-likelihood since the minimal sufficient statistics are the order statistics and there is no obvious simplification of the nonlinear likelihood equation. An algorithm is given to compute GPD maximum likelihood estimates by reducing the two-dimensional numerical search for the zeros of the gradient vector to a one-dimensional numerical search.

1. Generalized Pareto Distribution

A random variable X is defined to have a *Generalized Pareto Distribution* (GPD), with parameters k and a such that $-\infty < k < \infty$, $a > 0$, if the cumulative distribution function is given by

$$F_{\mathrm{GPD}}(x;k,a) = \begin{cases} 1-\left(1-\dfrac{kx}{a}\right)^{1/k}, & k<0,\ x>0 \\ 1-e^{-x/a}, & k=0,\ x>0 \\ 1-\left(1-\dfrac{kx}{a}\right)^{1/k}, & k>0, \\ & 0<x<a/k. \end{cases}$$

The density function is given by

$$f_{\mathrm{GPD}}(x;k,a) = \begin{cases} \dfrac{1}{a}\left(1-\dfrac{kx}{a}\right)^{(1/k)-1}, & k<0,\ x>0 \\ \dfrac{1}{a}e^{-x/a}, & k=0,\ x>0 \\ \dfrac{1}{a}\left(1-\dfrac{kx}{a}\right)^{(1/k)-1}, & k>0, \\ & 0<x<a/k, \end{cases}$$

and the quantile function, $Q(u) = F^{-1}(u)$, is given by

$$Q_{\mathrm{GPD}}(u;k,a) = -a \cdot g(1-u;k).$$

where $g(\cdot)$ is the *power transformation* (also called the Box-Cox transformation), defined for $z > 0$ by

$$g(z;\lambda) = \begin{cases} \dfrac{z^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \ln z, & \lambda = 0. \end{cases}$$

Pickands (1975) introduced the GPD as a two-parameter family of distributions for exceedences over a threshold. The parameters of the GPD are a, the scale parameter, and k, the shape parameter. Three special cases of the GPD are:

(i) if $k = 1$ the distribution is Uniform (0,a);
(ii) if $k = 0$ the distribution is Exponential ($1/a$);
(iii) if $k < 0$ the distribution is Pareto.

Maximum likelihood estimation of the parameters (k,a) has been considered by DuMouchel (1983), Davison (1984), R. L. Smith (1984,1987), J. A. Smith (1986), and Joe (1987). R. L. Smith (1984) showed that under certain conditions for regularity the maximum likelihood estimates are asymptotically normal and asymptotically

efficient. If $(\hat{k}_n, \hat{a}_n)$ denote the maximum likelihood estimates, then for $k < \frac{1}{2}$, as $n \to \infty$

$$\begin{bmatrix} \hat{k}_n \\ \hat{a}_n \end{bmatrix} \text{ is } \mathrm{AN}\left(\begin{bmatrix} k \\ a \end{bmatrix}, n^{-1} \begin{bmatrix} (1-k)^2 & a(1-k) \\ a(1-k) & 2a^2(1-k) \end{bmatrix} \right).$$

The maximum likelihood estimates must be derived numerically since the minimal sufficient statistics for the GPD are the order statistics and there is no obvious simplification of the nonlinear likelihood equation.

Hosking and Wallis (1987) proposed a modified Newton–Raphson algorithm to find the maximum of the log-likelihood. They also propose method of moments and method of probability-weighted moments as alternative parameter estimators for the GPD when a reduction of the parameter space to $-\frac{1}{2} < \rho < \frac{1}{2}$ is reasonable. These alternative estimators are inefficient, but are easier to compute than the maximum likelihood estimates.

In this paper an algorithm for computing the maximum likelihood estimates is presented. The two-dimensional numerical search for the zeros of the gradient of the GPD log-likelihood is reduced to a one-dimensional numerical search. This simplification is due to a reparameterization pointed out by Davison (1984).

2. Computing Maximum Likelihood Parameter Estimates

Suppose $\mathbf{X} = \{X_1, \ldots, X_n\}$ is a random sample from the GPD with largest value $X(n;n)$. The log-likelihood is given by

$$\mathcal{L}_{\mathrm{GPD}}(k, a; \mathbf{X}) = \begin{cases} -n \ln a + \left(\frac{1}{k} - 1\right) \sum_{i=1}^{n} \ln\left(1 - \frac{kX_i}{a}\right), & k < 0,\ a > 0 \\ -n \ln a - \frac{1}{a} \sum_{i=1}^{n} X_i, & k = 0,\ a > 0 \\ -n \ln a + \left(\frac{1}{k} - 1\right) \sum_{i=1}^{n} \ln\left(1 - \frac{kX_i}{a}\right), & k > 0, \\ & a > kX(n;n). \end{cases}$$

If $k > 1$, there is no maximum likelihood estimate since for any $k > 1$,

$$\lim_{a/k \to X(n;n)^+} \mathcal{L}_{\mathrm{GPD}}(k, a; \mathbf{X}) = \infty.$$

In order to obtain a finite maximum of the GPD log-likelihood, the constraint $k \leq 1$ must be imposed. Therefore, computing the GPD maximum likelihood estimators is an optimization on the constrained space

$$\mathcal{A} = \{k < 0, a < 0\} \cup \{0 < k < 1, a/k > X(n;n)\}.$$

There are two values of (k, a) which must be investigated to compute the GPD maximum likelihood estimator. The first is the local maximum of the log-likelihood on the space $\mathcal{A}$. The second is at the boundary of $\mathcal{A}$ where $k = 1$.

2.1. *Local Maximum on $\mathcal{A}$.* To compute the local maxima on the space $\mathcal{A}$, consider the gradient vector of the GPD log-likelihood given in the Appendix. The solution to the simultaneous equations may be simplified and written as

$$\left.\begin{aligned} \frac{\partial \mathcal{L}_{\mathrm{GPD}}(k, a; \mathbf{X})}{\partial k} &= 0 \\ \frac{\partial \mathcal{L}_{\mathrm{GPD}}(k, a; \mathbf{X})}{\partial a} &= 0 \end{aligned}\right\}$$

$$\Rightarrow \begin{cases} n(\hat{k} - 1) = \sum_{i=1}^{n} \ln\left(1 - \frac{\hat{k}X_i}{\hat{a}}\right) \\ \qquad + (\hat{k} - 1) \sum_{i=1}^{n} \left(1 - \frac{\hat{k}X_i}{\hat{a}}\right)^{-1} \\ n = (\hat{k} - 1) \sum_{i=1}^{n} \left(1 - \frac{\hat{k}X_i}{\hat{a}}\right)^{-1} \end{cases}$$

$$\Rightarrow \begin{cases} \left[1 + (1/n) \sum_{i=1}^{n} \ln\left(1 - \frac{\hat{k}X_i}{\hat{a}}\right)\right] \\ \cdot \left[(1/n) \sum_{i=1}^{n} \left(1 + \frac{\hat{k}X_i}{\hat{a}}\right)^{-1}\right] = 1 \\ \hat{k} = -1/n \sum_{i=1}^{n} \ln\left(1 - \frac{\hat{k}X_i}{\hat{a}}\right). \end{cases}$$

The bivariate search for the zeroes of the gradient vector over $\mathcal{A}$ can be reduced to a univariate search since the second equation is a closed form representation for the estimator of k given the ratio $\hat{k}/\hat{a}$, and the first equation depends only on $\hat{k}/\hat{a}$. Therefore, local maxima of the log-likelihood of the GPD correspond to zeros of the function

$$h(\theta) = \left[1 + (1/n) \sum_{i=1}^{n} \ln(1 - \theta X_i)\right] \cdot \left[(1/n) \sum_{i=1}^{n} (1 - \theta X_i)^{-1}\right] - 1, \tag{2.1}$$

with domain

$$\mathcal{B} = \{\theta < 1/X(n;n)\}. \tag{2.2}$$

However, it is important to recognize that not every zero of $h(\theta)$ corresponds to a zero of the gradient vector of the GPD log-likelihood. Therefore, while reducing the bivariate search to a univariate search provides the benefit of simplified computation, it comes with the complication of extraneous zeros.

For example, notice that $h(0) = 0$. Clearly then $\theta = 0$ is a zero of $h(\theta)$. However, $\theta = 0$ corresponds to $k = 0$ in the log-likelihood equations and it can be shown that the gradient vector at $k = 0$ has elements

$$\lim_{k\to 0} \frac{\partial \mathcal{L}_{\mathrm{GPD}}(k,a;\mathbf{X})}{\partial k} = \sum_{i=1}^{n} \frac{X_i^2}{2a^2} - \sum_{i=1}^{n} \frac{X_i}{a}$$

$$\lim_{k\to 0} \frac{\partial \mathcal{L}_{\mathrm{GPD}}(k,a;\mathbf{X})}{\partial a} = \frac{1}{a}\left(\sum_{i=1}^{n} \frac{X_i}{a} - n\right)$$

which are equal to zero if and only if $(1/n)\sum_{i=1}^{n} X_i^2 = 2\bar{X}^2$. Therefore, the zero $\theta = 0$ does not correspond to a local maxima of $\mathcal{L}_{\mathrm{GPD}}(k,a;\mathbf{X})$.

The presence of these extraneous zeros causes two significant complications. First, an algorithm must search the space $\mathcal{B}$ for more than one zero. In fact, since it is known $\theta = 0$ does not correspond to a local maxima of $\mathcal{L}_{\mathrm{GPD}}(k,a;\mathbf{X})$, the algorithm must be designed to avoid numerical convergence to $\theta = 0$. Second, every zero of $h(\theta)$ must be evaluated to determine if it corresponds to a local maxima, local minima, or saddle point of $\mathcal{L}_{\mathrm{GPD}}(k,a;\mathbf{X})$, or an extraneous zero of $h(\theta)$.

The following theorem states several properties of $h(\theta)$ which are useful in formulating an algorithm for determining zeros of $h(\theta)$.

Theorem: Consider the function $h(\theta)$ given in (2.1) defined on the space $\mathcal{B}$ given in (2.2). Then:

$$\lim_{\theta\to 1/X(n;n)} h(\theta) = -\infty$$

$$h(\theta) < 0 \text{ for all } \theta < \theta_L = \frac{2[X(1;n) - \bar{X}]}{[X(1;n)]^2}$$

$$h'(\theta) = \frac{1}{\theta}\left\{(1/n)\sum_{i=1}^{n}(1-\theta X_i)^{-2} - \left[(1/n)\sum_{i=1}^{n}(1-\theta X_i)^{-1}\right]^2 - \left[(1/n)\sum_{i=1}^{n}\ln(1-\theta X_i)\right] \cdot \left[(1/n)\sum_{i=1}^{n}(1-\theta X_i)^{-1} - (1/n)\sum_{i=1}^{n}(1-\theta X_i)^{-2}\right]\right\}$$

$$h'(0) = 0$$

$$h''(0) = (1/n)\sum_{i=1}^{n} X_i^2 - 2\bar{X}^2$$

The first result indicates an upper bound for any zero of $h(\theta)$ is given by $\theta_U = 1/X(n;n)$. Since this is a limiting result, an algorithm can use $\theta_U - \epsilon$ for some $\epsilon > 0$ as the upper bound. The second result, provided by an anonymous referee, provides a lower bound, θ_L, for any zero of $h(\theta)$. Coupling these two results with the fact that $\theta = 0$ is an extraneous zero, an algorithm must divide the space $\mathcal{B}$ into $(\theta_L, 0)$ and $(0, \theta_u)$ and numerically search for zeros of $h(\theta)$ on these two bounded intervals. Because bounds are known, modifications of the Newton–Raphson zero search algorithms can be made which limit step size so that iterative solutions remain within the known boundaries.

The third result is the derivative of $h(\theta)$, required for the Newton–Raphson algorithm to search for zeros of $h(\theta)$. The fourth result indicates that the extraneous zero of $h(\theta)$ given by $\theta = 0$ is either a local maxima or local minima of $h(\theta)$.

The fifth result can be used to determine whether $h(0)$ is a local maxima or local minima. If $h''(0) > 0$ then there are j_n roots on $(\theta_L, 0)$ where j_n is an odd integer and there are j_p roots on $(0, \theta_U)$ where j_p is an odd integer. This follows since $h''(0) > 0$ implies that for some $\epsilon > 0$, $h(\theta - \epsilon) > 0$, and since $h(\theta_L) < 0$, then the number of zeros on $(\theta_L, 0)$ given by j_n must be an odd integer. The argument for j_p odd is similar. In the data sets used investigating the GPD maximum likelihood estimators, it appears that $j_n = j_p = 1$.

If $h''(0) < 0$ then there are j_n roots on $(\theta_L, 0)$ where j_n is zero or an even integer and there are j_p roots on $(0, \theta_U)$ where j_p is zero or an even integer. This follows since $h''(0) < 0$ implies that for some $\epsilon < 0$, $h(\theta - \epsilon) < 0$, and since $h(\theta_L) < 0$, then either there exist no zero of $h(\theta)$ on $(\theta_L, 0)$ or the number of zeros on $(\theta_L, 0)$ given by j_n must be an even integer. The argument for j_p either zero or an even integer is similar. In the data sets used investigating the GPD maximum likelihood estimators, it appears that in many cases $j_n = j_p = 0$. This result agrees with the finding in Hosking and Wallis (1987) indicating that in many cases with $k > 0$ and $n < 25$ the GPD maximum likelihood estimators do not exist. The remaining data sets in the investigation indicated that either $j_n = 0$, $j_p = 2$ or $j_n = 2$, $j_p = 0$ or $j_n = 2$, $j_p = 2$.

The possible existence of multiple zeros of $h(\theta)$ on $\mathcal{B}$ complicates the numerical search, but an algorithm can be designed to find these multiple zeros.

Each zero of $h(\theta)$ indicates a candidate for the local maxima of the log-likelihood. For each of the $j_n + j_p$

zero(s), denoted by $\theta_i^{(0)}$, compute

$$k_i = (1/n)\sum_{i=1}^{n} \ln(1 - \theta_i^{(0)} X_i)$$

$$a_i = \frac{k_i}{\theta_i^{(0)}}.$$

This value must be evaluated using the Hessian matrix of the GPD log-likelihood given in the Appendix to determine if it is a local maxima, local minima or saddle point of the GPD log-likelihood. The point (k_i, a_i) is a local maxima, and therefore considered a candidate for the GPD maximum likelihood estimator, if the Hessian matrix evaluated at the estimators is negative definite.

The pair (k_i, a_i) which has the largest value of $\mathcal{L}_{\text{GPD}}(k_i, a_i; \mathbf{X})$ is identified as the local maximum on the space $\mathcal{A}$ and will be denoted by (k_m, a_m).

2.2 *Boundary Maximum on* $\mathcal{A}$. Any local maxima of the GPD log-likelihood on the domain $\mathcal{A}$ must exceed the log-likelihood evaluated at the boundary in order to be the maximum likelihood estimator. Hence, the second value which must be investigated is at the boundary of $\mathcal{A}$ where $k = 1$. Given $k = 1$, $a > X(n;n)$ then $\mathcal{L}_{\text{GPD}}(k, a; \mathbf{X}) = -n \ln a$. Therefore the boundary maximum, denoted by (k_b, a_b), is given by $k_b = 1$ and $a_b = X(n;n)$. The problem is complicated by the optimization being taken over an open set, but it is treated as a maximum taken over a closed set.

The GPD maximum likelihood estimator, denoted by $(\hat{k}, \hat{a})$, is then given by the local maximum (k_m, a_m) if $\mathcal{L}_{\text{GPD}}(k_m, a_m; \mathbf{X}) > -n \ln X(n;n)$ and is given by the boundary maximum (k_b, a_b) if $\mathcal{L}_{\text{GPD}}(k_m, a_m; \mathbf{X}) < -n \ln X(n;n)$.

If no local maximum is found, then there is no GPD maximum likelihood estimate and the alternative estimators given by Hosking and Wallis (1987) are recommended.

Appendix

Consider the space defined by

$$\mathcal{A} = \{k < 0, a < 0\} \cup \{0 < k < 1, a/k > X(n;n)\}.$$

On the space $\mathcal{A}$, the gradient vector of the GPD log-likelihood has elements

$$\frac{\partial \mathcal{L}_{\text{GPD}}(k, a; \mathbf{X})}{\partial k} = \frac{n}{k}\left(\frac{1}{k} - 1\right) - \frac{1}{k^2}\sum_{i=1}^{n} \ln\left(1 - \frac{kX_i}{a}\right) - \frac{1}{k}\left(\frac{1}{k} - 1\right)\sum_{i=1}^{n}\left(1 - \frac{kX_i}{a}\right)^{-1},$$

$$\frac{\partial \mathcal{L}_{\text{GPD}}(k, a; \mathbf{X})}{\partial a} = -\frac{n}{ka} + \frac{1}{a}\left(\frac{1}{k} - 1\right)\sum_{i=1}^{n}\left(1 - \frac{kX_i}{a}\right)^{-1}.$$

On the space $\mathcal{A}$, the Hessian matrix of the GPD log-likelihood has elements

$$\frac{\partial^2 \mathcal{L}_{\text{GPD}}(k, a; \mathbf{X})}{\partial k^2} = \frac{n}{k^2}\left(1 - \frac{3}{k}\right) + \frac{2}{k^3}\sum_{i=1}^{n} \ln\left(1 - \frac{kX_i}{a}\right) + \frac{2}{k^2}\left(\frac{2}{k} - 1\right)\sum_{i=1}^{n}\left(1 - \frac{kX_i}{a}\right)^{-1} - \frac{1}{k^2}\left(\frac{1}{k} - 1\right)\sum_{i=1}^{n}\left(1 - \frac{kX_i}{a}\right)^{-2},$$

$$\frac{\partial^2 \mathcal{L}_{\text{GPD}}(k, a; \mathbf{X})}{\partial a^2} = \frac{n}{ka^2} - \frac{1}{a^2}\left(\frac{1}{k} - 1\right)\sum_{i=1}^{n}\left(1 - \frac{kX_i}{a}\right)^{-2},$$

$$\frac{\partial^2 \mathcal{L}_{\text{GPD}}(k, a; \mathbf{X})}{\partial k \partial a} = \frac{n}{k^2 a} - \frac{1}{ka}\left(\frac{2}{k} - 1\right)\sum_{i=1}^{n}\left(1 - \frac{kX_i}{a}\right)^{-1} + \frac{1}{ka}\left(\frac{1}{k} - 1\right)\sum_{i=1}^{n}\left(1 - \frac{kX_i}{a}\right)^{-2}.$$

References

Davison, A. C. (1984), "Modelling excesses over high thresholds, with an application," in *Statistical Extremes and Applications*, ed. J. Tiago de Oliveira, Dordrecht: Reidel, pp. 461–482.

DuMouchel, W. (1983), "Estimating the stable index α in order to measure tail thickness: A Critique," *Annals of Statistics*, 11, 1019–1036.

Hosking, J. R. M. and Wallis, J. R. (1987), "Parameter and quantile estimation for the Generalized Pareto Distribution," *Technometrics*, 29, 339–349.

Joe, H. (1987), "Estimation of quantiles of the maximum of N observations," *Biometrika*, 74, 347–354.

Pickands, J. (1975), "Statistical inference using extreme order statistics," *Annals of Statistics*, 3, 119–131.

Smith, J. A. (1986), "Estimating the upper tail of flood frequency distributions," Technical Report MS–R8607, Dept. Math. Statist., Centrum voor Wiskunde en Informatica, Amsterdam.

Smith, R. L. (1984), "Threshold methods for sample extremes," in *Statistical Extremes and Applications*, ed. J. Tiago de Oliveira, Dordrecht: Reidel, pp. 621–638.

Smith, R. L. (1987), "Estimating tails of probability distributions," *Annals of Statistics*, 15, 1174–1207.

ASYMPTOTIC EFFICIENCY OF THE MAXIMUM LIKELIHOOD ESTIMATOR OF A PARAMETER FOR THE M/G/1 QUEUEING SYSTEM

SUDHA JAIN*

Department of Mathematics & Statistics
Queen's University, Kingston
Canada, K7L 3N6

ABSTRACT.

This paper discusses asymptotic efficiency of the maximum likelihood estimator of the parameters of the M/G/1 queueing system for full likelihood and reduced likelihood functions. The efficiency of the maximum likelihood estimator of the reduced likelihood function relative to full likelihood function is derived.

1. INTRODUCTION.

Clarke (1957) discussed the estimation problem of traffice intensity for M/M/1 queueing system using maximum likelihood principles. The problem of statistical inference for birth and death processes was considered as Markov processes by Wolff (1965). A large sample theory based on maximum liklihood theory for Markov processes developed by Billingsley (1961) was applied to make inference for arrival and service rates. Jenkins (1972) estimated the maximum likelihood estimate of mean waiting time in the simple M/M/1 queue under conditions of incomplete information. In 1981, Basawa and Prabhu considered the single server queueing model and obtained estimates for interarrival and service times distribution functions without assuming the steady-state. In this paper, asymptotic efficiencies of the estimators for the M/G/1 queueing model are derived for full likelihood and reduced likelihood functions based on Lehmann's (1983) work. Asymptotic relative efficiency of the estimator is obtained as the square of the correlation coefficient between estimators.

2. ESTIMATION PROCEDURES.

Let interarrival and service times be independent, identically distributed random variables. Their densities are defined by $f(t,\theta)$ and $g(x,\phi)$, respectively, where θ and ϕ are the parameters to be estimated. Denote

Interarrival times $\{t_k, k \geq 1\}$

and

Service times $\{x_k, k \geq 1\}$.

Initially customers arrive at t equals zero and observe the queue until the first n customers departed. Let the service times of these customers be $x_1, x_2, \ldots, x_n$. Let nth departure occur at D_n time. Observe the interarrival times of all customers during the interval $(0, D_n)$. Let their interarrival time be $t_1, t_2, \ldots, t_{N_A}$, where

$$N_A = N_A(D_n) = \max(k : t_1 + t_2 + \cdots + t_k \leq D_n).$$

Here,

$$N_A \geq n.$$

Likelihood function for estimating the parameters θ and ϕ is given by

$$L(\theta,\phi) = \left(\prod_{i=1}^{N_A} f(t_i;\theta)\left(\prod_{i=1}^{n} g(x_i;\phi)\right)\right)(1 - F(Z_n;\theta)), \tag{1}$$

where $(1-F(Z_n;\theta))$ corresponds to the incomplete arrival interval when sampling is terminated at the epoch D_n; and $Z_n = D_n - \sum_{i=1}^{N_A} t_i$.

Basawa and Prabhu (1981) considered the following reduced likelihood function:

$$L^a(\theta,\phi) = \left(\prod_{i=1}^{N_A} f(t_i;\theta)\right)\left(\prod_{i=1}^{n} g(x_i;\phi)\right). \tag{2}$$

Equation (2) is an approximation of equation (1). Taking the logarithms in equation (2) and differentiating with respect to θ and ϕ and equating to zero, we get

$$\frac{\partial L^a}{\partial \theta} = \sum_{i=1}^{N_A} \frac{\partial}{\partial \theta} \ln f(t_i;\theta) = 0 \tag{3}$$

*Research Supported by A.R.C. Grant (Queen's University).

and

$$\frac{\partial L^a}{\partial \phi} = \sum_{i=1}^{n} \frac{\partial}{\partial \phi} \ln g(x_i;\phi) = 0. \tag{4}$$

Let $\hat{\theta}_n^a$ and $\hat{\phi}_n^a$ are the estimates of θ and ϕ based on reduced likelihood function. Asymptotic properties of these estimators are given by Basawa and Prabhu (1981). Taking logaithms in full likelihood equation (1), we have

$$L' \equiv \ln L(\theta,\phi) = \sum_{i=1}^{N_A} \ln f(t_i;\theta) + \sum_{i=1}^{n} \ln g(x_i;\phi) + \ln(1 - F(Z_n;\theta)). \tag{5}$$

Differentiating equation (5) partially with respect to θ and ϕ and equating to zero, we have

$$\frac{\partial L'}{\partial \theta} = \sum_{i=1}^{N_A} \frac{\partial}{\partial \theta} \ln f(t_i;\theta) + H(Z_n;\theta) = 0 \tag{6}$$

and

$$\frac{\partial L'}{\partial \phi} = \sum_{i=1}^{n} \frac{\partial}{\partial \phi} \ln g(x_i;\phi) = 0, \tag{7}$$

where

$$H(Z_n;\theta) = \frac{\partial}{\partial \theta} \ln(1 - F(Z_n;\theta)).$$

Let $\hat{\theta}_n$ and $\hat{\phi}_n$ are the estimates of θ and ϕ based on the full likelihood function which can be obtained by solving equations (6) and (7). Comparing equations (6) and (7) with equations (3) and (4), it is clear

$$\hat{\phi}_n = \hat{\phi}_n^a \text{ and } \hat{\theta}_n \text{ differs from } \hat{\theta}_n^a.$$

It can be shown in the following particular case that $\hat{\theta}_n$ and $\hat{\theta}_n^a$ are asymptotically equivalent.

M/G/1 queue

Let the probability density function of mean interarrival times θ be given by

$$f(t,\theta) = (\theta)^{-1} \cdot \exp(-t/\theta).$$

Then the distribution function is

$$F(t,\theta) = 1 - \exp(-t/\theta).$$

Let the probability density function of the service time be $g(x,\phi)$. Consider equation (3), we have

$$\sum_{i=1}^{N_A} \frac{\partial}{\partial \theta} \ln(\theta^{-1} \exp(-t_i/\theta)) = 0. \tag{8}$$

Simplification of the left side of equation (8) yields

$$-\frac{1}{\theta} N_A + \frac{1}{\theta^2} \sum_{i=1}^{N_A} t_i = 0. \tag{9}$$

Now considering equation (6)

$$\frac{\partial L'}{\partial \theta} = \sum_{i=1}^{N_A} \frac{\partial}{\partial \theta} \ln(\theta^{-1} \exp(-t_i/\theta)) + \frac{\partial}{\partial \theta} \ln(\exp(-Z_n/\theta)) = 0. \tag{10}$$

Simplifying equation (10), we get

$$-\frac{N_A}{\theta} + \frac{1}{\theta^2} \sum_{i=1}^{N_A} t_i + \frac{Z_n}{\theta^2} = 0. \tag{11}$$

Solving equation (9), we get

$$\hat{\theta}_n^a = \frac{\sum_{i=1}^{N_A} t_i}{N_A}. \tag{12}$$

Solving equation (11), we have

$$\hat{\theta}_n = \frac{\sum_{i=1}^{N_A} t_i + Z_n}{N_A} = \frac{D_n}{N_A}. \tag{13}$$

3. ASYMPTOTIC EFFICIENCY OF THE ESTIMATORS.

In this section, asymptotic efficiencies of the estimators are derived for some particular queueing models. Some of the preliminary results related to asymptotic properties of the estimates based on Lehmann's (1983) work are reviewed.

Preliminary Results

The following theorem establishes that any consistent root of the likelihood equation is asymptotically normal and efficient.

Theorem 3.1. Suppose that $X_1, X_2, \ldots, X_n$ are independent, identically distributed and satisfy appropriate regularity assumptions [Lehmann (1983), pages 406 and 415] then any consistent sequence $\hat{\theta}_n = \theta_n(X_1, X_2, \ldots, X_n)$ of roots of the likelihood equation satisfies

$$\sqrt{n}(\hat{\theta}_n - \theta) \to N(0, (I(\theta))^{-1}), \tag{14}$$

where

$$I(\theta) = E\left(\frac{\partial}{\partial \theta} \ln f(X,\theta)\right)^2.$$

Remark 3.1. Any sequence $\hat{\theta}_n^*$ of estimators satisfying equation (14) will be said to be asymptotically efficient.

Suppose that δ_n is any consistent estimator of θ, and the assumptions of *Theorem 3.1* hold, then the root $\hat{\theta}_n$ of the likelihood equation closest to θ_n is also consistent [Lehmann (1983), Theorem 2.2, page 430].

The usual iterative methods for solving the likelihood equation

$$L'(\theta) = 0$$

are based on replacing the left side by the linear terms of a Taylor's series expansion about an approximate solution $\tilde{\theta}$. If $\hat{\theta}$ denotes a root of the likelihood equation ($L'(\theta) = 0$) then this leads to the approximation

$$0 = L'(\hat{\theta}) \approx L'(\tilde{\theta}) + (\hat{\theta} - \tilde{\theta})L''(\tilde{\theta})$$

and hence

$$\hat{\theta} = \tilde{\theta} - \frac{L'(\tilde{\theta})}{L''(\tilde{\theta})} \tag{15}$$

the procedure is then to iterate by replacing $\tilde{\tilde{\theta}}$ by the value of $\tilde{\theta}$ of the right side of the equation (15) and so on.

The following *theorems* give conditions on $\tilde{\theta}$ under which the resulting sequence of estimators is consistent, asymptotically normal and efficient.

Theorem 3.2 Suppose that the assumptions of *theorem 3.1* hold and that $\tilde{\theta}_n$ is not only a consistent but a $\sqrt{n}$-consistent estimator of θ, that is, that $\sqrt{n}(\tilde{\theta}_n - \theta)$ is bounded in probability so that $\tilde{\theta}_n$ tends to θ at least at the rate of $(\sqrt{n})^{-1}$. Then the estimator sequence

$$\delta_n = \tilde{\theta}_n - \frac{L'(\tilde{\theta}_n)}{L''(\tilde{\theta}_n)} \tag{16}$$

is asymptotically efficient, that is, it satisfies equation (14) with δ_n in place of $\hat{\theta}_n$.

Theorem 3.3. Suppose that the assumptions of theorem 3.2 hold and that $I(\theta)$ is a continuous function of θ. Then the estimator

$$\delta'_n = \tilde{\theta}_n + \frac{L'(\tilde{\theta}_n)}{nI(\tilde{\theta}_n)} \tag{17}$$

is also asymptotically efficient.

M/M/1-queue

The estimates for M/M/1 queue with mean interarrival time θ and mean service time ϕ using full likelihood and reduced likelihood equations are

$$\hat{\theta}_n = \frac{D_n}{N_A},$$

$$\hat{\theta}^a_n = (N_A)^{-1} \sum_{i=1}^{N_A} t_i$$

and

$$\hat{\phi}_n = \hat{\phi}^a_n = (n)^{-1} \sum_{i=1}^{n} x_i.$$

Theorem 3.3 can be applied (for fixed number of observations) to show that $\hat{\theta}_n$ is asymptotically efficient. [Basawa and Prabhu (1981), page 479]. Rewriting equation (10), we have

$$\frac{\partial L'}{\partial \theta} = \sum_{i=1}^{N_A} \frac{\partial}{\partial \theta} \ln((\theta)^{-1} \exp(-t_i/\theta)) + \frac{\partial}{\partial \theta} \ln(\exp(-Z_n/\theta))$$

$$= -\frac{N_A}{\theta} + \frac{1}{\theta^2} \sum_{i=1}^{N_A} t_i + \frac{1}{\theta^2} Z_n.$$

Therefore,

$$\left(\frac{\partial L'}{\partial \theta}\right)_{at \hat{\theta}^a_n} = \frac{Z_n N_A^2}{(\sum t_i)^2} \tag{18}$$

and

$$I = E\left(\frac{\partial}{\partial \theta} \ln f(t, \theta)\right)^2$$

$$= E\left(\frac{\partial}{\partial \theta} \ln((\theta)^{-1} \exp(-t/\theta))\right)^2 = \frac{1}{\theta^2}. \tag{19}$$

Therefore,

$$(I)_{at \hat{\theta}^a_n} = \frac{(N_A)^2}{\left(\sum_{i=1}^{N_A} t_i\right)^2}. \tag{20}$$

Using equation (17), we have

$$\delta'_n = (N_A)^{-1} \sum_{i=1}^{N_A} t_i + \frac{\left(\frac{\partial L'}{\partial \theta}\right)_{at \hat{\theta}^a_n}}{N_A (I)_{at \hat{\theta}^a_n}} \tag{21}$$

Hence, equation (21) after simplification yields

$$\delta'_n = \frac{1}{N_A} \sum_{i=1}^{N_A} t_i + \frac{Z_n \cdot N_A^2 / \left(\sum_{i=1}^{N_A} t_i\right)^2}{N_A \cdot \left(N_A / \sum_{i=1}^{N_A} t_i\right)^2}. \tag{22}$$

Thus,

$$\delta'_n = \frac{1}{N_A} \sum_{i=1}^{N_A} t_i + \frac{Z_n}{N_A} = \frac{D_n}{N_A}. \tag{23}$$

Since $\hat{\theta}_n = \frac{D_n}{N_A}$, we have

$$\delta'_n = \hat{\theta}_n. \tag{24}$$

It can be easily seen from *Theorem 3.1* that

$$\sqrt{n}(\hat{\theta}_n - \theta) \approx N\left(0, \frac{1}{I(\theta)}\right) = N(0, \theta^2), \tag{25}$$

which implies that $\hat{\theta}_n$ is asymptotically efficient.

4. ASYMPTOTIC EFFICIENCY AND ITS RELATIONSHIP WITH CORRELATION COEFFICIENT.

Let $\hat{\theta}^a_n$ be the likelihood equation estimator such that

$$U_n = \sqrt{n}(\hat{\theta}^a_n - \theta) \approx N(0, \sigma_1^2), \tag{26}$$

where

$$\widehat{\sigma}_1^2 = (\widehat{\theta}_n^a)^2 = \left(\sum_{i=1}^{N_A} \frac{t_i}{N_A}\right)^2 = \left(\frac{D_n - Z_n}{N_A}\right)^2. \quad (27)$$

Let $\widehat{\theta}_n$ be a $\sqrt{n}$-consistent estimator such that

$$V_n = \sqrt{n}(\widehat{\theta}_n - \theta) \approx N(0, \sigma_2^2), \quad (28)$$

where

$$\widehat{\sigma}_2^2 = \widehat{\theta}_n^2 = \left(\frac{D_n}{N_A}\right)^2. \quad (29)$$

The estimate of asymptotic relative efficiency (ARE) of V_n with respect to U_n is

$$e = \frac{\widehat{\sigma}_1^2}{\widehat{\sigma}_2^2} \quad (30)$$

If $Z_n \to 0$, then $e = 1$.

If $Z_n \to D_n$, then $e = 0$.

The following *theorem* establishes the relationship of ARE with correlation of U_n and V_n.

Theorem 4.1 Let U_n and V_n tend to bivariate normal distribution given by

$$\begin{pmatrix} U_n \\ V_n \end{pmatrix} \approx N_2 \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} \right\},$$

where

$$\sigma_{11} = \sigma_1^2 \text{ and } \sigma_{22} = \sigma_2^2, \text{ and } \sigma_2^2 \geq \sigma_1^2.$$

Then, ARE of V_n with respect to U_n is given by

$$e = \rho^2,$$

where

$$\rho = \frac{\sigma_{12}}{(\sigma_{11}\ \sigma_{22})^{\frac{1}{2}}}$$

is the correlation coefficient between U_n and V_n.

Proof. Proof requires only to show that

$$\sigma_{12} = \sigma_{11} = \sigma_1^2.$$

Consider

$$\begin{aligned} Z &\equiv \text{Var}[(1-\alpha)U_n + \alpha V_n] \\ &= \alpha^2[\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}] + \alpha[-2\sigma_1^2 + 2\sigma_{12}] + \sigma_1^2. \end{aligned} \quad (31)$$

The quantity Z is nonnegative and approaches the minimum value when $\alpha = 0$, since $\widehat{\theta}_n^a$ is asymptotically efficient. Thus, for minimization, differentiating Z with respect to α, and equating to zero, we have

$$\frac{\partial Z}{\partial \alpha} = 2\alpha(\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}) + (-2\sigma_1^2 + 2\sigma_{12}) = 0. \quad (32)$$

Solving equation (32), we get

$$\alpha = \frac{\sigma_1^2 - \sigma_{12}}{\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}}. \quad (33)$$

Since $\alpha = 0$, we have

$$\sigma_1^2 = \sigma_{12}. \quad (34)$$

Using $\sigma_{11} = \sigma_1^2$ and $\sigma_{22} = \sigma_2^2$, we have

$$\rho^2 = \left(\frac{\sigma_1^2}{\sigma_1\sigma_2}\right)^2 = \frac{\sigma_1^2}{\sigma_2^2}. \quad (35)$$

But σ_1^2/σ_2^2 is the ARE of V_n with respect to U_n.

Remark 4.1. The *theorem 4.1* clearly indicates that under sufficiently regular conditions, good efficiency of $\{\widehat{\theta}_n^a\}$ is equivalent to high correlation with $\{\widehat{\theta}_n\}$.

Acknowledgement

The author is grateful to Professor J. G. C. Templeton for his comments.

References

Basawa, I. V. and Prabhu, N. U. (1981), Estimation in single server queues, *Naval Res. Log. Quart.*, **28**,475-487.

Billingsley, P. (1961), Statistical Inference for Markov Processes, *University of Chicago Press.*

Clark, A. B. (1957), Maximum likelihood estimates in a simple queue, *Annals of Math. Statist.*, **28**, 1036-1040.

Jenkins, J. H. (1972), The relative efficiency of direct and maximum likelihood estimators of mean waiting time in the simple queue M/M/1, *Jour. of Appl. Prob.*, **9**, 396-403.

Lehmann, E. L. (1983), Theory of Point Estimation, *Wiley.*

Wolff, R. W. (1965), Problems of statistical inference for birth - death queueing models., *Oper. Res.*, **13**, 343-357.

ADDRESS LIST FOR INTERFACE '91 REGISTRANTS

David A. Adler
University of Washington
Pathology SM-30
Seattle, WA 98195

Pramila Agarwala
AT&T Bell Laboratories
Crawfords Corner Rd, 2K515
Holmdel, NJ 07735
email: att!haqaa!pramila

Demissie Alemayehu
American Cyanamid Com.
60/203 Stat. Dept.,Lederle Labs.
Pearl River, NY 10965

John Aleong
University of Vermont
Math and Stat
Burlington, VT 05405
email: J_ALEONG@UVMVAX.bitnet

Barney Alm
Numerical Algorithms Group Inc.
1400 Opus Place, Suite 200
Downers Grove, IL 60515

Russell Almond
University of Washington
GN-2
Seattle, WA 98195
email: almond@stat.washington.edu

Roberto E. Altschul
Boeing Computer Services
6826 31st Ave. NE
Seattle, WA 98115

Barbara Alvin
Eastern Washington University
Math Dept., ons-32
Cheney, WA 99004

Kevin Anderson
1303 NE Campus Park Way, #105
Seattle, WA 98105

Mark Andrews
CSIRO, Div. of Maths & Stats
PO Box 218
Lindfield NSW 2070
Australia
email: marka@syd.dms.csiro.au

Greg Anglin
University of Waterloo
200 Univ. Ave., W.,
Dept of Statistics
Waterloo, Ontario N2L 3G1
Canada
email: dganglin@watstat.waterloo.edu

Peter Arzberger
National Science Foundation
1800 G St., NW
Washington, DC 20550
email: parzberg@note.nsf.gov

Taskin Atilgan
AT&T Bell Labs
600 Mountain Ave.
Murray Hill, NJ 07974

E. Neely Atkinson
UTMDACC, Dept. of Biomathematics
1515 Holcombe Blvd., Box 237
Houston, TX 77030
email: an123651@uthvm1.bitnet

Tom Siu-Tong Au
AT&T Bell Laboratories
7B-517, 600 Mountain Ave.
Murray Hill, NJ 07974

Reo Audette
Simon Fraser University
Academic Computing Service
Burnaby, B.C. V5A 1S6
Canada
email: userraid@cc.sfu.ca

Gail B. Badner
Searle
4901 Searle Parkway
Skokie, IL 60077

Helen M. Barr
University of Washington Medical School
6033 29th Ave. NE
Seattle, WA 98115

Douglas Bates
University of Wisconsin
Dept. of Stat., 1210 W.Dayton St.
Madison, WI 53706-1693
email: bates@stat.wis.edu

Ron Baxter
CSIRO, Div Maths & Stats
PO Box 218
Lindfield, 2070, NSW
Australia
email: ronb@syd.dms.csiro.au

Richard A. Becker
AT&T Bell Laboratories
600 Mountain Ave. Rm. 2C259
Murray Hill, NJ 07974
email: rab@research.att.com

Edward J. Bedrick
University of New Mexico
Dept., Math and Stat.
Albuquerque, NM 87131

Kerry G. Bemis
Eli Lilly and Company
Lilly Corporate, Center, DC2233
Indianapolis, IN 46285

Peter M. Bentler
University of California
Dept. of Psych., 4627 Franz Hall
Los Angeles, CA 90024-1563

Mark Berman
CSIRO
PO Box 218
Lindfield 2070, NSW
Australia
email: mark@syd.dms.csiro.au

Julian Besag
U. of Newcastle upon Tyne
Dept. of Math. and Stat.
Newcastle upon Tyne, NE1 7RU
U.K.
email: julian.besag@ncl.ac.uk

M.E. Bock
Purdue University
1399 Math Bldg.
W. Lafayette, IN 47906-1399

Fred L. Bookstein
University of Michigan
1547 Washtenaw Ave.
Ann Arbor, MI 48104
email: Fred_L._Bookstein@UM.CC.Umich.EDU

Kimiko O. Bowman
Oak Ridge National Laboratory
PO Box 2008, Bldg. 6012A, MS-6367
Oak Ridge, TN 37831-6367
email: bowman@epm.ornl.gob

Elbert Branscomb
Lawrence Livermore Natl. Labs.
PO Box 5507
Livermore, CA 94551

Barry W. Brown
UTMDACC, Dept. of Biomathematics
1515 Holcombe Blvd., Box 237
Houston, TX 77030
email: an12bwb1@uthvm1.bitnet

Peter F. Brown
IBM
Room J2-H22, PO box 704
Yorktown Heights, NY 10598
email: PBROWN@IBM.COM

Andreas Buja
Bellcore, MRE 2Q-362
445 South Street
Morristown, NJ 07962-1910
email: andreas@bellcore.com

Henning Bunzel
University of Aarhus, Inst. of Ec.
Building 350
DK-8000 Aarhus C
Denmark

Patrick J. Burns
Statistical Sciences, Inc.
1700 Westlake Ave., N Suite 500
Seattle, WA 98109
email: pat@statsci.com

Gregory Campbell
National Institutes of Health
Bldg. 12A, Room 3045
Bethesda, MD 20892
email: GGC@NIHCU

M.T. Canhao
University of Zimbabwe
P.O. Box 101
Harare, Zimbabwe

Daniel M. Cap
BF Goodrich Research & Dev.
9921 Brecksville Road
Brecksville, OH 44141

Vincent Carey
Johns Hopkins School, Public Health
4 Upland Rd. #12
Baltimore, MD 21210

Daniel B. Carr
George Mason University
SITE 1 , ORAS
Fairfax, VA 22039
email: dcarr@gmuvax2.gmu.edu

Tom Caudell
Boeing Computer Service
MS 7L-22,PO Box 24346
Seattle, WA 98124-0346

Kung-Sik Chan
University of Chicago
5734 University Ave., Dept. of Stat.
Chicago IL 60737
email:Chan@Galton.Uchicago.edu

Joseph T. Chang
Yale University
Statistics Dept., Box 2179
New Haven, CT 06520-2179

Hung Chen
State University of New York
Dept. of Applied Math. and Stat.
Stony Brook, NY 11794-3600

Ling Chen
Florida Inter. University
University Park
Miami, FL 33199
email: chenl@servax.bitnet

Omar Cherkaoui
Universite du Quebec a Montreal
C.P. 8888 Succ. A
Montreal, Quebec H3C 3P8
Canada
email: Cherkaou@mips1.uqam.ca

Daniel C. Chin
The John Hopkins University/APL
John Hopkins Rd.
Laurel, MD 20723

Hugh Chipman
University of Waterloo
Dept. of Statistics
Waterloo, Ontario N2L 3G1
Canada
email: hachipman@violet.waterloo.edu

Tzi-cker Chiueh
U.C. Berkeley
Evans Hall, Computer Sc. Div.
Berkeley, CA 94720
email: chiueh@sprite.berkeley.edu

Bruce C. Chow
435 Athens Street
San Francisco, CA 94112

Kenneth Church
AT&T Bell Laboraotries
600 Mountain Avenue, 2D-444
Murray Hill, NJ 07974
email: church@venera.isi.edu

John Cibulskis
Searle
4901 Searle Parkway
Skokie, IL 60077

Linda Clark
AT&T Bell Laboratories
600 Mountain Ave.
Murray Hill, NJ 07974
email: lac@research.att.com

Kevin J. Coakley
National Inst. of Standards and Tech.
Administration Bldg. Rm. A337
Gaithersburg, MD 20899
email: kevin@stat.cam.nist.gov

Fred Cohen
University of California
Box 0446
San Francisco, CA 94143
email: cohen@cgl.ucsf.edu

Jarrell D. Collier
Litton Data Systems
8000 Woodley Avenue
Van Nuys, CA 91412

Thomas W. Colthurst
Brown University
Dept. of Math., PO Box 5727
Providence, RI 02912

Michael Conlon
University of Florida
Box J-212, J. H. Miller Health Cen.
Gainesville, FL 32610
email: mconlon@stat.ufl.edu

Di Cook
Rutgers University
Stat. Dept., Hill Cen., Busch
New Brunswick, NJ 08904
email: dcook@stat.rutgers.edu

Joseph S. Costa Jr.
National Security Agency
9800 Savage Road
Fort G. G. Meade, MD 20755-6000

Dennis D. Cox
University of Illinois
101 Illini Hall, Dept. of Stat.
Champaign, IL 61821
email: dcox@vmd.cso.uiuc.edu

Stuart L. Crawford
Advanced Decision Systems
1500 Plymouth Street
Mountain View, CA 94043-1230

Rob Creecy
Census Bureau
SRD 3215-4
Washington, DC 20233

Neil Crellin
Stanford University
PO Box 8879, Dept. of Stat.
Stanford, CA 94309
email: neilc@wallaby.stanford.edu

John J. Crowley
Hutchinson Ca Res. Ctr.
1124 Columbia St.
Seattle, WA 98104
email: John@fhcrcvm.bitnet

Kazimierz Dadak
Fordham University
2120 Wallace Ave., 4A
Bronx, NY 10462
email: dadak@fordmurh.bitnet

Siddhartha R. Dalal
Bellcore 2P392
445 South St
Morristown, NJ 07962-1910

Christian J. Darken
Yale University, Computer Sc. Dept.
P.O. Box 2158 Yale Station
New Haven, CT 06520
email: darken@cs.yale.edu

Charmaine Dean
Simon Fraser University
Dept. of Math & Stat.
Burnaby, BC V5A 1S6
Canada

Michael DeCrescenzo
American Health Network
3988 N. Central Expwy.
Dallas, TX 75204

Joseph Deken
California Museum of Sc. & Industry
21833 Kent Avenue
Los Angeles, CA 90503

Lorraine Denby
AT&T Bell Laboratories
Rm 2C-255, 600 Mountain Ave.
Murray Hill, NJ 07974
email: ld@research.att.com

Gene Denzel
York University, 4700 Keele St.
North York, Ontario M3J aP3
Canada

Thomas F. Devlin
Montclair State College
10 Symor Drive
Convent Station, NJ 07961
email: devlin@mozart.montclair.edu

Nadine Dilworth
University of Idaho
703 S. Adams
Moscow, ID 83843

Kim-Anh Do
Australian National University
Dept. of Statistics
Canberra, ACT 2601
Australia
email: dokstat@statl.anu.oz.au

Curt Doetkott
North Dakota State University
S. Eng. 216, PO Box 5164
Fargo, ND 58105
email: NUOZO312@ndsuvm1.bitnet

Miriam G. Donoho
2830 Buena Vista Way
Berkeley, CA 94708

Deva C. Doss
Canadian Union College
Box 430
College Heights, Alberta T0C 0Z0
Canada

Steve Duke
Weyerhaeuser and Co.
WTC 1B20
Tacoma, WA 98477

Bill Dunlap
Statistical Sciences, Inc.
1700 Westlake Ave. N., Suite 500
Seattle, WA 98109
email: bill@statsci.com

Rex A. Dwyer
N. Carolina State U.
Computer Science, Box 8206
Raleigh, NC 27695-8206
email: DWYER@CSC.NCSU.EDU

William F. Eddy
Carnegie Mellon University
Department of Statistics
Pittsburgh, PA 15213
email: bill@stat.cmu.edu

Mary Emond
University of Washington
3930 Linden Ave. N. No. 301
Seattle, WA 98103

Laszlo Engelman
SYSTAT, Inc.
1800 Sherman Ave
Evanston, IL 60201

Katherine B. Ensor
Rice University
P.O. Box 1892
Houston, Texas 77251-1892
email: ensor@rice.edu

Leonardo D. Epstein
The Johns Hopkins University
Dept. of Biostat., 615 N. Wolfe
Baltimore, MD 21205

Joseph Felsenstein
University of Washington
Department of Genetics
Seattle, WA 98195
email: joe@genetics.washington.edu

David R. Ferguson
Boeing Computer Services
MS 7L-21, PO Box 24346
Seattle, WA 98124

Juan Ferrandiz
University of Valencia
Dr. Moliner 50, Brujasot
Valencia, Spain 46100

Scott Fertig
Yale University
Department of Computer Science
New Haven, CT 06520-2158

Leonid Feygin
SPSS, Inc.
444 N. Michigan, 30th Fl.
Chicago, IL 60611

Gwennaele Fichant
Los Alamos National Laboratory
Theoretical Bio. and Bioph, Gr. T-10
Los Alamos, NM 87545
email: gaf@trna.lanl.gov

Nicholas I. Fisher
CSIRO Div. of Math. & Stat.
PO Box 218
Lindfield, NSW 2070
Australia
email: nickf@syd.dms.csiro.au

Gilbert FitzGerald
University of California
Hilgard Ave.
Los Angeles, CA 90024-1563

Leska Fore
University of Washington
CQS HR-20
Seattle, WA 98195
email: leska@cqs.washington.edu

Dean Foster
University of Chicago
1101 E. 58th St.
Chicago, IL 60637
email: foster@junior@uchicago.edu

Jerome Friedman
Stanford University
Sequoia Hall
Stanford, CA 94305

Dargan Frierson Jr.
UNC-W, Dept. of Mathematical Sc.
601 S. College Rd.
Wilmington, NC 28403
email: frierson@vxc.uncwil.edu

Jonathan B. Fry
SPSS Inc.
444 N. Michigan Ave.
Chicago, IL 60611

Thomas A. Furness, III
Industrial Engineering, FU-20
Human Interface Technology Lab
University of Washington
Seattle, WA 98195

Fah Fatt Gan
National University of Singapore
10 Kent Ridge Crescent
Singapore 0511
Republic of Singapore
email: bitnet: matganff@nusvm

Dan Geiger
Northrop Research & Tech. Center
One Research Park
Palos Verdes Pen[illegible], CA 90274
email: dgeiger@n[illegible]northrop.com

Alan Genz
Washington State University
School of EE and Computer Sc.
Pullman, WA 99164-2752

Alexander A. Georgiev
Medical University of S. Carolina
171 Ashley Ave.
Charleston, SC 29425-2503
email: georgiev@musc.bitnet

Thomas Gerling
Johns Hopkins Univ. Applied Physics Lab.
47-116 Johns Hopkins Road
Laurel, MD 20707

John Geweke
University of Minnesota
Department of Economics
Minneapolis, MN 55455
email: geweke@atlas.socsci.umn.edu

Charles Geyer
1400 E. 55th Pl, Apt. 807
Chicago, IL 60637
email: geyer@galton.uchicago.edu

Krishnendu Ghosh
University of Montana
Dept. of Math. Science
Missoula, MT 59812

Rudy Gideon
University of Montana
Dept. of Mathematical Sciences
Missoula, MT 59812-1032

Sheri Gilley
SPSS Inc.
444 N Michigan Ave.
Chicago, IL 60611

Claude Ginsburg
Boeing Computer Services
7L-46, P.O. Box 24346
Seattle, WA 98124-0346

Ram Gnanadesikan
425 Fairmont Avenue
Chatham, NJ 07928

David Gomberg
UCSF
Seven Gateview Court
San Francisco, CA 94116-1941
email: GOMBERG@UCSFVM.bitnet

Colin Goodall
Princeton University
Program in Stat., E-Quad E220
Princeton, NJ 08544
email: colin@jackknife.princeton.edu

Arnold Goodman
County of Los Angeles
18231 Hillcrest Circle
Villa Park, CA 92667

Ted A. Gooley
Box 344
Harrington, WA 99134

Michael L. Goris
Div. of Nuclear Medicine
Stanford U. School of Medicine
Stanford, CA 94305

Michael Grant
14206 Golden Woods
San Antonio, TX 78249

William R. Greco
Roswell Park Cancer Inst.
Carlton & Elm Streets
Buffalo, NY 14263
email: rosgreco@ubvms

Scott D. Grimshaw
University of Maryland
College of Business and Management
College Park, MD 20742

Georges Grinstein
Computer Science Department
University of Lowell
Lowell, MA 01854

Eric Grosse
AT&T Bell Laboratories
281 Timber Drive
Berkeley Heights, NJ 07922
email: eht@research.att.com

Sunwei Gui
University of Washington
2260 NE 53 St.
Seattle, WA 98105
email: swguo@spock.biostat.washington.edu

Donald Guthrie
UCLA
760 Westwood Plaza
Los Angeles, CA 99924-1759

Peter Guttorp
University of Washington
Dept. of Stat., GN-22
Seattle, WA 98195

Perry Haaland
Becton Dickinson Research Cen.
P.O. Box 12016
Research Triangle Park, NC 27709
email: pdh@bdrc.bd.com

Michael L. Hand
Willamette University
Grad School of Management
Salem, OR 97301

James W. Hardin
Texas A&M University
3006 Bluestem
College Station, TX 77840

E. James Harner
West Virginia University
Dept. of Stat. & Computer Sc.
Morgantown WV 26506

John Haslett
Trinity College
Dept. of Statistics
Dublin 2 Ireland
email: jhaslett@vax1.tcd.ie

Trevor Hastie
AT&T Bell Laboratories
2C261, 600 Mountain Ave.
Murray Hill, NJ 07974

David R. Haynor
University of Washington
Dept. of Radiology RC-05
Seattle, WA 98195

Michael J. Healy
Boeing Computer Services
PO Box 24346, MS 7L-22
Seattle, WA 98124-0346
email: mjhealy@atc.boeing.com

Leonard B. Hearne
George Mason University
242 Science and Tech. Building
Fairfax, VA 22030
email: COMPSTAT@GMUVAX.GMU.EDU

Nancy Heckman
The Univ. of British Columbia
Statistics Dept.
Vancouver, BC V6T 1W5
Canada
email: nancy@stat.ubc.ca

Richard Heiberger
Temple University
Department of Statistics
Philadelphia, PA 19122
email: rmh@gaia.stat.temple.edu

Joe R. Hill
EDS Research
5951 Jefferson St. NE
Albuquerque, NM 87109
email: joe@edsr.eds.com

Mary Ann Hill
SYSTAT., Inc.
1800 Sherman Ave
Evanston, IL 60201

Erin M. Hodgess
Temple University
Dept. of Stat., Speakman Hall
Philadelphia, PA 19122
email: erin@gaia.stat.temple.edu

Sally E. Howe
National Inst. of Standards & Tech.
Building 225, Room B146
Gaithersburg, MD 20899
email: howe@cam.nist.gov

Ken Hung
Western Washington University
College of Business Economics
Bellingham, WA 98225

Catherine Hurley
George Washington University
315 Funger
Washington, DC 20052
email: CHURLEY@gwuvm.gwu.edu

Nathan Intrator
Brown University
Center for Neural Science
Providence, RI 02912
email: nin@brownvm.brown.edu

Orna Intrator
Brown University, Div., of Applied Math.
Box F
Providence, RI 02912
email: orna@brownvm.brown.edu

Louis A. Jaeckel
NASA Ames Research Center
Mail Stop Ellis Street
Moffett Field, CA 94035-1000
email: ljaeckel@riacs.edu

R.K. Jain
Memorial University of Newfoundland
Dept. of Mathematics and Statistics
St. John's, Newfoundland A1C 5S7
Canada
email: MATHSTAT@MUN

Sudha Jain
Jeffery Hall
Kingston, Ontario K7L 3N6
Canada
email: statistics.department@queensu.ca

David A. James
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
email: att!research!dj

Derek Janszen
Medical U. of South Carolina
171 Ashley Avenue
Charleston, SC 29425-2503

Jean Jenkins
SPSS Inc.
444 N. Michigan Ave.
Chicago, IL 60611

Robert I. Jennrich
Dept. of Math
UCLA
405 Hilgard Ave.
Los Angeles, CA 90024

Gretchen K. Jones
National Center for Health Stat.
6525 Belcrest Road, Room 915
Hyattsville, MD 20782
email: GYJ@CU.NIH.GOV

Deborah Joseph
University of Wisconsin
Comp. Sc. Dept., 1210 W. Deyton St.
Madison, WI 53706
email: joseph@cs.wisc.edu

Karen Kafadar
National Cancer Institute
CPC Biometry Branch-EPN 344
Bethesda, MD 20892
email: kk@helix.nih.gov

Ralph Kahn
Jet Propulsion Lab
MS169-237,4800 Oak Grove Dr.
Pasadena, CA 91109

Stephen Kaluzny
Stat. Sci.
1700 Westlake N., Suite 500
Seattle, WA 98109

Yana Kane-Esrig
Bellcore, 2Z-211
331 Newman Springs Rd.
Red Bank, NJ 07701-7030

Sallie Keller-McNulty
Kansas State University
Dept. of Statistics, KSU
Manhattan, KS 66506

Jon R. Kettenring
Bellcore, MRE 2Q326
445 South St.
Morristown, NJ 07962-1910
email: jon@bellcore.com

Mark J. Kiemele
United States Air Force Academy
Dept. of Mathematical Sciences
Colorado Springs, CO 80840-5701

Jim Knighton
Jet Propulsion Lab.
MS169-237,4800 Oak Grove Dr.
Pasadena, CA 91109
email: jek@huntress.jpl.nasa.gov

Alan P. Knoerr
Department of Mathematics
Occidental College
Los Angeles, CA 90041

Augustine Kong
Department of Statistics
University of Chicago
5734 University Ave.
Chicago, IL 60637

Charles Kooperberg
University of Washington
Department of Statistics, GN22
Seattle, WA 98195

Andrzej S. Kosinski
Emory University
1599 Clifton Rd, NE, Div., Biostat.
Atlanta, GA 30329

Robert Koyak
U.S. Department of Justice
555 Fourth St., NW, Rm 11-830
Washington, DC 20001

Mary K. Kuhner
University of Washington
Genetics Dept., SK-50, VW
Seattle, WA 98195
email: mkkuhner@genetics.washington.edu

An-Hsiang Kuo
SAS Institute Inc.
SAS Campus Dr.
Cary, NC 27513

Lynn Kuo
Naval Postgraduate School
Dept. of Operations Research
Monterey, CA 93943-5000
email: Bitnet:5261p@NAVPGS

Calvin Lai
University of British Columbia
Computing Services
6356 Agricultural Road
Vancouver, BC V6T 1W5
Canada

Han-Lin Lai
Washington State Dept. of Fisheries
7600 Sand Point Way NE
Seattle, WA 98115

E.S. Lander
Whitehead Institute
Room 467F
Nine Cambridge Center
Cambridge, MA 02142

Peter Lane
Statistics Dept.
Rothamsted Experimental Station
Harpenden, Herts, AL5 2JQ
England

Nicholas Lange
Brown University
Box G-A424
Providence, RI 02912
email: lange@diamond.mit.edu

Wayne Larsen
DuPont
Box 6091
Newark, DE 19714-6091

Carlos G. Lazaro
UCLA
School of Public Health, Biostat.
Los Angeles, CA 90024

Michael LeBlanc
University of Toronto
Dept., Prevent. Medicine and Biostat.
Toronto, Ontario M5S 1A8
Canada

Garrett LePage
University of Washington
11224 Meridian Ave. N. A101
Seattle, WA 98133
email: Lepage@Spock.biostat.wash.edu

Raoul D. LePage
Michigan State University
A428 Wells Hall
East Lansing, MI 48824
email: rdl@lepage-sun.stt.msu.edu

Jack C. Lee
Bellcore, MRE 2Q-374
445 South St.
Morristown, NJ 07962-1910
email: jackl@bellcore.com

Subhash Lele
The Johns Hopkins University
615 North Wolfe Street
Baltimore, MD 21205

Peter Lenk
The University of Michigan
School of Business Admin.
Ann Arbor, MI 48109-1234
email: Peter_Lenk@ub.cc.umich.edu

Ming-Ying Leung
The University of Texas
College of Sciences and Eng.
San Antonio, TX 78285-0664
email: 1MCMYL@utsa86.sa.utexas.edu

Danika Lew
University of Washington
Statistics Dept., GN-22
Seattle, WA 98195

Michael Lewis
University of Waterloo
Dept. of Statistics
Waterloo, Ontario N2L 3G1
Canada
email: milewis@watstat.waterloo.edu

Steve Lewis
Statistics Dept., GN-22
University of Washington
Seattle, WA 98195

Hongzhe Li
University of Montana
Dept. of Math. Sciences
Missoula, MT 59812-1032

Keh-Shin Lii
University of California
Dept. of Statistics
Riverside, CA 92521
email: ksl@ucrstat.ucr.edu

Mei-Hsiu Ling
Schering-Plough Co.
2000 Galloping Hill Rd., K-6-2-H7
Kenilworth, NJ 07033

Carol Link
Data Link
13375 SW Howard Drive
Tigard, OR 97223

Albert G.S. Liou
Harvard Medical School, Channing Lab
180 Longwood Ave.
Boston, MA 02115

Shwu-Fen Susan Liou
GTE Laboratories, Inc.
40 Sylvan Road
Waltham, MA 02254

Alan Lippman
University of Washington
WB -10
Seattle, WA 98195

Lon-Mu Liu
The University of Illinois
Box 4348
Chicago, IL 60680
email: U29431@UICVM on BITNET

John R. Liukkonen
Tulane University
Math Department
New Orleans, LA 70118

Michael Lloyd
Heriot-Watt University
Dept. of Actuarial Math and Stat.
Edinburgh, Scotland EH14 4AS
UK
email: mike@cara.maths.heriot-watt.ac.uk

Stephen W. Looney
Louisiana State University
3190 Ceba Lane
Baton Rouge, LA 70803-6316

Ruey-Pyng Lu
North Dakota State University
Department of Statistics
Fargo, ND 58105

C.E. Lunneborg
University of Washington
Dept. of Stat., GN-22
Seattle, WA 98195

Dr. James R. Maar
National Security Agency
9608 Basket Ring Road
Columbia, MD 21045

David Madigan
University of Washington
GN-22, Dept. of Stat.
Seattle, WA 98195

Dr. Martin Maechler
Seminar fuer Statistik, SOL F5
ETH-Zentrum
CH-8092 Zurich
SWITZERLAND
email: maechler@stat.math.ethz.ch

Colin Mallows
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

Dennis Mar
Naval Postgraduate School
Code 51 (Computer Center)
Monterey, CA 93943
email: 2001P@NAVPGS.BITNET

K.V. Mardia
University of Leeds
Dept. of Statistics
Leeds, LS2 9JT
UK
email: STA6KVM@UK.AC.Leeds.CMS1

George Marsaglia
The Florida State University
Supercomputer Comp. Res. Inst.
Tallahassee, FL 32306
email: geo.stat.fsu.edu

R. Douglas Martin
University of Washington
Dept. of Statistics, GN-22
Seattle, WA 98196

Nancy J. McDermott
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513

John Alan McDonald
University of Washington
Stat. Dept.
Seattle WA 98195

John W. McDonald
University of Southampton
Dept. of Social Statistics
Southampton, SO9 5NH
ENGLAND
email: socstats@uk.ac.soton.ibm

John D. McKenzie
Babson College
Babson Park
Wellesley, MA 02157
email: MCKENZIE@BABSON.BITNET

Steve McKinney
904 1/2 North 35th St.
Seattle, WA 98103

Barbara McKnight
University of Washington
Biostatistics, SC-32
Seattle, WA 98195
email: barb@biostat.washington.edu

Mary McLeish
University of Guelph
Dept. of Comp. and Inf. Science
Guelph, Ontario N1G 2W1
Canada

Mary Sara McPeek
U.C. Berkeley
Dept. of Stat., 367 Evans
Berkeley, CA 94720
email: mcpeek@stat.berkeley.edu

Cyrus R. Mehta
Cytel Software
137 Erie Street
Cambridge, MA 02139
email: mehta@jimmy.harvard.edu

Kent Meneghim
Washington State ESD
MS KG-11
Olympia, WA 98502

Michael M. Meyer
Carnegie Mellon University
Dept. of Statistics
Pittsburgh, PA 15213
email: mikem@stat.cmu.edu

Saira Mian
University of California
Sinsheimer Laboratory
Santa Cruz, CA 95064
email: siara@fangio.ucsc.edu

G. Arthur Mihram
Princeton
PO Box No. 1188
Princeton, NJ 08542-1188

Michael I. Miller
Washington University
Dept. of Electrical Eng., Box 1127
Seattle, WA 98124-0346

Toby J. Mitchell
Oak Ridge National Laboratory
Bldg. 6012A, PO Box 2008,MS 6367
Oak Ridge, TN 37831-6367
email: mitchell@msr.epm.ornl.gov

John F. Monahan
North Carolina State University
Box 8203, Dept. of Statistics
Raleigh, NC 27695-8203

John Moody
Yale University
PO Box 2158 Yale Station
New Haven, CT 06520

Karen L. Moore
Rothamsted Experimental Station
Harpenden, Herts, AL5 2JQ
U.K.
email: KMOORE@UK.AC.AFRC.RESA

Peter J. Munson
National Inst. of Health
Bldg. 12A, Room 2009
Bethesda, MD 20892

D. Murdock
University of Waterloo
Dept. of Statistics
Waterloo, Ontario N2L 3G1
Canada
email: dmurdock@watstat.waterloo.edu

Brian P. Murphy
University of Western Australia
Department of Management
Nedlands, Perth 6009
Western Australia

Eugene Myers
University of Arizona
Dept. Computer Science
Tucson AZ 85721
email:gene@cs.arizona.edu

Arthur Nadas
IBM T.J. Watson Research Center
Room J2-C43, P.O. Box 704
Yorktown Heights, NY 10598

Vijay Nair
AT&T Bell Laboratories
MH 2C262
Murray Hill, NJ 07974
email: vnn@research.att.com

John C. Nash
University of Ottawa
Faculty of Administration
Ottawa, Ontario K1N 6N5
Canada

Guy Nason
University of Bath
School of Mathematical Sciences
Bath, England BA2 7AY
UK
email: G.P.Nason@maths.bath.ac.uk

Padraic G. Neville
PO Box 114
Port Costa, CA 94569

Lee Newberg
UC Berkeley
2039 Francisco Street
Berkeley, CA 94709-2125

David S. Newman
Boeing Computer Services
P.O. Box 24346, MS 7L-22
Seattle, WA 98124-0346
email: dnewman@caissa.boeing.com

Ken Newman
University of Washington
3903 Woodland Park Ave. N, 202
Seattle, WA 98193
email: newman@dome.stat.washington.edu

Joe Newton
Texas A & M Univ.
Dept. of Statistics
College Station, TX 77843

Susan Ng
University of British Columbia
6356 Agricultural Road
Vancouver BC V6T 1W5, Canada

Wesley Nicholson
Pacific Northwest Lboratory
P.O. Box 999
Richland, WA 99352

Bob Obenchain
Eli Lilly & Co.
Lilly Corp. Center
Indianapolis, IN 46285

Klaus Obermayer
University of Illinois
3157 Beckman Inst.,405 N.Mathews Av.
Urbana, IL 61801
email: oby@lisboa.ks.uiuc.edu

David Olagunju
G.D. Searle
4901 Searle Pkwy.
Skokie, IL 60077

R.W. Oldford
University of Waterloo
100 University Ave. W
Waterloo, Ont. N2L 3G1
Canada

Frank Olken
Lawrence Berkeley Laboratory
1 Cyclotron Road
Berkeley, CA 94720
email: F_Olken@lbl.gov

Richard Olshen
Stanford University
HRP Building, Room 110A
Stanford, CA 94305-5092
email: olshen@playfair.stanford.edu

Panickos Palettas
Virginia Tech.
Statistics Dept., VPI & SU
Blacksburg, VA 24061
email: panickos@vtvm2.cc.vt.edu

David Patterson
University of Montana
Dept. of Mathematical Sciences
Missoula, MT 59812
email: ma_dp@selway.umt.edu

Jan Pedersen
Xerox PARC
3333 Coyote Hill Rd.
Palo Alto, CA 94304
email: pedersen@xerox.com

Shane P. Pederson
Los Alamos National Laboratory
Mail Stop F-600
Los Alamos, NM 87545
email: shane@sid.lanl.gov

Alan P. Peterson
D-S-M Labs
320 Willow St.
Walla Walla, WA 99362

Bruce Peterson
22904 NE 51st St.
Redmond, WA 98052

Jens Praestgaard
University of Washington
Dept. of Stat. GN22
Seattle, WA 98195

Daryl Pregibon
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
email: daryl@research .att.com

Haiganoush K. Preisler
Pacific Southwest Exp. Station
1960 Addison St.
Berkeley, CA 94701
email: forest@ucbstat.bitnet

Yves Quentin
Los Alamos National Laboratory
P.O. Box 1663, Mail Stop K710
Los Alamos, NM 87545

T. Ramalingam
Northern Illinois University
Mathematical Sciences Dept.
DeKalb, IL 60115
email: rama@math.nin.edu

Ernesto Ramos
Bolt Beranek & Newman
150 Cambridge Park Drive
Cambridge, MA 02140

James O. Ramsay
McGill University
1205 Dr Penfield Ave.
Montreal, Quebec PQ H3A 1B1
Canada
email: js10@mcgilla.bitnet

Rose Ray
Failure Analysis
149 Commonwealth
Menlo Park, CA 94025

William J. Raynor, Jr.
Kimberly Clark Corp.
2100 Winchester Rd.
Neenah, WI 54956

Lidia Rejtö
University of Delaware
501 Ewing Hall
Newark, DE 19716
email: rejto@chopin.udel.edu

Michael Riley
AT&T Bell Laboratories
600 Mountain Ave.
Murray Hill, NJ 07974
email: riley@research.att.com

Christian Ritter
University of Wisconsin
1220 Jenifer Str.
Madison, WI 53706
email: critter@bayes.stat.wisc.edu

Richard J. Roberts
Cold Spring Harbor Laboratory
PO Box 100
Cold Spring Harbor, NY 11724

Kathryn Roeder
Yale University
Box 2179, Yale Station
New Haven, CT 06520
email: kroeder@yalevm.ycc.yale.edu

Wilf Rosenbaum
Simon Fraser University
404-7360 Halifax street
Burnaby, British Columbia V5A 1M4
Canada
email: rosen@cs.sfu.ca

James L. Rosenberger
Penn State Univ.
Dept. of Statistics
University Park, PA 16802
email: JLR@PSUVM.PSU.EDU

Steven E. Rummel
University of Montana
Department of Math. Sciences
Missoula, MT 59812-1032

George Runger
RPI
5225 CII
Troy, NY 12180

Bert W. Rust
U.S. Department of Commerce
National Inst. of Standards and Tech.
Gaithersburg, MD 20899
email: bwr@vaxcam.nist.gov

Barbara Ryan
Stanford University
Statistics Dept., Sequoia Hall
Stanford, CA 94305

Richard S. Sacher
University of Delaware
Academic Computing Support
Newark, DE 19716
email: dsacher@brahms.ude.edu

George Sadowsky
New York University
251 Mercer Street
New York, NY 10012-1185
email: sadowsky@nyu.edu

Rami B. Safadi
Olin Research Center
350 Knotter Drive
Cheshire, CT 06410

Paul D. Sampson
University of Washington
Dept. of Statistics, GN-22
Seattle, WA 98195
email: pds@stat.washington.edu

Michael Sannella
University of Washington
311 NE Longwood Place
Seattle, WA 98115
email: sannellal@cs.washington.edu

Thomas Santner
Ohio State University
141 Cookins Hall
1958 Neil Ave.
Columbus, OH 43210

Stanley Sawyer
Washington University
Department of Mathematics
St. Louis, MO 63130
email: c31801ss@wuvmd.bitnet

Bo E.H. Saxberg
Lilly Research Labs., Stat. & Math.Sc.
Lilly Corporate Center
Indianapolis, IN 46285

Susannah B. Schiller
National Inst. of Standards & Tech.
A 337 Admin.
Gaithersburg, MD 20899

Tamar Schlick
NYU, Courant Institute
251 Mercer Street
New York, NY 10012
email: schlick@acfclu.nyu.edu

Ronald J. Schoenberg
Dyad Software
16950-151st Ave., SE
Renton, WA 98058
email: YYR@CU.NIH.GOV

Fritz Scholz
Boeing Computer Services
P.O. Box 24346
Seattle, WA 98124-0346

Nicholas Schork
University of Michigan
1444 Mason Hall
Ann Arbor, MI 48109

Eugene F. Schuster
The University of Texas
Department of Math. Sciences
El Paso, TX 79968-0514
email: gene@math.ep.utexas.edu

David Scott
Bond University
School of Inf. and Comp. Sciences
Gold Coast, Queensland 4229
Australia
email: scottd@surf.sics.bu.oz.au

John Seaman
Baylor University
P.O. Box 98005
Waco, TX 76798-8005

Ed Seymore
Weyerhaeuser and Co.
WTC 1B20
Tacoma, WA 98477

C. Frank Shen
Wyeth-Ayerst Research
CN 8000
Princeton, NJ 08543-8000

Sunil Shende
University of Nebraska
305 Ferguson Hall, Dept of CS & E
Lincoln, NE 68588-0115
email: sunil@calypso.unl.edu

David Shera
428 Broadway #3
Somerville, MA 02145

Wei-Kei Shiue
Southern Ill. University
Rm. 1333, Science Building
Edwardsville, IL 62026-1653
email: cd00@siuemus

Beatrice Shube
Chapman & Hall
166 East 34 ST -14G
New York, NY 10016

Jing Shyr
SPSS Inc.
444 N. Michigan Ave, 30th Fl.
Chicago, IL 60611

Abraham Silvers
Electric Power Research Institute
3412 Hillview Ave.
Palo Alto, CA 94304

Pippa Simpson
MCV, Virginia Commonwealth U.
Biostatistics Dept., Box 32
Richmond, VA 23298-0032
email: pmsimpson@vcuvax

Karan P. Singh
University of Alabama
900 19th St.S
Dept. Biostat & Biomath
Birmingham, AL 35294

Robert Small
Merck Sharp & Dohme Res. Labs.
West Point, PA 19486

Burton J. Smith
Tera Computer Company
400 North 34th St.
Suite 300
Seattle, WA 98103

Scott Smith
Boeing Computer Services
MS 7L-22, PO Box 24346
Seattle, WA 98124

Donald L. Snyder
Washington University
Campus Box 1127
St. Louis, MO 63130

Fei Song
University of Guelph
22 Oxford St., Apt. C
Guelph, Ontario N1H 2H3
Canada
email: fsong@snowhite.cis.uoguelph.ca

Phil Spector
UC Berkeley
367 Evans Hall, Dept. Stat.
Berkeley, CA 94720
email: spector@stat.berkeley.edu

Terence Paul Speed
University of California
Dept. of Statistics
Berkeley, CA 94720
email: terry@stat.berkeley.edu

Cathie Spino
Harvard Public Health
667 Huntington Ave.
Boston, MA 02115
email: spino@biostat.harvard.edu

James Stafford
University of Toronto
55 Snowcrest Ave.
Willowdale, Ontario M2K 2K9
Canada

Craig Stanfill
Thinking Machines
245 First Street
Cambridge, MA 02139

Kirk Steinhorst
University of Idaho
Mathematics & Statistics
Moscow, ID 83843
email: Kirk@IdUI1.bitnet

Theodor D. Sterling
Simon Fraser University
School of Computing Science
Burnaby, B.C. V5A 1S6 Canada
email: TMWl@sfu.bitnet

Robert A. Stine
Univ. of Pennsylvania
3015 Dietrich Hall
Philadelphia PA 19104

Judith S. Sunley
National Science Foundation
1800 G Street, Rm. 339
Washington, DC 20550
email: jsunley@note.nsf.gov

Alistair Sutherland
University of Strathclyde
Dept. of Stat., Richmond St.
Glasgow, 61 1NP
Great Britain

Clifton D. Sutton
George Mason University
Dept. OR & AS, 4400 University Dr.
Fairfax, VA 22030
email: csutton@gmuvax.gmu.edu

Deborah F. Swayne
Bellcore, MRE 2L-331
445 South St.
Morristown, NJ 07962-1910
email: dfs@bellcore.com

William F. Szewczyk
National Security Agency
Ft. George G. Meade, MD 20755-6000

Evangelos Tabakis
MIT
390 Riverway, Apt.18
Boston, MA 02115
email: taba@math.mit.edu

Martin Tanner
Department of Statistics
University of Rochester
Box 630, 601 Elmwood Ave.
Rochester, NY 16802
email: tanm@seneca.bst.rochester.edu

Michael Tarter
UC. Berkeley
32 Warren Hall
Berkeley, CA 94720

George Tauchen
Duke University
305 Social Science Building
Durham, NC 27706

Simon Tavaré
University of Southern California
Los Angeles, CA 90089-1113
email: tavare@mtha.usc.edu

Charles C. Taylor
University of Leeds
Department of Statistics
Leeds, England LS2 9JT UK
email: STA6CCT@cms1.ucs.leeds.ac.uk

Leslie A. Taylor
UCSF Computer Graphics Lab
Rm 926, Box 0446, 513 Parnassus
San Francisco, CA 94143
email: ltaylor@cal.ucsf.edu

Robert F. Teitel
Abt Associates Inc.
4800 Montgomery Lane
Bethesda, MD 20814

George R. Terrell
Virginia Polytechnic Inst.
Department of Statistics
Blacksburg, VA 24061-0439
email: terrell@vtvm2.cc.vt.edu

Terry M. Therneau
Mayo Clinic
200 1st St., SW, Gugg.10
Rochester, MN 55905

Yves Thibaudeau
U.S. Bureau of the Census
Federal Building 4, Room 3000
Washington, DC 20233

H. Mathis Thoma
Ciba-Geigy Corp.
556 Morris Ave.
Summit, NJ 07901

Ronald Thomas
VA Medical Center 151-K
3801 Miranda Ave.
Palo Alto, CA 94304

William Thomas
University of Minnesota
420 Delaware SE,A-460 Mayo,Box 197
Minneapolis, MN 55455
email: will@muskie.biostat.umn.edu

E.A. Thompson
University of Washington
Dept. Statistics, 9N-22
Seattle, WA 98195
email: thompson@stat.washington.edu

G.L. Thompson
Southern Methodist University
Dept. of Statistical Science
Dallas, TX 75275

Steve Thomson
University of Kentucky Comp. Center
128 McVey Hall
Lexington, KY 257-2259

Jeff Thorne
University of Washington
Dept. of Genetics, SK-50
Seattle, WA 98195
email:jeff@evolution.genetics.washington.edu

Anthony D. Thrall
Electric Power Research Institute
PO Box 10412
Palo Alto, CA 94303
email: ea250tt@epri.BITNET

William J. Threlfall
British Columbia Cancer Agency
600 West 10th Ave.
Vancouver, B.C.
Canada

R. Tibshirani
University of Toronto
Dept. of Prev. Med. & Biostats.
Toronto, Ont. M5S 1A8
Canada

Camlin Tierney
University of Washington
SC-32, Dept. of Biostatistics
Seattle, WA 98195
email: camlin@biostat.washington.edu

Luke Tierney
University of Minnesota
School of Statistics
Minneapolis, MN 55455
email: luke@umnstat.stat.umn.edu

Diane Tipping
Rhone-Poulenc Rorer Central Res.
568 Pioneer Circle
Harleysville, PA 19438

Robert Tipping
568 Pioneer Circle
Harleysville, PA 19438

Randy Tobias
SAS Institute
SAS Circle, Box 8000
Cary, NC 27513

Mitch Toland
Weyerhaeuser and Co.
WTC 1B20
Tacoma, WA 98477

David Tritchler
Ontario Cancer Institute
500 Sherbourne St.
Toronto, Ontario M4X 1K9
Canada
email: TRITCHLE%UTOROCI.bitnet

Simon Tse
AT&T Bell Laboratories
Rm. TD-509, 600 Mountain Ave.
Murray Hill, NJ 07974

Robert Tsou
TRW Marketing Services
City Parkway West, Suite 1000
Orange, CA 92668

John R. Tucker
National Academy of Sciences
RM. NAS 312, 2101 Constitution Ave.,NW
Washington, DC 20418
email: JTUCKER@NAS.BITNET

Paul A. Tukey
Bellcore, MRE 2M-391
445 South St.
Morristown, NJ 07962-1910
email: paul@bellcore.com

N. Scott Urquhart
Oregon State Univ.
Statistics, Kidder Hall #44
Corvallis OR 97331-4606

Pirooz Vakili
Boston University
44 Cummington St., Man. Eng.
Boston, MA 02215
email: vakili@buenga.bu.edu

Alessio Valentini
Universita Tuscia
Via de Lellis
Viterbo, 01100 Italy
email: tusciazo@icnucevm.bitnet

Scott VanderWiel
AT&T Bell Laboratories
2C-277, 600 Mountain Ave.
Murray Hill, NJ 07974
email: scottyv@research.att.com

Mark C. van Pul
Centre for Math. and Comp. Sc.
P.O. Box 4079, 1009 AB Amsterdam
The Netherlands
email: mark@cwi.nl

John C. Varady
Syntex Labs
3401 Hillview Ave. L02500
Palo Alto, CA 94303

Alex D. Varshavsky
Eli Lilly and Company
Lilly Corporate Center
Indianapolis, IN 46285

Dominic F. Vecchia
NIST
Mail Code 882.01, 325 Broadway
Boulder, CO 80303-3328

Achilles Venetoulias
Bank of America 5887
555 California St., 13th Fl.
San Francisco, CA 94104
email: axilleas@playfair.stanford.edu

Silvia C. Vega
University of Washington
Statistical Sciences
204 NW Bowdoin Rd.
Seattle, WA 98107

Rakesh Vohra
Ohio State University
1775 College Rd.
Columbus, OH 43210

Chaiho C. Wang
U.S. Department of Justice
555 Fourth Street, Rm. 11811
Washington, DC 20001

Ping Wang
University of Georgia
100 Rogers Rd. M104
Athens, GA 30605
email: cmsja18@jga.bitnet

R.H. Wang
Olin Research Center
350 Knotter Drive, P.O. Box 586
Cheshire, CT 06410-0586

Tandy Warnow
University of California
2931 Dwight Way, 21D
Berkeley, CA 94720
email: tandy@ernie.berkeley.edu

Michael Waterman
Univ. of Southern California
Math Dept.
Los Angeles, CA 90089-1113
email: msw@msw.usc.edu

Barbara Weeks
Ciba Geigy
556 Morris Ave.
Summit, NJ 07901

Daniel E. Weeks
University of Pittsburgh
130 DeSoto St., A300 Crabtree Hall
Pittsburgh, PA 15261
email: WEEKS@VMS.CIS.PITT.EDU

Edward J. Wegman
George Mason University
242 Science-Tech. Bldg.
Fairfax, VA 22030
email: EWEGMAN@GMUVAX.GMU.EDU

Andreas Weigend
Stanford University
Building 420, Jordan Hall
Stanford, CA 94305
email: andreas@psych.stanford.edu

Norris Weimer
University of Alberta
University Computing Systems
Edmonton, Alberta T6G 2H1
Canada
email: USERNWHY@MTS.VCS.UALBERTA.CA

James J. Weinkam
Simon Fraser University
School of Computing Science
Burnaby, B.C. V5A 1S6
Canada
email: JJW1@cc.sfu.ca

Gunter M.T. Weiss
University of Western Ontario
Room 3005, EMS BLdg., Stats.
London, Ontario N6A 5B9
Canada

Larry Weldon
Simon Fraser University
Burnaby, B.C. V5A 1S6
Canada

T.S. Weng
FDA,CDRH,OST,Div.,Biometric Sc.
12908 Missionwood Way
Potomac, MD 20854

Robert Wesley
Clinical Center, Nat. Inst. Health
9807 Owen Brown Road
Columbia, MD 21045

Robert Wilkinson
The Lubrizol Corporation
29400 Lakeland Boulevard
Wickliffe, OH 44092

William E. Winkler
Bureau of the Census
Room 3000-4
Washington, DC 20233

Augustine Wong
University of Waterloo
Dept. of Stat and Act. Sci.
Waterloo, Ontario N2L 3G1
Canada
email: cmwong@watstat.waterloo.edu

Frank Wright
Univ. of Edinburgh, Scot. Agric.
J.C.M.B., Rm 3610, Kings Buildings
Edinburgh, EH9 3JZ
United Kingdom
email: frank@sass.sari.ac.uk

Shu-Chen H. Wu
RW Johnson Pharm. Research
4245 Sorrento Valley Blvd.
San Diego, CA 92121

Trong Wu
Southern Illinois University
Rm. 1339, Science Building
Edwardsville, IL 62026-1653

Momiao Xiong
University of Georgia
Statistics Department
Athens, GA 30602

Chong-wei Xu
Southern Illinois Univ.
Dept. of Computer Science
Edwardsville, IL 62026-1653

Brian S. Yandell
University of Wisconsin
Horticulture and Statistics Dept.
Madison, WI 53706
email: yandell@stat.wisc.edu

Grace L. Yang
University of Maryland
Dept. of Mathematics
College Park, MD 20742
email: gyang@umd2.umd.edu

Mark C.K. Yang
University of Florida
Department of Statistics
Gainesville, FL 32611-2049

Demirhan Yenigun
AT&T Bell Labs
600 Mountain Ave., 7B-511A
Murray Hill, NJ 07974

Jeremy York
Statistics Dept., GN-22
University of Washington
Seattle, WA 98195
email: jeremy@lisbon.stat.washington.edu

Dean M. Young
Baylor University
P.O. Box 98005
Waco, TX 76798-8005

Forrest W. Young
University of North Carolina
Psychometrics, CB#3270 Davie
Chapel Hill, NC 27599-3270
email:ULURU@UNC.BITNET

Paul Young
University of Washington
Computer Sc. & Eng. FR-35
Seattle, WA 98195
email: young@cs.washington.edu

Stan Young
Glaxo Inc.
5 Moore Drive
Research Triangle Park, NC 27709

Gholam-Ali Zakeri
California State University
18111 Nordhoff Street - MATH
Northridge, CA 91330

David Zeitler
4141 Eastern Avenue, S.E.
Grand Rapids, MI 49518-8727
email: zeitler@si.com

Peter J. Zemroch
Shell Research LTD
Thornton Research Centre, PO Box 1
Chester, CH1 3SH, U.K.

Guangrui Ray Zhu
Schering-Plough Research
200 Galloping Hill Rd, K6-2H-18
Kenilworth, NJ 07033
email: zgr@playfair.stanford.edu

Alan R. Zinsmeister
Mayo Clinic
7th Floor, Harwick Bldg.
Rochester, MN 55905

INDEX OF AUTHORS