

AD-A111 893

MICHIGAN TECHNOLOGICAL UNIV HOUGHTON

F/S 17/9

STATISTICAL PATTERN RECOGNITION TECHNIQUES AS APPLIED TO RADAR --ETC(U)

DEC 81 W A FORDON; A A FRASER

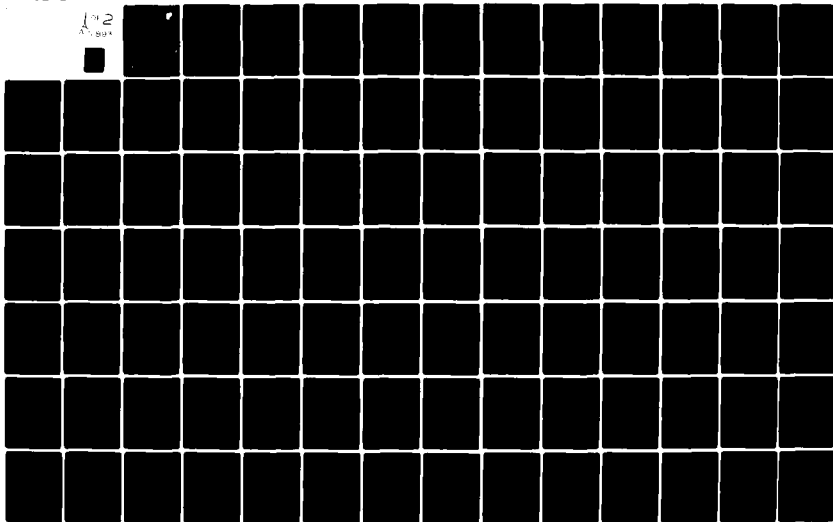
F30602-78-C-0102

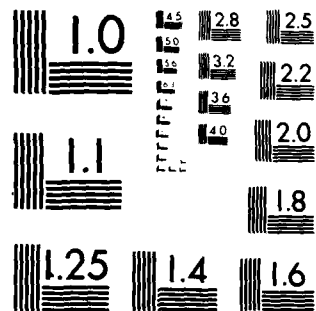
UNCLASSIFIED

RADC-YR-81-61

ML

12
A-1004





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A111893

RADC-TR-81-61
Final Technical Report
December 1981



STATISTICAL PATTERN RECOGNITION TECHNIQUES AS APPLIED TO RADAR RETURNS

Michigan Technological University

W. A. Fordon
A. A. Fraser

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC
EXTRACTED
MAR 11 1982
D
H

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

"Original contains color
plates: All DTIC reproduct-
ions will be in black and
white"

03 10 014

DTIC FILE COPY

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-81-61 has been reviewed and is approved for publication.

APPROVED:

William L. Simkins

WILLIAM L. SIMKINS, Jr.
Project Engineer

APPROVED:

Frank J. Rehm

FRANK J. REHM
Technical Director
Surveillance Division

FOR THE COMMANDER:

John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC.(OCTS) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-81-61	2. GOVT ACCESSION NO. AD-A111893	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) STATISTICAL PATTERN RECOGNITION TECHNIQUES AS APPLIED TO RADAR RETURNS		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report Apr 79 - Sep 79
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) W. A. Fordon A. A. Fraser		8. CONTRACT OR GRANT NUMBER(s) F30602-78-C-0102
9. PERFORMING ORGANIZATION NAME AND ADDRESS Michigan Technological University Houghton MI 49931		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 450611PF
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (OCTS) Griffiss AFB NY 13441		12. REPORT DATE December 1981
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		13. NUMBER OF PAGES 178
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: William L. Simkins (OCTS)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Pattern Recognition RADAR Ground Clutter		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report presents a summary of the basic principles of pattern recognition and statistical decision theory and applies them to the problem of classifying radar returns. While pattern recognition techniques have been applied to radar signal detection problems, they have rarely been used in testing hypothesis for classifying radar returns. Two techniques, the parametric Bayes and the non-parametric K-Nearest Neighbor algorithms, were compared using simulated radar backscatter		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

data. The error rate of these algorithms was the chief criterion used for the evaluation of performance. The results showed that the Nearest Neighbor technique gives a smaller error rate than the Bayes technique for the limited data sets tested.



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Diss	

A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Statistical Pattern Recognition Techniques
as Applied to Radar Returns

	<u>Page</u>
1.0 Introduction	1
2.0 Fundamentals of Statistical Decision Theory	2
2.1 Detection	2
2.2 Parameter Estimation	5
2.3 Loss Functions	6
2.4 Binary Detection	9
2.5 Bayes' Decision Rule	12
2.6 Error Probabilities	15
2.7 The Neyman-Pearson Criterion	17
2.8 The Minimax Approach	19
2.9 Bayes' Solutions for Complex Cost Functions	22
2.10 Preferred Neyman-Pearson Strategy	24
2.11 Intuitive Substitute	24
2.12 Fixed and Sequential Testing	26
2.13 Concluding Remarks	27
3.0 Parameter Estimation and Supervised Learning	34
3.1 General Bayesian Learning	35
4.0 Unsupervised Learning and Clustering	40
4.1 Mixture Densities and Identifiability	40
4.2 Clustering	41
4.2.1 Clustering Methodology	42

	<u>Page</u>
4.2.1.1 Squared-Error Clustering Algorithms	42
4.2.1.2 Hierarchical Clustering	45
4.2.1.3 Graph-Theoretic Methods	45
5.0 Testing Methods	48
6.0 Tests With Simulated Radar Data	53
7.0 Tests With Actual Radar Data	56
8.0 Summary, Conclusions, and Recommendations	57
Glossary of Terms	59
Bibliography	62
Appendix A Bayes Classifier Program	64
Appendix B FORGY/Jancey Program - Squared-Error Clustering	70
Appendix C Hierarchical Clustering Program	76
Appendix D Minimal Spanning Tree Program - Graph-Theoretic Method	79
Appendix E "The Comparison of a Bayesian Classifier and a k-Nearest Neighbor Statistical Pattern Recognition Technique as Applied to Radar Ground Clutter," M.S. Thesis - A. A. Fraser	82

EVALUATION

The increasing trend towards automated radar systems and "intelligent" signal processing requires the sensor to treat the environmental scatter as information as well as "clutter" or interference. By inference from measurable quantities and statistics, the processor will recognize the existence of weather, chaff, discrete targets, statistically defined "homogeneous" areas, shadowing as opposed to specular reflection, and other environmental categories. This information will allow the system to adapt its waveform, energy budget, detection/CFAR and tracking algorithms for optimum performance. Unfortunately, while some clutter parameters can be modeled as deterministic or as simple random variables with excellent results, many observable characteristics appear to be nonstationary, time-varying, or otherwise ill-defined. The development of "intelligent" autonomous sensors requires an improved approach for analyzing and testing large data sets in support of modeling these unknown quantities.

This post-doctoral effort presents a summary of pattern recognition and statistical decision theory and stresses the strengths, weaknesses, and peculiarities of parametric and nonparametric algorithms. The effort provides valuable insight into the robustness and limitations of several algorithms and emphasizes the care required in using these techniques for data analysis. This effort supports the Air Force requirements as defined in TPO 4A.

William L. Simkins, Jr.
WILLIAM L. SIMKINS, JR.
Project Engineer

1.0 Introduction

The application of pattern recognition techniques to radar problems has been applied previously to signal detection problems. The basic theories of statistical hypothesis testing and decision theory apply.

This paper is a summary of the basic principles of pattern recognition and statistical decision theory. The effort has been to produce a brief exposition of the theory and terminology, with sufficient rigor to allow an understanding of the fundamentals. Emphasis has been to select references for their lucidity and tie them together to illuminate understanding.

The second section of the paper deals with the fundamentals of statistical decision theory. It can be seen from this exposition that the terminology applied to radar detection is quite similar, if not identical to, pattern recognition terminology. To this end, sections three and four deal with supervised and unsupervised learning respectively. The fifth section contains a discussion of testing methods, and the sixth is a summary of test results on simulated radar data. The seventh section is a brief discussion of some results using actual radar data, while section eight contains conclusions and recommendations. A glossary and bibliography are appended, together with information about the computer programs used to implement the various algorithms discussed in the report.

2.0 Fundamentals of Statistical Decision Theory

The objectives of a radar system are to: (a) detect the presence of objects in clutter and noise, and (b) estimate their positions and motions in space relative to the radar. These objectives can be studied in terms of the discipline known as 'statistical decision theory'. Reference (1) has an excellent discussion of statistical decision theory as it applies to radar problems. The following treatment is taken from Chapter 8 of reference (1).

2.1 Detection

A radar echo is generally immersed in some form of additive noise, and also usually in clutter return. Since noise and clutter are random phenomena, a decision must be made, (statistical in nature), which concerns the presence or absence of a target echo. We would like to minimize the number of incorrect decisions. Consequently, if we have a priori information concerning the echo signal, noise, and clutter we can take advantage of this in making our decisions.

As a problem in hypothesis testing, the detection of a signal in noise can be seen as making a decision with regard to a finite-duration sample of a noisy waveform. This sample may or may not contain a signal. Thus, the hypothesis that the received waveform does not contain a signal is to be tested against the hypothesis that the waveform does contain a signal. The first hypothesis, denoted by H_0 , is often called the 'null' hypothesis. The second hypothesis, denoted by H_1 is referred to as the 'alternative' hypothesis. If the signal to be detected is deterministic (i. e., its structure is completely known) then H_1 is called a 'simple' alternative. In radar this situation

almost never occurs, since echo amplitude and phase are usually unknown. When the signal to be detected is a member of a finite or infinite set of signals, H_1 is true, we can conclude only that one member of the signal class is present whose identity is not revealed by the test.

Let us represent the class of possible signals, (echoes), as vector points $\bar{s}$ in signal space Ω . Each point in the space represents a waveform with a particular combination of signal parameter values such as amplitude, phase, doppler, etc. When possible, a probability of occurrence is assigned to each combination of signal parameters. This information is contained in a joint a priori probability density function $\sigma(\bar{s})$ over all the points $\bar{s}$ in signal space Ω .

In a similar way noise and clutter spaces can be defined whose points $\bar{n}$ describe all possible waveform realizations of the noise and clutter process within the observation interval. From the statistical and spectral properties of the noise and clutter, an a priori joint probability density $p(\bar{n})$ can be deduced that describes the frequency of occurrence of waveforms in this space.

Next, an observation space, Γ , is defined whose points $\bar{v}$ represent all possible joint combinations of signal and noise waveforms within the observation interval. The frequency of occurrence of members of this space can also be described by an a priori probability density function which is written as a conditional probability $p(\bar{v} | \bar{s})$ to show the explicit dependence of the observed waveform $\bar{v}$ in the signal $\bar{s}$. For convenience, we include the null hypothesis $\bar{s} = \bar{0}$ as a point in signal space Ω .

An essential feature of the theory is the decision rule by which a decision is made. This rule depends only on the observed waveform $\bar{v}$ and not on the signal $\bar{s}$. A decision rule leading to a decision 'd' as a result of the observation $\bar{v}$ is denoted by $D(d | \bar{v})$. $D(d | \bar{v})$ describes the conditional probability of deciding d having observed $\bar{v}$. Thus, for a particular waveform $\bar{v}$ there is only a probability that a decision $d_1 =$ "yes" or $d_2 =$ "no" will be made. Such a decision rule is called a 'randomized decision rule' and its implementation requires a chance mechanism as part of the receiver structure. In practical applications, the decision rule has usually reduced to a 'nonrandom decision rule' where a probability of 0 or 1 is assigned to $D(d_1 | \bar{v})$ and $D(d_2 | \bar{v})$ for each observation $\bar{v}$. In this case the receiver does not require a chance mechanism.

The set of possible decisions $\bar{d}$ in a statistical decision problem can be described as points in a decision space Δ . If the interpretation of a decision rule $D(\bar{d} | \bar{v})$ as a probability (or probability density if a continuum of possible decisions is considered) is retained, then $D(\bar{d} | \bar{v})$ describes the probability (density) of each point in decision space for every possible waveform $\bar{v}$. In a signal detection problem, decision space for every possible waveform $\bar{v}$. In a signal detection problem, decision space contains only two points: signal present and signal absent.

Figure 1 shows the general decision problem in terms of the various spaces previously defined. A decision rule may be interpreted as an operation that maps points in observation space into points in decision space with a preassigned probability $D(\bar{d} | \bar{v})$. The essence of the decision

problem is to choose decision rules that accomplish this mapping in an optimum way with respect to a chosen criterion of performance. The mathematical operations embodied in the decision rule define the operations performed by an "optimum" decision receiver on the received waveform $\bar{v}$ in order to render a decision $\bar{d}$ in accordance with the selected criterion.

2.2 Parameter Estimation

Some attributes of a radar target can be deduced from modifications of the reflected radar waveform. These modifications are conveniently characterized by unknown signal parameters of an otherwise deterministic echo signal structure. Theoretically, were it not for the presence of noise, the values of these parameters could be measured to any desired degree of precision.

Parameter estimation is formulated as a problem in statistical decision theory by an extension of the concept of radar detection. In detection, observation space Γ is mapped into two points in decision space Δ by means of decision rule $D(\bar{d} | \bar{v})$, namely signal present and signal absent. If decision space Δ is enlarged to include a selected subset of points in signal space Ω , Figure 1 shows the parameter estimation problem in terms of decision theory. In fact, the set of points in decision space may contain the entire set of points in decision space. In this case the dimensionalities of signal space and decision space are identical. Often, however, less precision is required, in which case the dimensionality of decision space is smaller than that of signal space. Figure 2 illustrates two possible situations - one in which the dimensionalities of signal and decision space are the same, and the

other, shown by dashed lines, in which the dimensionality of decision space is less than that of signal space. A similar situation exists when signal space is of infinite dimension.

In summary, parameter estimation divides observation space into subsets of points that are mapped by a decision rule into signal points in decision space Δ . Thus, decision d_i is assigned to observed waveform $\bar{v}$ in accordance with decision rule $D(d_i | \bar{v})$, when $\bar{v}$ is a member of the i th subset of points in Γ . As before, the optimum decision rule is determined by the selected optimality criterion.

Since in both detector and parameter estimation the decision rule maps the space of observations Γ into the space of decisions Δ , the simple detection problem is seen to be merely a special case of the parameter estimation problem where all the points in decision space corresponding to signal present ($\bar{s} \neq \bar{0}$) are grouped together. It should be noted, however, that a decision receiver that is optimum for parameter estimation may not be optimum for detection. Thus, it is necessary to treat detection and parameter estimation as separate statistical decision problems.

2.3 Loss Functions

In order to select an optimum decision rule in a statistical decision problem, we evaluate the relative performance of each possible decision rule, selecting the rule that yields the "best" performance. This means that a method of evaluating performance is required. The concept of a simple 'cost' or 'loss function', which associates a quantitative cost $C(\bar{s}, \bar{d})$ with each point $\bar{s}$ in signal space Ω and each point $\bar{d}$ in decision

space Δ was introduced by Wald (2). The cost function describes the loss incurred by a receiving system that results in a decision $\bar{d}$ when the input signal is $\bar{s}$. In the case of a correct decision, the loss in cost function may be interpreted as a gain.

A substantial theory has been developed for problems in which 'average loss' is used as a measure of comparative system performance. This choice is motivated by the fact that average loss is representative of system performance evaluated over all possible modes of behavior. A decision rule that describes a receiver with the least average loss is called a 'Bayes rule', and the receiver a 'Bayes receiver'. Other performance criteria lead to different decision rules (e. g. , minimax, Neyman-Pearson).

It is convenient to define two loss functions. The 'conditional loss' $L_c(D | \bar{s})$ is a useful measure of loss when the input signal is known, or when the input signal is not known and the a priori probability density $\sigma(\bar{s})$ over signal space Ω is also unknown. If the a priori probability density $\sigma(\bar{s})$ is known, a more complete performance loss rating is provided by the 'average loss' $L(D, \sigma)$.

The conditional loss $L_c(D | \bar{s})$ is defined as the mathematical expectation of the loss with respect to all possible decisions $\bar{d}$ for a given $\bar{s}$ and decision rule D . Thus,

$$(1a) \quad L_c(D | \bar{s}) = E_{\bar{d} | \bar{s}} [C(\bar{s}, \bar{d})]$$

$$(1b) \quad = \int_{\Delta} C(\bar{s}, \bar{d}) p(\bar{d} | \bar{s}) d\bar{d}.$$

Equation (1) states that the conditional loss is the sum of costs associated with all possible decisions weighted by their probability of occurrence, assuming that $\bar{s}$ is the true state of nature. The conditional probability of deciding $\bar{d}$ given $\bar{s}$, $p(\bar{d}|\bar{s})$, can also be expressed by:

$$(2) \quad p(\bar{d}|\bar{s}) = \int_{\Gamma} p(\bar{d}, \bar{v}|\bar{s}) d\bar{v}.$$

The form of equation (2) indicates that $p(\bar{d}|\bar{s})$ can be considered a (conditional) marginal density function that can be derived from the (conditional) joint probability density function $p(\bar{d}, \bar{v}|\bar{s})$. By means of the chain rule for conditional probabilities:

(3) $p(\bar{d}, \bar{v}|\bar{s}) = D(\bar{d}|\bar{v}, \bar{s})p(\bar{v}|\bar{s})$. Thus, equation (2) can be expressed as:

$$(4a) \quad p(\bar{d}|\bar{s}) = \int_{\Gamma} D(\bar{d}|\bar{v}, \bar{s})p(\bar{v}|\bar{s})d\bar{v}$$

$$(4b) \quad = \int_{\Gamma} D(\bar{d}|\bar{v})p(\bar{v}|\bar{s})d\bar{v}.$$

Equation (4b) used the fact:

(5) $D(\bar{d}|\bar{v}, \bar{s}) = D(\bar{d}|\bar{v})$, since the decision rule $D(\bar{d}|\bar{v})$ is only a function of the waveform $\bar{v}$, as previously discussed, and is therefore dependent of $\bar{s}$. Inserting (4b) into (1) results in:

$$(6) \quad L_c(D|\bar{s}) = \int_{\Gamma} p(\bar{v}|\bar{s})d\bar{v} \int_{\Delta} C(\bar{s}, \bar{d})D(\bar{d}|\bar{v}) d\bar{d}.$$

When the input signal is not known, but the a priori probability density function $\Sigma(\bar{s})$ is known the average loss $L(D, \Sigma)$ is defined as the mathematical expectation of the conditional loss with respect to the input signal statistics. Thus:

$$(7a) \quad L(D, \sigma) = E_{\bar{s}} [L_c(D | \bar{s})]$$

$$(7b) \quad = \int_{\Omega} \sigma(\bar{s}) d\bar{s} \int_{\Gamma} p(\bar{v} | \bar{s}) d\bar{v} \int_{\Delta} C(\bar{s}, \bar{d}) D(\bar{d} | \bar{v}) d\bar{d}.$$

Alternatively, the average loss can be defined as the sum of costs associated with decisions $\bar{d}$ and inputs $\bar{s}$ weighted according to their joint probability of occurrence. Thus,

$$(8a) \quad L(D, \sigma) = E_{\bar{d}, \bar{s}} [C(\bar{s}, \bar{d})]$$

$$(8b) \quad = \int_{\Omega} \int_{\Delta} C(\bar{s}, \bar{d}) p(\bar{d}, \bar{s}) d\bar{d} d\bar{s}$$

$$(8c) \quad = \int_{\Omega} \sigma(\bar{s}) d\bar{s} \int_{\Delta} C(\bar{s}, \bar{d}) p(\bar{d} | \bar{s}) d\bar{d}.$$

The inner integral in (8c) is the conditional loss defined in (1b). Therefore, $L(D, \sigma)$ can also be written as:

$$(9) \quad L(D, \sigma) = \int_{\Omega} L_c(D | \bar{s}) \sigma(\bar{s}) d\bar{s}, \text{ which is a restatement of (7a).}$$

In summary, the average loss function $L(D, \sigma)$ provides a measure for evaluating the performance of different systems when complete a priori statistics concerning the signal and noise are available. We will next examine the binary detection problem.

2.4 Binary Detection

Binary detection involves making a decision between two possible outcomes: Noise alone or signal plus noise.

Let H_0 denote the hypothesis (null) that noise alone is present, and H_1 the composite alternative hypothesis that signal plus noise is present. Thus:

$$(10) \quad H_0: \bar{s} \in \Omega_0$$

$H_1: \bar{s} \in \Omega_1$, where Ω_0 and Ω_1 are nonoverlapping regions of signal space. It therefore follows from (1) that Ω_0 contains the single point $\bar{s} = \bar{0}$, and Ω_1 contains all points, $\bar{s} \neq \bar{0}$.

We can find an expression for the a priori probability density $\sigma(\bar{s})$ defined over signal space as follows. Let P and Q be the a priori probabilities of signal present and signal absent, respectively. Then:

$$(11) \quad \sigma(\bar{s}) = Q \delta(\bar{s} - \bar{0}) + P \omega(\bar{s}), \text{ where the Dirac delta function}$$

$\delta(\bar{s} - \bar{0})$ describes the discrete probability distribution of $\bar{s}$ over Ω_0 and $\omega(\bar{s})$ describes the probability density of $\bar{s}$ over space Ω_1 . We see that:

$$(12) \quad \int_{\Omega_1} \omega(\bar{s}) d\bar{s} = 1. \text{ When (11) is substituted into (7b), the expres-}$$

sion for average loss $L(D, \sigma)$ may be rewritten as:

$$(13) \quad L(D, \sigma) = Q \int_{\Gamma} p(\bar{v} | \bar{0}) d\bar{v} \int_{\Delta} C(\bar{0}, \bar{d}) D(\bar{d} | \bar{v}) d\bar{d} \\ + P \int_{\Omega_1} \omega(\bar{s}) d\bar{s} \int_{\Gamma} p(\bar{v} | \bar{s}) d\bar{v} \int_{\Delta} C(\bar{s}, \bar{d}) D(\bar{d} | \bar{v}) d\bar{d}.$$

Equation (13) can be simplified with the definition:

$$(14) \quad E_{\bar{s}} [p(\bar{v} | \bar{s})] = \overline{p(\bar{v} | \bar{s})}_{\bar{s}} = \int_{\Omega_1} \omega(\bar{s}) p(\bar{v} | \bar{s}) d\bar{s}, \text{ to give:}$$

$$(15) \quad L(D, \sigma) = Q \int_{\Gamma} p(\bar{v} | \bar{0}) d\bar{v} \int_{\Delta} C(\bar{0}, \bar{d}) D(\bar{d} | \bar{v}) d\bar{d} \\ + P \int_{\Gamma} p(\bar{v} | \bar{s}) d\bar{v} \int_{\Delta} C(\bar{s}, \bar{d}) D(\bar{d} | \bar{v}) d\bar{d}.$$

Let cost assignments $C(\bar{s}, \bar{d})$ and $C(\bar{0}, \bar{d})$ be made as shown in Table 1, where C_{α} and C_{β} denote costs of errors. C_{α} is the penalty or cost associated with deciding signal is present when, in fact, there is no signal. C_{β} is the cost associated with deciding no signal when, in fact, there is a signal present. The notation reflects the fact that α is the false-alarm probability, and β is the average missed-detection probability. The quantities $C_{1-\alpha}$ and $C_{1-\beta}$ represent the costs of correct decisions - that is:

$$(16a) \quad C_{1-\alpha} = C(\bar{s} \in \Omega_0, d_0)$$

(16b) $C_{1-\beta} = C(\bar{s} \in \Omega_1, d_1)$. These costs can be carried through the remaining derivations. However, since no penalty is usually associated with correct decisions, it is convenient to set the cost of correct decisions to 0:

$$(17) \quad C_{1-\alpha} = C_{1-\beta} = 0.$$

Table 1 - Cost Matrix for Binary Detection

		signal $\bar{s}$	
		$\bar{s} = 0$	$\bar{s} \neq 0$
Decision $\bar{d}$	d_0	$C_{1-\alpha}$	C_{β}
	d_1	C_{α}	$C_{1-\beta}$

Substituting the cost matrix of Table 1 and equation (17) into (15)

gives:

$$(18) \quad L(D, \sigma) = QC_{\alpha} \int_{\Gamma} D(d_1 | \bar{v}) p(\bar{v} | \bar{0}) d\bar{v} \\ + PC_{\bar{\beta}} \int_{\Gamma} D(d_0 | \bar{v}) p(\bar{v} | \bar{s})_{\bar{s}} d\bar{v}.$$

Equation (18) can be written in another form. If α denotes the probability of deciding a signal is present when there is no signal (Type I error or false alarm), and $\bar{\beta}$ denotes the probability of deciding that signal is absent when it is really present (Type II error or missed detection):

$$(19) \quad \alpha = \int_{\Gamma} p(\bar{v} | \bar{0}) D(d_1 | \bar{v}) d\bar{v}$$

$$(20) \quad \bar{\beta} = \frac{\int_{\Gamma} p(\bar{v} | \bar{s})_{\bar{s}} D(d_0 | \bar{v}) d\bar{v}}{\int_{\Gamma} p(\bar{v} | \bar{s})_{\bar{s}} D(d_0 | \bar{v}) d\bar{v} / \bar{s}} \\ = \bar{\beta}(\bar{s})_{\bar{s}}. \text{ Note that } \bar{\beta} \text{ by definition is the Type II error}$$

probability averaged with respect to the a priori distribution of signal.

Substituting (19) and (20) into (18) gives:

$$(21) \quad L(D, \sigma) = Q \alpha C_{\alpha} + P \bar{\beta} C_{\bar{\beta}}. \text{ Equation (21) relates average loss } L(D, \sigma) \text{ to the a priori probability of signal } P = 1 - Q, \text{ the probabilities of Type I and Type II errors, } \alpha \text{ and } \bar{\beta}, \text{ and the costs of Type I and Type II errors, } C_{\alpha} \text{ and } C_{\bar{\beta}}, \text{ respectively.}$$

2.5 Bayes' Decision Rule

Bayes' decision rule D_B results from the minimization of $L(D, \sigma)$. Since binary decision space Δ contains only the two points d_0 (no signal) and d_1 (signal plus noise), decision rule $D_B(\bar{d} | \bar{v})$ satisfies the relation:

$$(22) \quad D_B(d_0 | \bar{v}) + D_B(d_1 | \bar{v}) = 1. \text{ Substituting (22) into (18) and eliminating } D_B(d_1 | \bar{v}) \text{ yields:}$$

$$(23) L(D, \sigma) = Q C_{\alpha} + \int_{\Gamma} D_B(d_0 | \bar{v}) [P C_{\beta} \overline{p(\bar{v} | \bar{s})} - Q C_{\alpha} p(\bar{v} | \bar{0})] d \bar{v}$$

Note that $D_B(d_0 | \bar{v})$ is positive and less than unity. Also $P, Q, C_{\alpha}, C_{\beta}$ are positive quantities. Then, to minimize $L(D, \sigma)$ choose:

$$(24a) D_B(d_0 | \bar{v}) = 1$$

$$(24b) D_B(d_1 | \bar{v}) = 0 \text{ that is, decide signal is absent when}$$

$$(25) P C_{\beta} \overline{p(\bar{v} | \bar{s})} < Q C_{\alpha} p(\bar{v} | \bar{0}), \text{ and choose}$$

$$(26a) D_B(d_0 | \bar{v}) = 0$$

$$(26b) D_B(d_1 | \bar{v}) = 1 \text{ that is, decide signal is present when:}$$

$$(27) P C_{\beta} \overline{p(\bar{v} | \bar{s})} \geq Q C_{\alpha} p(\bar{v} | \bar{0})$$

Inequalities (25) and (27) can be rewritten in terms of a function $\ell(\bar{v})$, called the 'generalized likelihood ratio':

$$(28) \ell(\bar{v}) = \frac{P \overline{p(\bar{v} | \bar{s})}}{Q p(\bar{v} | \bar{0})}. \text{ With this definition, the Bayes' decision rule'}$$

reduces to:

$$(29a) \text{ Decide } d_1 \text{ when } \ell(\bar{v}) \geq T \text{ (signal present)}$$

$$(29b) \text{ Decide } d_0 \text{ when } \ell(\bar{v}) < T \text{ (signal absent),}$$

where:

$$(30) T = \frac{C_{\alpha}}{C_{\beta}}. \text{ Equation (29) specifies a test strategy in terms of}$$

likelihood ratio $\ell(\bar{v})$, which is a function of data $\bar{v}$; and threshold T , which is a function of error cost assignments. The Bayes decision rule divides observation space Γ into two regions Γ' and Γ'' which are separated by the boundary $\ell(\bar{v}) = T$. The acceptance region Γ'' for hypothesis $H_0 (\bar{s} = \Omega_0)$ contains all $\bar{v}$ for which $\ell(\bar{v}) < T$. The rejection region for hypothesis H_0 contains all $\bar{v}$ for which $\ell(\bar{v}) \geq T$. The rejection region for hypothesis H_0 is, of course, the acceptance region for

hypothesis $H_1(\bar{s} \in \Omega_1)$.

When P , Q , $\omega(\bar{s})$, $p(\bar{v} | \bar{s})$, and cost assignments, C_α and C_β are known, the Bayes strategy requires that the generalized likelihood ratio be computed for received data $\bar{v}$ and the result compared with a threshold T , defined by (30). In general, the computation of the likelihood ratio is a complex nonlinear operation on data $\bar{v}$. In radar, approximations for the important cases of threshold signals and very large signals permit physical interpretation of receiver structure.

2.6 Error Probabilities

Expressions for type I and type II error probabilities α and $\bar{\beta}$, respectively, are given by equations (19) and (20). These expressions apply, in general, to both Bayes and non-Bayes decision rules and are not restricted to optimum systems. In statistical terminology α , the probability of rejecting H_0 when, in fact, it is true, is called the 'level' or 'size' of the test; $1 - \bar{\beta}$ the probability of rejecting H_0 when, in fact, it is false, is called the 'power' of the test. In radar, $1 - \bar{\beta}$ is the probability of target detection.

Since observation space Γ consists of nonoverlapping regions Γ' and Γ'' , we can rewrite equations (19) and (20) for a Bayes decision rule receiver as:

$$(31) \quad \alpha = \int_{\Gamma'} p(\bar{v} | \bar{o}) D_B(d_1 | \bar{v}) d\bar{v} + \int_{\Gamma''} p(v | \bar{o}) D_B(d_1 | \bar{v}) d\bar{v}$$

$$(32) \quad \bar{\beta} = \int_{\Gamma'} \overline{p(\bar{v} | \bar{s})}_{\bar{s}} D_B(d_0 | \bar{v}) d\bar{v} + \int_{\Gamma''} \overline{p(\bar{v} | \bar{s})}_{\bar{s}} D_B(d_0 | \bar{v}) d\bar{v}.$$

Note that from earlier remarks that:

$$(33) \quad \left. \begin{aligned} D_B(d_1 | \bar{v}) &= 0 \\ D_B(d_0 | \bar{v}) &= 1 \end{aligned} \right\} \text{ for } \bar{v} \text{ in } \Gamma' \text{ and}$$

$$(34) \quad \left. \begin{aligned} D_B(d_0 | \bar{v}) &= 0 \\ D_B(d_1 | \bar{v}) &= 1 \end{aligned} \right\} \text{ for } \bar{v} \text{ in } \Gamma'', \text{ so that}$$

equations (31) and (32) simplify to:

$$(35) \quad \alpha = \int_{\Gamma''} p(\bar{v} | \bar{o}) d\bar{v}$$

$$(36) \quad \bar{\beta} = \int_{\Gamma'} \overline{p(\bar{v} | \bar{s})}_{\bar{s}} d\bar{v}. \text{ To illustrate, consider a simple example in which signal space contains a single member } \bar{s} = s, \text{ and a single observation } v \text{ is made. In this case, observation space } \Gamma \text{ may be}$$

represented by the real line $-\infty \leq v < \infty$. Probability densities $p(v | o)$ and $p(v | s)$ are graphed with real line v as abscissa, as shown in Figure 3.

Partitioning of observation space into two parts is equivalent to partitioning the real line $-\infty \leq v < \infty$ by a point v_o , which is obtained by solving $l(v) = T$ for $v = v_o$. It follows that α and $\bar{\beta}$ are given by:

$$(37) \quad \alpha = \int_{v_o}^{\infty} p(v | o) dv$$

$$(38) \quad \bar{\beta} = \beta = \int_{-\infty}^{v_o} p(v | s) dv.$$

Equations (37) and (38) state that the type I error or false-alarm probability is the area under the probability density curve $p(v | o)$ over the interval in v for which signal present is decided. The type II error or false-dismissal probability is the area under probability curve $p(v | s)$ over the interval in v for which signal absent is decided.

It can also be seen in Figure 3 that if we move the threshold v_o to the left we can eliminate the cross-hatched area and reduce the probability of error. In general, if $P \frac{p(\bar{v} | \bar{s})}{p(\bar{v} | s)} C_{\bar{\beta}} \geq Q \frac{p(\bar{v} | \bar{o})}{p(\bar{v} | o)} C_{\alpha}$, it is advantageous to have $\bar{v}$ be in Γ' so that the smaller quantity will contribute to the integral (36). This is exactly what the Bayes decision rule achieves. If $C_{\bar{\beta}} = C_{\alpha}$, and $C_{1-\alpha} = C_{1-\bar{\beta}} = 0^*$, the Bayes classifier possesses the property that the optimal decision minimizes the probability of error in classification.

* If $C_{\bar{\beta}} = C_{\alpha}$ and $C_{1-\alpha} = C_{1-\bar{\beta}} = 0$, this is called a 'symmetrical' or 'zero-one' loss function.

2.7 The Neyman-Pearson Criterion

The Neyman-Pearson theory of hypothesis testing antedates the development of statistical decision theory. It does not require knowledge of a priori signal statistics, nor does it require an explicit assignment of cost functions. An optimum test is defined as one that minimizes the probability of certain errors. In a test of hypothesis H , two types of errors can be made: H may be rejected when it is true, or it may be accepted when it is false. An optimum test is one which minimizes the probability of committing both types of errors - that is, the test should have a small probability of rejecting H when it is true and a large probability of rejecting H when it is false. A test with a probability of rejecting H when it is true is called a 'test of level ϵ '. The Neyman-Pearson criterion asserts that among all tests of level ϵ , the 'best' test is the one which has the greatest probability of rejecting H when it is false.

When applied to radar, the Neyman-Pearson test is a test between two alternative hypotheses, H_0 and H_1 , only one of which is true. The Neyman-Pearson criterion requires that, for a fixed false-alarm probability α , a test be found that minimizes the missed target-detection probability $\bar{\beta}$ or, equivalently maximizes the probability of target detection ($1 - \bar{\beta}$).

In general, hypothesis H_1 can be a simple or composite hypothesis. In the classical Neyman-Pearson test, hypothesis H_1 is assumed to be simple - that is, the signal consists of a single known value $\bar{s} = s_1$. The simple alternative hypothesis does not apply to radar since the target echo is generally a function of many variables. When signal space consists of more than one element, H_1 is a composite hypothesis.

In this case, the probability of a type II error is a function of the signal parameters. For this situation, the classical Neyman-Pearson strategy must be modified.

One extension of the Neyman-Pearson test strategy, when H_1 is composite hypothesis, is to minimize the total type II error probability that has been averaged with respect to the a priori probability density of signal. This requires a priori statistics. Thus, we minimize $P_{\bar{\beta}}$ subject to a total fixed type I error probability Q_{α} . This extension is referred to as the 'modified' Neyman-Pearson criterion. As before, P is the a priori probability of signal present. $Q = 1 - P$ is the a priori probability density of signal absent, and $\bar{\beta}$ is given by equation (20). Following the method of Lagrangian multipliers, the best decision rule D_{NP} , in the modified Neyman-Pearson sense, minimizes:

(39) $L_{NP} = P \bar{\beta} + \lambda Q_{\alpha}$, where λ is the Lagrange multiplier that is undetermined at this point. Note, equation (39) is the same as equation (21) with $C_{\alpha} = \lambda$ and $C_{\bar{\beta}} = 1$. Substituting equations (19) and (20) into (39) gives:

$$(40) L_{NP} = P \int_{\Gamma} \overline{p(\bar{v} | \bar{s})}_{\bar{s}} D(d_0 | \bar{v}) d\bar{v} + \lambda Q \int_{\Gamma} p(\bar{v} | \bar{o}) D(d_1 | \bar{v}) d\bar{v}.$$

With equation (22), (40) becomes:

$$(41) L_{NP} = \int_{\Gamma} D(d_0 | \bar{v}) [P \overline{p(\bar{v} | \bar{s})}_{\bar{s}} - \lambda Q p(\bar{v} | \bar{o})] d\bar{v} + \lambda Q.$$

This expression is minimized by choosing:

$$(42) \left. \begin{aligned} D_{NP}(d_0 | \bar{v}) &= 1 \\ D_{NP}(d_1 | \bar{v}) &= 0 \end{aligned} \right\} \quad \text{- that is, decide no signal when}$$

$$(43) \lambda(\bar{v}) = \frac{P \overline{p(\bar{v} | \bar{s})}_{\bar{s}}}{Q p(\bar{v} | \bar{o})} < \lambda, \text{ and choosing}$$

$$(44) \left. \begin{aligned} D_{NP}(d_0 | \bar{v}) &= 0 \\ D_{NP}(d_1 | \bar{v}) &= 1 \end{aligned} \right\} , \text{ (signal present) when}$$

(45) $\ell(\bar{v}) \geq \lambda$. Comparing this rule with that of equation (29), we see that the modified Neyman-Pearson strategy is identical to a Bayes test strategy with threshold $T = \lambda$. The choice of λ is not arbitrary but depends in the specification of α , since its value:

$$(46) \alpha = \int_{\Gamma} p(\bar{v} | \bar{o}) D_{NP}(d_1 | \bar{v}) \text{ is determined by the surface } \ell(\bar{v}) = \lambda \text{ separating the regions } \Gamma' \text{ and } \Gamma'' \text{ in observation space.}$$

This strategy is often employed in radar problems.

2.8 The Minimax Approach

To apply Bayes' criterion for minimizing average loss, it is necessary to know the statistics of the noise process, as well as $p(\bar{v} | \bar{s})$ and a priori signal statistics $\sigma(\bar{s})$. In many practical cases, probability density function $\sigma(\bar{s})$ is not known and it is not feasible to obtain experimental data to establish $\sigma(\bar{s})$. As a result, Bayes' criterion cannot be applied. Another criterion that may be reasonable is the 'minimax criterion'.

As an example, consider a situation in which signal space Ω contains four points, denoted by S_1 , S_2 , S_3 , and S_4 . It follows from (1b) that there is a conditional loss $L_c(D | S_i)$, $i = 1, 2, 3, 4$, associated with each member of signal space. The values of conditional loss are dependent on decision rule D . Figure 4 shows three sets of conditional losses corresponding to three different decision rules D_1 , D_2 , and D_3 . The maximum value of conditional loss is circled for each signal. Note that decision rule D_2 results in a peak or maximum conditional loss that

is less than the maximum losses resulting from decision rules D_1 and D_3 . A decision rule that minimizes the maximum conditional loss is called a 'minimax' rule. If the set of admissible decision rules contains only rules D_1 , D_2 , and D_3 , then rule D_2 is the minimax rule.

A minimax decision rule D_M results in a maximum conditional loss equal to, or less than, that resulting from any other admissible decision rule D :

$$(47) \max_{\bar{s}} L_c(D_M | \bar{s}) \leq \max_{\bar{s}} L_c(D | \bar{s}) \quad D \text{ or}$$

$$(48) \max_{\bar{s}} L_c(D_M | \bar{s}) = \max_{\bar{s}} \min_D L_c(D | \bar{s}).$$

For very general conditions, which are almost always met in practice, Wald⁽²⁾ has shown that:

$$(49) \max_{\bar{s}} \min_D L_c(D | \bar{s}) = \min_D \max_{\bar{s}} L_c(D | \bar{s}), \text{ from}$$

which the origin of the name minimax is apparent. It can be shown that a minimax decision rule D_M is a Bayes rule relative to a least-favorable a priori distribution $\sigma_{lf}(\bar{s})$. Also, the Bayes' average loss $L_M(D_M, \sigma_{lf})$, corresponding to D_M and $\sigma_{lf}(\bar{s})$, is larger than the Bayes average loss corresponding to any other a priori signal distribution, i. e.

$$(50) L_M(D_M, \sigma_{lf}) \geq L_B(D_B, \sigma) \quad \forall \sigma(\bar{s}), \text{ where}$$

L_B is the Bayes loss resulting from Bayes rule D_B and a priori signal distribution $\sigma(\bar{s})$. Thus, the minimax loss is the largest Bayes loss when all a priori distributions $\sigma(\bar{s})$ are considered.

For example, consider the case where a test was obtained for the presence of a positive mean A in Gaussian noise with variance σ^2 . When only one observation v is available, the boundary between

decision regions in observation space reduces to:

$$(51) \quad v_o = \frac{A}{2} + \frac{\sigma^2}{A} \ln \frac{C_\alpha}{P C_\beta}.$$

For known A , σ^2 , and specified C_α and C_β , v_o is a function only of P (since $Q = 1 - P$). The error probabilities can be expressed as:

$$(52) \quad \alpha = \int_{v_o(P)}^{\infty} p(v | o) dv$$

$$(53) \quad \beta = \int_{-\infty}^{v_o(P)} p(v | A) dv, \text{ from which both } \alpha \text{ and } \beta \text{ can be}$$

found as functions of P . From equation (21), the Bayes average loss is given by:

$$(54) \quad L_B(\sigma) = (1 - P) \alpha(P) C_\alpha + P \beta(P) C_\beta. \text{ This loss can be}$$

computed for various values of a priori probability P of signal present and plotted as shown in Figure 5. The maximum loss is obtained by differentiating equation (54) with respect to P and setting the result equal to zero:

$$(55) \quad \alpha(P) C_\alpha = \beta(P) C_\beta. \text{ Equation (55) can be solved for } P = P_M,$$

at which the maximum loss occurs. When $P = P_M$, the Bayes loss is equal to the minimax loss. This follows since the minimax solution corresponds to the Bayes strategy for the worst a priori signal statistics. The minimax criterion in effect compensates for ignorance of the true state of nature by assuming the worst state of nature.

To summarize, a Bayes decision rule takes into consideration all of the a priori statistics relating to both signal and noise. When signal statistics relating to both signal and noise are unavailable, a minimax decision rule sometimes offers a reasonable alternative. A minimax rule is a Bayes rule relative to a least favorable distribution; the minimax average loss is the maximum of all Bayes losses.

2.9 Bayes Solutions for Complex Cost Functions

In binary detection, signal $\bar{s}$ is a function of a number of parameters $\bar{\theta}$. Such parameters often include: amplitude ($\theta_1 = A$), time delay ($\theta_2 = \tau$), initial phase ($\theta_3 = \phi$), etc. In radar, the signal parameters provide information about various target parameters such as range, range rate, acceleration, azimuth, elevation angle, angular rate and acceleration. The signal statistics are described by the a priori (existence) probability P and the a priori probability density $\omega(\bar{\theta})$. The discussion in this section differs from that in section 2.4 in that the cost of a correct decision, $C_{1-\beta}$ is not chosen equal to zero. Instead, $C_{1-\beta}$ is assumed to be a function of signal parameters $\bar{\theta}$. In the cost matrix of Table 2, $C_{1-\beta}(\bar{\theta})$ is the cost of a correctly detected signal. Substituting this matrix and equation (11), with $\omega(\bar{s})$ replaced by $\omega(\bar{\theta})$, into equation (13) yields the average loss function:

$$(56) \quad L(D, \sigma) = Q C_{\alpha} \int_{\Gamma} D(d_1 | \bar{v}) p(\bar{v} | \bar{\theta}) d\bar{v} \\ + P C_{\beta} \int_{\Theta} \omega(\bar{\theta}) d\bar{\theta} \int_{\Gamma} D(d_0 | \bar{v}) p[\bar{v} | \bar{s}(\bar{\theta})] d\bar{v} \\ + P \int_{\Theta} \omega(\bar{\theta}) d\bar{\theta} \int_{\Gamma} C_{1-\beta}(\bar{\theta}) D(d_1 | \bar{v}) p[\bar{v} | \bar{s}(\bar{\theta})] d\bar{v}.$$

When $C_{1-\beta}(\bar{\theta})$ in (56) is set equal to a constant independent of parameters $\bar{\theta}$, it can be shown that minimizing equation (56) leads again to a Bayes solution, similar to that discussed in section 2.5, in terms of the generalized likelihood ratio of equation (28). With equation (22), equation (56) reduces to:

$$(57) \quad L(D, \sigma) = P C_{\beta} + \int_{\Gamma} D(d_1 | \bar{v}) \left\{ P \int_{\Theta} C_{1-\beta}(\bar{\theta}) \omega(\bar{\theta}) p[\bar{v} | \bar{s}(\bar{\theta})] d\bar{\theta} \right. \\ \left. - P C_{\beta} \overline{p[\bar{v} | \bar{s}(\bar{\theta})]}_{\bar{\theta}} + Q C_{\alpha} p(\bar{v} | \bar{\theta}) \right\} d\bar{v}.$$

Minimizing (57) yields the Bayes decision rule D_B :

$$(58) \left. \begin{aligned} D_B(d_1 | \bar{v}) &= 1 \\ D_B(d_0 | \bar{v}) &= 0 \end{aligned} \right\} \quad (\text{signal present}) \text{ (see equations (26a) and (26b)).}$$

when:

$$(59) P \{ C_{\bar{\beta}} p[\bar{v} | \bar{s}(\bar{\theta})]_{\bar{\theta}} - \int_{\bar{\theta}} C_{1-\bar{\beta}}(\bar{\theta}) \omega(\bar{\theta}) p[\bar{v} | \bar{s}(\bar{\theta})] d\bar{\theta} \geq Q C_{\alpha} p(\bar{v} | \bar{0}) \}$$

Table 2 - Cost Matrix for Complex Cost Functions

$$\begin{array}{c} \bar{s} = 0 \quad \text{signal } \bar{s} \\ \bar{s} \neq \bar{0} \end{array}$$

$$\text{Decision} \left\{ \begin{array}{l} d_0 \left[\begin{array}{cc} 0 & C_{\bar{\beta}} \end{array} \right] \\ d_1 \left[\begin{array}{cc} C_{\alpha} & C_{1-\bar{\beta}}(\bar{\theta}) \end{array} \right] \end{array} \right.$$

Otherwise, decide signal absent. This inequality can be rewritten as:

$$(60) \frac{P p[\bar{v} | \bar{s}(\bar{\theta})]_{\bar{\theta}}}{Q p(\bar{v} | \bar{0})} - \frac{P \int_{\bar{\theta}} C_{1-\bar{\beta}}(\bar{\theta}) \omega(\bar{\theta}) p[\bar{v} | \bar{s}(\bar{\theta})] d\bar{\theta}}{Q C_{\bar{\beta}} p(\bar{v} | \bar{0})} \geq \frac{C_{\alpha}}{C_{\bar{\beta}}}$$

The first term in (60) is the generalized likelihood ratio defined in (28). Equation (60) is similar to equations (29a) and (29b) with the addition of a second term depending on cost assignment $C_{1-\bar{\beta}}(\bar{\theta})$.

One example of this type of problem is for a signal parameter $\theta_1 = \tau$, where τ is the expected time of signal arrival, as approximated by a discrete set; assuming two different cost matrix assignments. In one case, $C_{1-\bar{\beta}}(\bar{\theta})$ is set equal to zero which yields a solution which depends on the generalized likelihood ratio and results in a Bayes receiver that averages the output of a matched filter with respect to the a priori probability density of τ .

In the second case, the cost function $C_{1-\bar{\beta}}(\bar{\theta})$ is chosen to be a step function with penalty C_m when the detection occurs at an arrival

time other than the true value. The Bayes strategy for this case is a threshold test for each of a set of discrete expected arrival times. The threshold is determined by both the cost assignments and the a priori probability density of expected arrival time τ . This strategy corresponds to the use of separate range bin tests - which is intuitively reasonable.

2.10 Preferred Neyman-Pearson Strategy

In many situations, the previous approach cannot be used because the a priori statistics are lacking on a reasonable basis for choosing cost penalties $C_i(\bar{\theta})$ is not available. An alternative is the preferred Neyman-Pearson strategy. This strategy is to find a decision surface which separates the acceptance and rejection regions (with respect to hypothesis H_0) such that Type II error probability $\beta(\bar{\theta})$ is minimized for a fixed value of α (the level of the test); or equivalently, the probability of detection (the power of the test) is maximized. Since Type II error probability $\beta(\bar{\theta})$ is in general a function of signal parameters θ , the solution differs for each set of parameters. In special cases, the test is the same for all admissible values of $\bar{\theta}$. Such a test is called 'uniformly most powerful'. These tests do not often occur.

When a uniformly most powerful solution cannot be found, other criteria can be employed. For example, the class of tests may be reduced by considering only those with some additional desirable characteristics. A uniformly most powerful test may then exist within the reduced class.

2.11 Intuitive Substitute

When a uniformly most powerful test does not exist, an alternate intuitive strategy is to average the power of the test - that is, the probability detection $P_d(\bar{\theta})$ - with respect to both the a priori probability

density function governing the signal parameters $\bar{\theta}_1$, whose statistics are unknown. A test is then sought that maximizes the average detection probability. This approach is related to the modified Neyman-Pearson strategy previously discussed in which $P_d(\bar{\theta})_{\theta_1}$ is maximized for a fixed level α and to the minimax strategy, where an averaging is performed with respect to least favorable a priori statistics $\sigma_{lf}(\bar{\theta}_2)$. This is a conservative philosophy since, on the average, the value of P_d obtained is the worst that can be expected.

For some radar parameters, solutions obtained with the intuitive substitute approach yield good results. In other cases, poor results are obtained. For example, consider the radar parameters: amplitude A , delay τ , doppler ω_d , and initial phase θ . Statistical information is often available concerning signal amplitude; this is expressed by describing the target model as Rayleigh, one dominant plus Rayleigh the so-called Swerling models (4), (5). Averaging P_d with respect to the appropriate amplitude probability density generally leads to a satisfactory result. On the other hand, the intuitive approach is generally unsatisfactory for both delay and doppler. In particular, averaging over the regions of uncertainty of delay and doppler leads to an unsatisfactory test in a multiple-target environment.

For starting phase $\emptyset$, the intuitive approach does provide satisfactory performance. A priori information concerning $\emptyset$ is usually unavailable; hence, a least favorable distribution - a uniform probability density function is employed. Averaging phase leads to an optimum receiver structure in which a matched filter is followed by an envelope detector. When compared to an optimum receiver for which $\emptyset$ is assumed to be known, it can be shown that the loss in detectability for $\emptyset$ with a uniform probability density is small (less than 1 Db) in the

region of primary concern to the radar designer, namely, high signal - to-noise ratio.

2.12 Fixed and Sequential Testing

In the foregoing discussion, it has been tacitly assumed that a decision is made after a fixed observation interval in which data are collected. The observations made during this interval may consist, in general, of discrete or (sampled) continuous input waveforms. In some systems, the observation interval is not fixed but is of variable length and is dependent on the input data. This might be an advantage where it is desirable to keep the observation interval as short as possible. For example, when a large radar echo signal is received from a nearby target, it may be desirable to take advantage of this circumstance to shorten the observation interval.

A test procedure for a variable-length observation period has been developed by Wald and is known as a 'sequential test' (6). A similar concept was considered by Neyman and Pearson in 1933 as an extension of their theory of hypothesis testing. They defined three possible decisions: accept H , reject H , and no decision. In Wald's method, it is decided whether to make a decision based upon the data already taken or to continue taking more data following each measurement. Thus, the length of the observation interval depends on the quality of the available data. Although it is theoretically possible for a test to continue indefinitely, it has been demonstrated that on the average the observation interval is shorter in a sequential test than in a fixed test. Furthermore, in practice, a sequential test is usually truncated after some predetermined number of observations.

2.13 Concluding Remarks

The application of statistical decision theory to problems in communications and radar is being actively pursued and is identical to similar efforts to apply the theory to other fields such as character, speech, and speaker recognition, weather prediction, medical diagnosis, and stock market prediction.⁽³⁾ Despite its power, certain limitations, restrict its range of application. These limitations result from the requirements on the system model which can never be completely satisfied in practice.

One limitation has to do with cost assignments. Assignments are usually made by the system designer, and therefore, are subject to individual bias. Fortunately in many applications, the structure of the optimum system is insensitive to variations in cost assignment. For example, the structure of a Bayes receiver for simple radar binary detection is independent of the magnitude of the preassigned costs. This is not true for complex cost assignments, however.

A more fundamental limitation stems from the need for a priori information concerning both the signal and noise processes. If such information is not available, the theory cannot be rigorously applied. If a priori information is available about the noise process but signal statistics are unavailable, a solution may be possible by invoking other criteria such as minimax.

In cases where more sophistication is required and less sensitivity to underlying distributions is desired ('robust' procedures), an adaptive system in which the system decision rule varies as 'learning' takes place is desirable.

The use of a decision tree, evaluating the features sequentially until a decision is made, requires considering the cost of measuring features as well as the cost of making errors. This is the subject of sequential decision theory (6). Reference (13) shows how techniques developed for searching game trees can be applied to such problems.

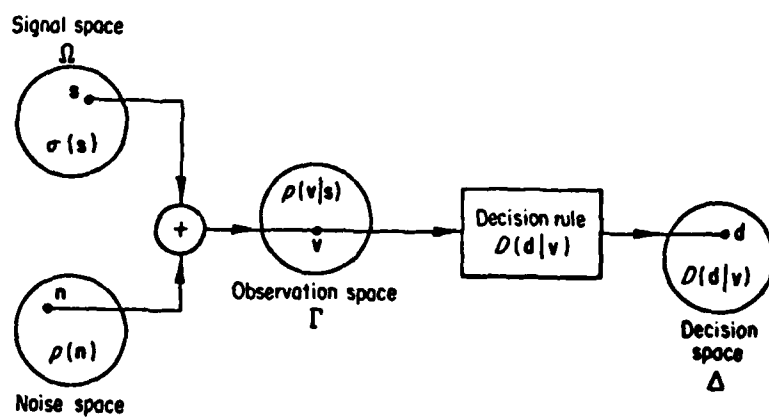


Figure 1. Reception as a decision problem

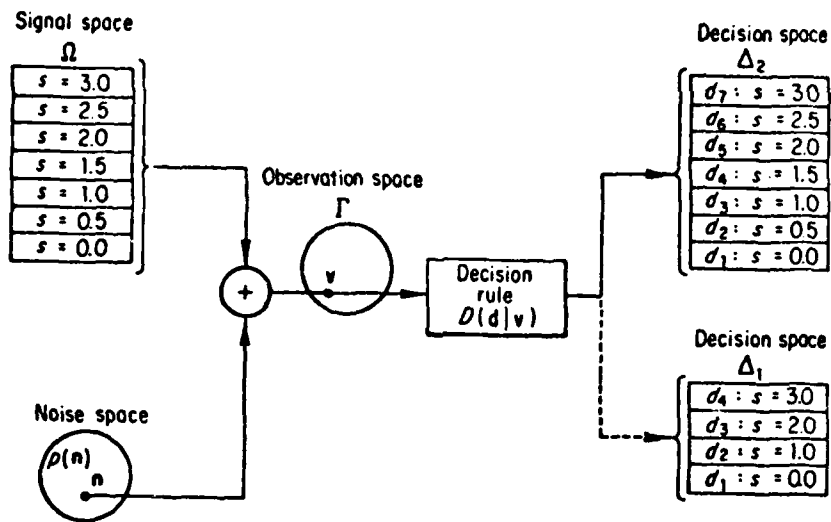


Figure 2. Example of parameter estimation as a decision problem

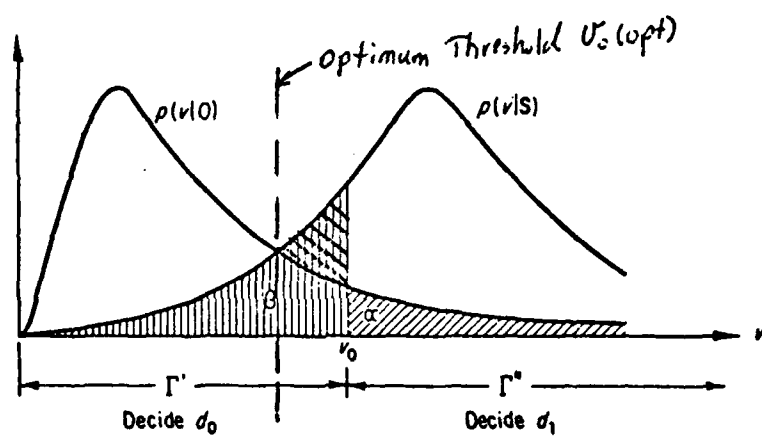


Figure 3. Error probabilities

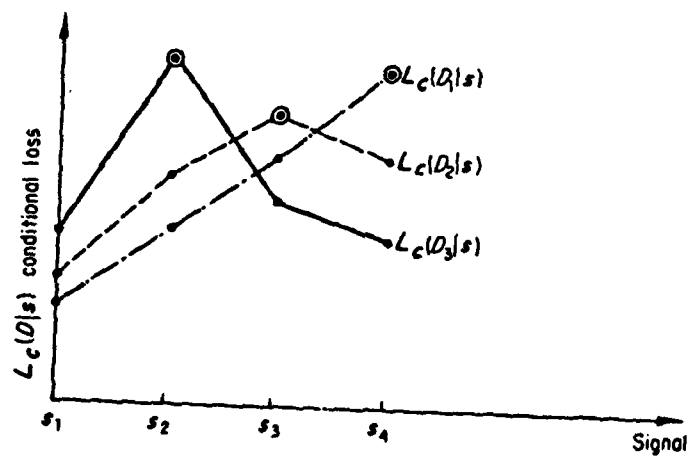


Figure 4. Conditional loss function for discrete s as a function of decision rule D_i

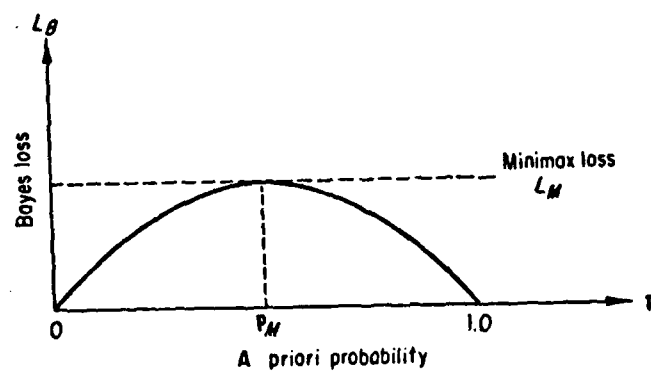


Figure 5. Bayes loss and minimax solution

3.0 Parameter Estimation and Supervised Learning (7)

In section 2.6 it was shown that an optimal classifier for detection, (two-class problem), could be designed if the a priori probabilities, P and Q , and the class-conditional densities $p(\bar{v} | \bar{s})_{\bar{s}}$ and $p(\bar{v} | \bar{0})$ were known. Unfortunately, in pattern recognition applications we rarely have this 'luxury'. In a typical case we have some vague, general knowledge and a number of design samples, the classification of which are known.

One approach to designing the classifier is to use the samples; waveforms in which we know that a target(signal) is/is not present to estimate the unknown probabilities and probability densities - and use the resulting estimates as if they were true values. Usually, the estimation of the class-conditional densities is not feasible since the number of available samples (waveforms) is almost always too small for the time available. If we can parameterize the conditional densities, the severity of the problem can be significantly reduced. Suppose, for example, that we can reasonably assume the $p(\bar{v} | \bar{s})_{\bar{s}}$ comes from a distribution with mean $\bar{\mu}_{\bar{s}}$ and covariance matrix $\bar{\Sigma}_{\bar{s}}$, although we do not know the exact values of these quantities. The problem is then simplified to be one of estimating $\bar{\mu}_{\bar{s}}$ and $\bar{\Sigma}_{\bar{s}}$, and not the probabilities.

The problem of parameter estimation can be approached in several ways. Two of these procedures, outlined in Section 2 are the 'maximum likelihood' estimation and 'Bayesian' estimation. Although the results obtained by these two procedures are often nearly identical, the approaches are conceptually quite different. Maximum likelihood methods view the parameters as quantities whose values are fixed but unknown. The best estimate is defined to be one that maximizes the probability of obtaining the samples actually observed. Bayesian methods view the

parameters as random variables having some known a priori distribution. Observation of the samples converts this to an a posteriori density changing our opinion about the true parameter values.

In the Bayesian case, the typical effect of observing additional samples is to sharpen the a posteriori density function, causing it to peak near the true values of the parameters. This phenomenon is known as 'Bayesian learning'. We shall consider only this case here.

3.1 General Bayesian Learning

Let us assume that the class 'target present' is signified by the symbol ' ω_2 ', and target absent by ' ω_1 '. Let X denote a set of samples (e. g., waveforms representing a scan of the radar). We can emphasize the role of the samples by stating that our goal is to compute the a posteriori probabilities $P(\omega_i | \bar{v}, X)$. From these probabilities we obtain the Bayes classifier:

$$(61) \quad P(\omega_i | \bar{v}, X) = \frac{p(\bar{v} | \omega_i, X) P(\omega_i, X)}{\sum_{j=1}^2 p(\bar{v} | \omega_j, X) P(\omega_j | X)}$$

Thus, we can use the information provided by the samples to help determine both the class-conditional densities and the a priori probabilities.

It will be assumed that the true values of the a priori probabilities are known so that $P(\omega_i | X) = P(\omega_i)$. Thus, in our case $P(\omega_1) = Q$, and $P(\omega_2) = P$. Furthermore, in treating the 'supervised learning' case we can separate the samples by class into two subsets X_1 and X_2 , with the samples in X_i belonging to ω_i . In the cases treated here, we assume that the samples in X_j have no influence in $p(\bar{v} | \omega_i, X)$ if $i \neq j$. This has two simplifying consequences. First, it allows us to work with each

class separately, using only the samples in X_i to determine $p(\bar{v} | \omega_i, X)$.

This allows us to write equation (61) as:

$$(62) \quad P(\omega_i | \bar{v}, X) = \frac{p(\bar{v} | \omega_i, X_i) P(\omega_i)}{\sum_{j=1}^2 p(\bar{v} | \omega_j, X_j) P(\omega_j)}$$

A second simplifying consequence is that each class can be treated independently, and we can dispense with needless class distinctions and simplify our notation. In essence, we have 2 separate problems of the following form: use a set X of samples drawn independently according to the fixed but unknown probability $p(\bar{v})$ to determine $p(\bar{v} | X)$.

Although the desired probability density $p(\bar{v})$ is unknown, we assume that it has a known parametric form. The only thing assumed unknown is the value of the parameter vector $\bar{\theta}$. The fact that $p(\bar{v})$ is unknown, but of known parametric form, will be expressed by saying that the function $p(\bar{v} | \bar{\theta})$ is completely known. The Bayesian approach assumes that the unknown parameter vector is a random variable. Any information we might have about $\bar{\theta}$ prior to observing the samples is assumed to be contained in a known a priori density $p(\bar{\theta})$. Observation of the samples converts this to an a posteriori density $p(\bar{\theta}, X)$, which we hope will be sharply peaked about the true value of $\bar{\theta}$.

Our basic goal is to compute $p(\bar{v} | X)$, which is as close as we can come to obtaining the unknown $p(\bar{v})$. We do this by integrating the joint density $p(\bar{v}, \bar{\theta} | X)$ over $\bar{\theta}$:

$$(63) \quad p(\bar{v} | X) = \int p(\bar{v}, \bar{\theta} | X) d\bar{\theta}, \text{ where the integration extends over the entire parameter space. We can always write } p(\bar{v}, \bar{\theta} | X) \text{ as the product of } p(\bar{v} | \bar{\theta}, X) p(\bar{\theta} | X). \text{ Since the selection of } \bar{v} \text{ and of the samples in } X \text{ is done independently the first factor is merely } p(\bar{v} | \bar{\theta}). \text{ That is, the}$$

distribution of $\bar{v}$ is known completely once we know the value of the parameter vector.

Thus:

$$(64) \quad p(\bar{v} | X) = \int p(\bar{v} | \bar{\theta}) p(\bar{\theta} | X) d\bar{\theta}.$$

Equation (64) links the desired density $p(\bar{v} | X)$ to the a posteriori density $p(\bar{\theta} | X)$ for the unknown parameter vector. If $p(\bar{\theta} | X)$ peaks very sharply about some value $\hat{\bar{\theta}}$, we obtain $p(\bar{v} | X) \approx p(\bar{v} | \hat{\bar{\theta}})$, which is the result we would obtain by substituting the estimate $\hat{\bar{\theta}}$ for the true parameter vector. If we are less certain about the exact value of $\bar{\theta}$, equation (64) directs us to average $p(\bar{v} | \bar{\theta})$ over the possible values of $\bar{\theta}$. Thus, when the unknown densities have a known parametric form, the samples exert their influence in $p(\bar{v} | X)$ through the a posteriori density $p(\bar{\theta} | X)$.

The basic assumptions for Bayesian learning are then:

- (1) The form of the density $p(\bar{v} | \bar{\theta})$ is assumed to be known, but the value of the parameter vector $\bar{\theta}$ is not known exactly.
- (2) Our initial knowledge about $\bar{\theta}$ is assumed to be contained in a known a priori density $p(\bar{\theta})$.
- (3) The rest of our knowledge about $\bar{\theta}$ is contained in a set X of n samples $\bar{v}_1, \bar{v}_2, \dots, \bar{v}_n$ drawn independently according to the unknown probability law $p(\bar{v})$.

The basic problem is to compute the a posteriori density $p(\bar{\theta} | X)$, since from this we can use equation (64) to compute $p(\bar{v} | X)$.

By Bayes rule,

$$(65) \quad p(\bar{\theta} | X) = \frac{p(X | \bar{\theta}) p(\bar{\theta})}{\int p(X | \bar{\theta}) p(\bar{\theta}) d\bar{\theta}}, \text{ and by the assumption that the}$$

samples are independent:

$$(66) \quad p(X | \bar{\theta}) = \prod_{k=1}^n p(\bar{v}_k | \bar{\theta}).$$

This constitutes the formal solution to the problem. Equations (64) and (65) illuminate its relation to the maximum likelihood solution. Suppose that $p(X | \bar{\theta})$ reaches a sharp peak at $\bar{\theta} = \hat{\bar{\theta}}$. If the a priori density $p(\bar{\theta})$ is not zero at $\bar{\theta} = \hat{\bar{\theta}}$ and does not change much in the surrounding neighborhood, then $p(\bar{\theta} | X)$ also peaks at that point. Thus, equation (64) shows that $p(\bar{v} | x)$ will be approximately $p(\bar{v} | \hat{\bar{\theta}})$, the result obtained by using the maximum likelihood estimate as the true value. If the peak of $p(X | \bar{\theta})$ is not so sharp that the influence of a priori information or the uncertainty in the true value of $\bar{\theta}$ can be ignored, then the Bayesian solution tells us how to use the available information to compute the desired density $p(\bar{v} | X)$.

To indicate explicitly the number of samples in a set, we write $X^n = \bar{v}_1, \bar{v}_2, \dots, \bar{v}_n$. Then, from equation (65), if $n > 1$,

$$(67) \quad p(X^n | \bar{\theta}) = p(\bar{v}_n | \bar{\theta}) p(X^{n-1} | \bar{\theta}).$$

Substituting equation (67) into equation (65) and using Bayes' rule,

$$(68) \quad p(\bar{\theta} | X^n) = \frac{p(\bar{v}_n | \bar{\theta}) p(\bar{\theta} | X^{n-1})}{\int p(\bar{v}_n | \bar{\theta}) p(\bar{\theta} | X^{n-1}) d\bar{\theta}}$$

With the understanding that $p(\bar{\theta} | X^0) = p(\bar{\theta})$, repeated use of this equation produces the sequence of densities $p(\bar{\theta})$, $p(\bar{\theta} | \bar{v}_1)$, $p(\bar{\theta} | \bar{v}_1, \bar{v}_2)$, etc.

This is called the 'recursive' Bayes approach to parameter estimation. When this sequence of densities converges to a Dirac delta function centered about the true parameter value, the resulting behavior is frequently called 'Bayesian learning'.

For most of the typically encountered probability densities $p(\bar{v} | \bar{\theta})$, the sequence of a posteriori densities does converge to a delta function. This implies that with a large number of samples there is only one value for $\bar{\theta}$ that causes $p(\bar{v} | \bar{\theta})$ to fit the data, i. e., that $\bar{\theta}$ can be determined uniquely from $p(\bar{v} | \bar{\theta})$. When this is the case, $p(\bar{v} | \bar{\theta})$ is said to be 'identifiable'.

There are occasions, however, when more than one value of $\bar{\theta}$ may yield the same value for $p(\bar{v} | \bar{\theta})$, (the 'multimodal' case). In such cases, $\bar{\theta}$ cannot be determined uniquely from $p(\bar{v} | \bar{\theta})$, and $p(\bar{\theta} | X^n)$ will peak near all of the values of $\bar{\theta}$ that explain the data. Fortunately, this ambiguity is erased by the integration in equation (64), since $p(\bar{v} | \bar{\theta})$ is the same for all of these values of $\bar{\theta}$. Thus, $p(\bar{v} | X^n)$ will typically converge to $p(\bar{v})$ whether or not $p(\bar{v} | \bar{\theta})$ is identifiable when supervised learning is considered. When the classification of samples is not known a priori as in 'unsupervised learning', identifiability is one of the major problems.

Appendix A includes a description of a Bayesian classifier program (20).

4.0 Unsupervised Learning and Clustering

In supervised learning, the membership of the training samples used to design the classifier are assumed known. In 'unsupervised' learning, the membership of the training samples is unknown a priori.

This is precisely the type of problem characterized by the radar detection of a target in a background of noise and clutter. The reasons for this are as follows.

Firstly, the collection and labeling of a large set of sample patterns and their categorization can be costly and time consuming. If we could crudely design a classifier based upon a small set of samples whose classification is known, and then allow it to run without supervision on a large, unlabeled set, we might save a good deal of effort.

Secondly, in applications such as the radar detection of targets in ground clutter, the signal - as well as the background - can change slowly with time. An unsupervised mode classifier can track these changes and make timely corrections.

Additionally, in the early stages of an investigation such as this, it is necessary to gain some insight into the nature and structure of the data as applied to pattern recognition. The discovery of unanticipated subclasses may significantly alter the classifier design.

4.1 Mixture Densities and Identifiability

As a take-off point, let us assume the following:

- (1) The samples come from two classes.
- (2) The a priori probabilities, P and Q , are known.
- (3) The forms for the class-conditional probability densities $p(\bar{v} | \omega_j, \bar{\theta}_j)$, $j = 1, 2$ are known.

(4) All that is unknown are $\bar{\theta}_1$ and $\bar{\theta}_2$.

The probability density function for samples assumed to be obtained by selecting a state of nature with the a priori probabilities P and Q is:

$$(69) \quad p(\bar{v} | \bar{\theta}) = p(\bar{v} | \omega_1, \bar{\theta}_1) P + p(\bar{v} | \omega_2, \bar{\theta}_2) Q.$$

A density of this form is called a 'mixture density'. The conditional densities $p(\bar{v} | \omega_j, \bar{\theta}_j)$ are called the 'component densities', and P and Q are the 'mixing parameters'. The mixing parameters can be included among the unknown parameters, but we shall assume that only the $\bar{\theta}_j$'s are unknown.

As discussed in section 3.1, a density $p(\bar{v} | \bar{\theta})$ is said to be 'identifiable' if $\bar{\theta} \neq \bar{\theta}'$ implies that there exists a $\bar{v}$ such that $p(\bar{v} | \bar{\theta}) \neq p(\bar{v} | \bar{\theta}')$.

Most mixtures of commonly encountered density functions are identifiable. Discrete distribution mixtures are often not identifiable. We will assume further that the mixture densities are identifiable.

4.2 Clustering⁽¹⁵⁾

Although nothing is assumed to be known about the category structure, one frequently has some intuitive feelings about desirable and undesirable features for a classification scheme. One might ask, "Why not enumerate all the possibilities and choose the best"?

The number of ways of sorting n observations into m groups is given by:

$$S_n^{(m)} = \frac{1}{m!} \sum_{k=0}^m (-1)^{m-k} \begin{bmatrix} m \\ k \end{bmatrix} k^n.$$

Even for the detection problem, where $m = 2$, and a number of observations $n = 25$, the number of combinations is 16,777,215. For $n = 25$, $m = 3$, the number grows to 141,197,991,025. ⁽¹⁴⁾ p. 835

If the number of groups (classes) is unknown, the number of possibilities rises to $> 4 \times 10^{18}$. This makes an exhaustive examination of the alternatives impractical.

Cluster algorithms are used to generate hypotheses about category structure. A role often exploited is that of discovering 'natural classes'. If a suitable algorithm is applied to a set of data, and the resulting clusters are only weakly differentiated, then the data probably belong to only one class. Thus, the user of a clustering algorithm is often trying to understand the data set and uncover what structure resides in the data.

4.2.1 Clustering Methodology⁽¹⁶⁾

As discussed in Reference 16, the number of clustering techniques can be considered in three groups - minimization of squared error, hierarchical, and graph-theoretic. Each of these techniques will be briefly discussed.

.1 Squared-Error Clustering Algorithms

Squared-error algorithms try to define clusters which are hyperellipsoidal in shape. Let the i th pattern, $i = 1, \dots, n$ from the data set under study be written as:

$$(70) \quad x_i = (x_{i1}, x_{i2}, \dots, x_{iN})^T, \text{ where } (\quad)^T$$

indicates the vector transpose (a column vector in this case). The number of patterns, n , is assumed to be much greater than the number of features, N . A clustering is a partition $[C_1, C_2, \dots, C_K]$ of the integers $[1, 2, \dots, n]$ that assigns each pattern a single cluster label. The patterns corresponding to the integers in C_k form the k th cluster, whose center is:

$$(71) \quad c_k = (c_{k1}, c_{k2}, \dots, c_{kn})^T, \text{ where}$$

$$(72) \quad c_{kj} = \frac{1}{M_k} \sum_{i \in C_k} x_{ij}, \text{ and } M_k \text{ is the cardinality of } C_k \text{ (the}$$

number of patterns in cluster k). Thus, a cluster center is the centroid, or sample mean, of all patterns in the cluster.

The squared error for cluster k is:

$$(73) \quad e_k^2 = \sum_{i \in C_k} (x_i - c_k)^T (x_i - c_k), \text{ and the squared error for the}$$

clustering is:

$$(74) \quad E_K^2 = \sum_{k=1}^K e_k^2.$$

The squared error for eq. (74) can be expressed in many ways, such as the sum of "within" and "between" squared errors used in discriminant analysis.⁽⁷⁾ The objectives are to define, for a given K, a clustering that minimizes E_K^2 , and to find a suitable K, much smaller than n. Since an exhaustive search is computationally infeasible, the various squared-error programs implement different tactics for searching through the possible clusterings. All programs try to find a local minimum of E_K^2 . The user hopes that this local minimum also coincides with the global minimum.

An example of such a methodology is Forgy's method, for which a simplified flow chart is shown in Fig. 4.1. The heart of the method is the inner loop in Fig. 4.1 which establishes the way in which clusters are updated. Given a set of cluster centers, the cluster label of the closest cluster center is assigned to each pattern. The cluster centers are then recomputed as sample means, or centroids, of all patterns having the same cluster label.

A new cluster is created in the inner loop when a pattern is found that is sufficiently removed from the existing structure. Let $d_k(i)$ be the distance between pattern i and cluster center k . Let $\bar{d}(i)$ be the average distance from pattern i to all K cluster centers.

$$(75) \quad \bar{d}(i) = \frac{1}{K} \sum_{k=1}^K d_k(i).$$

A new cluster is created, centered at pattern i if:

$$(76) \quad |d_{k_o}(i) - \bar{d}(i)| \leq |\bar{d}(i) T_F|, \text{ where } k_o \text{ is the cluster center}$$

closest to pattern i and T_F is a user-supplied threshold between zero and one. The larger T_F , the more new clusters that will be created. The inner loop is repeated until either two successive passes through all patterns produce the same clustering or a user-supplied limit, L_F , on the number of loops has been exceeded. The number of patterns, M_k , in cluster k is then computed for each k and compared to the user-supplied number N_F . If $M_k < N_F$, all patterns in cluster k are removed and henceforth ignored. Thus, such patterns are considered to be 'outliers'. This is the only means available in FORGY for reducing the number of clusters.

FORGY was constructed to be as direct as possible. The initialization procedure follows this philosophy and fixes the initial number of cluster centers by selecting K_F patterns, where K_F is supplied by the user.

A program listing of the program which analyzes data stored in core memory via Forgy's and Jancey's method as written for a CDC 6500-type computer is shown in Appendix B.

. 2 Hierarchical Clustering

The hierarchical clustering techniques produce a 'dendrogram' which describes the clustering of the patterns. The dendrogram connects groups of patterns at levels of similarity. It may be used to group the patterns into a given number of clusters as well as to indicate how many clusters there are at a given similarity level. Similarity is often defined from the interpattern distances. A program description of such a program is described in Appendix C.

These techniques begin with a triangular dissimilarity matrix, whose rows and columns correspond to patterns, and whose entries measure dissimilarity between patterns; the larger the entry, the more dissimilar the patterns.

The number of patterns that can be handled by such methods is limited since such techniques are very expensive in computer time and memory.

. 3 Graph-Theoretic Methods

Not all natural groupings of patterns are globular or hyperellipsoidal in shape. For example, patterns that are spaced along a straight line or in a plane in the pattern space are well structured. Squared-error methods force a globular or Gaussian-based model on such structures and cannot work. Graph-theoretic methods provide one means for uncovering unconventional data structures.

One example is the technique of Zahn⁽¹⁷⁾ to produce a minimal spanning tree. A description of the program is included in Appendix D. The routine generates a minimal spanning tree, and then evaluates

the tree for self-consistent clusters of patterns. The algorithm used is that of Prim and Dijkstra^{(18), (19)} as implemented by Whitney⁽²⁰⁾.

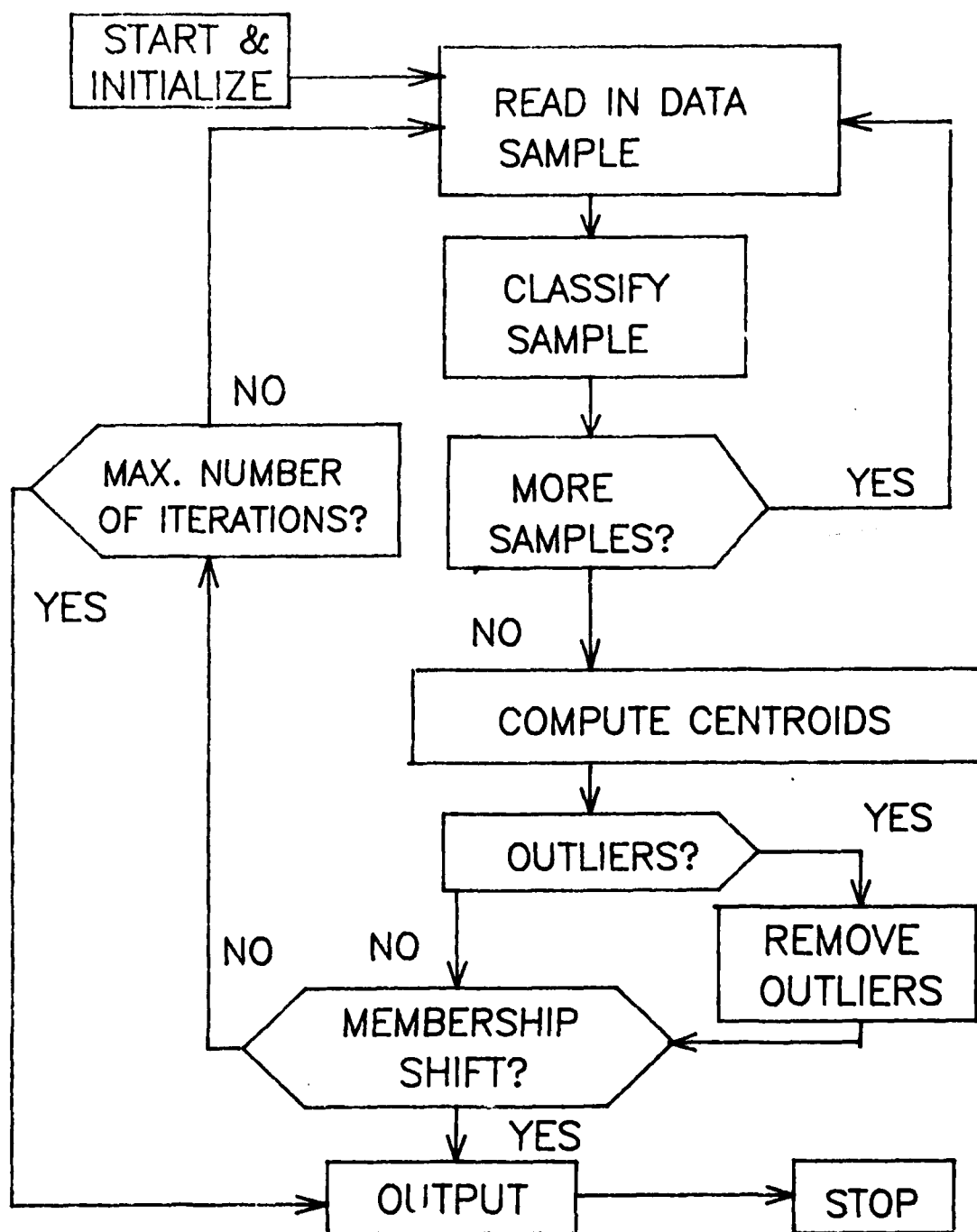


FIGURE 4.1 — FORGY/JANCEY METHOD

5.0 Testing Methods

Given a set of data and various algorithms to analyze the data, the question to answer is, "which algorithm performs best with the given data?"

If the underlying phenomenology of the data is either unknown or changes so much due to the influence of various factors, a parametric statistical technique may become impractical. Considering what is known about the statistics of radar ground clutter, these remarks seem to apply.

The term "best" in the context of this problem will be defined to mean the algorithm which yields the lowest average probability of error when operating on a known data set. This definition may not be entirely adequate when faced with implementing the algorithm to give real-time radar operation. However, the overriding concern at the initial evaluation phase is to determine which of the various algorithms proposed are most compatible with the general data structure presented by radar ground clutter.

Even at this early evaluation phase, certain constraints on economy of cost and computer running time force the consideration of a limited data set for the evaluation.

Since the data set is to be limited for economy, we see that the problem with having only a small number of samples is that the resulting classifier will not perform well on new data. The error rate is therefore expected to be a function of the number of samples, typically decreasing to some minimum value as the number of samples becomes much larger.

One approach to estimating the error rate is to compute from an empirically derived parametric model. There are many pitfalls to this approach - not the least of which is the uncertainty of the underlying probability distributions.

An empirical approach is to test the classifier experimentally. In practice, this is frequently done by running the classifier on a set of test samples using the fraction of the samples misclassified as an estimate of the error probability. Obviously the test samples should be different from the design samples, or the results will be highly optimistic. If the true but unknown error rate of the classifier is "p", and if k of the n independent, randomly drawn tests samples are misclassified, then k has the binomial distribution.

$$(77) \quad P(k) = \binom{n}{k} p^k (1 - p)^{n-k} .$$

Thus, the fraction of test samples misclassified is the maximum likelihood estimate for p:

$$(78) \quad \hat{p} = \frac{k}{n} .$$

The properties of p for a binomial distribution are well known. Figure 5.1 shows the 95% confidence intervals as a function of $\hat{p}$ and n .⁽⁷⁾ For a given value of $\hat{p}$, the probability is 0.95 that the true value of p lies between the upper and lower curves for the number n of test samples. These curves show that unless n is large, this maximum likelihood estimate should be carefully interpreted. For example, if no errors are made on 50 test samples, the true error probability lies between 0 and

8% with probability 0.95. The classifier would have to make no errors on more than 250 samples to be reasonably sure that the true error rate is below 2%.

The need for data to design the classifier and additional data to evaluate it presents a dilemma when the number of samples has been limited. If most of the data is reserved for design, the test will not be reliable. If most of the data is reserved for test, the design will be poor. The questions of how best to partition a set of samples into a design set and a test set cannot be answered definitively.

The technique that comes closest to the true error probability is the "leaving-one-out method." This involves running the classifier to train it, (design), using $n-1$ samples, and testing it on the remaining sample. The classifier is then run n times, leaving each sample out for a given run. Thus, almost all of the samples are used in the design, which should lead to a good design. Also, all of the samples are used for test. The problem with this technique is that it is only practical when n is quite small.

A practical compromise is the π or rotation method²². In this technique, a small subset of P pattern samples is chosen, where $1 \leq P \leq n$, n/P is an integer, and $P/n < 1/2$. The classifier is trained on the $n-P$ remaining samples, and tested in the P samples. An estimate of the error probability $\hat{P}_e[\pi]_i$ is obtained for the i th run. The runs are made n/P times, using a different set of P samples each time for test, and

training on the remaining samples. The resulting approximate estimate of $\hat{P}_e$ is calculated as:

$$(79) \quad E\{\hat{P}_e[\pi]\} = \frac{P}{n} \sum_{i=1}^{n/P} \hat{P}_e[\pi]_i, \text{ where}$$

$E\{\hat{P}_e[\pi]\}$ is the expected value of $\hat{P}_e[\pi]$. Note that when $P = 1$, the method reduces to the leaving-one-out method. Reference (21) also suggests that a better estimate of the true error probability might be obtained by:

$$(80) \quad \hat{P}_e^* = 1/2 [E\{\hat{P}_e[\pi]\} + E\{\hat{P}_e[R]\}], \text{ where}$$

$E\{\hat{P}_e[R]\}$ = the estimated error probability based upon training on all of the samples and testing all of the samples, giving as previously discussed a highly optimistic error probability, but a reasonable lower bound to the true error probability.

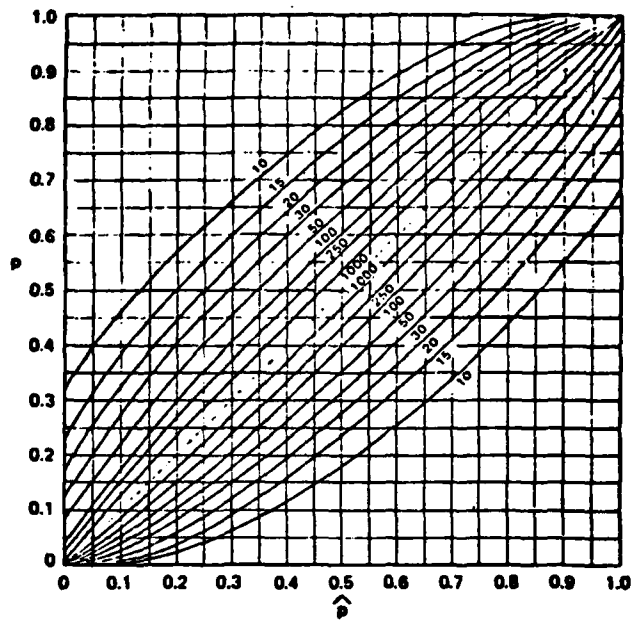


Figure 5.1. Confidence Intervals for Error-Rate Estimates

6.0 Tests with Simulated Radar Data

Preliminary evaluation of the various algorithms was conducted using simulated radar data supplied by W.L. Simkins, Jr., of RADC. The data consisted of 65,536 (256 x 256) samples in x-y presentation. Ground clutter measurements for each xy coordinate in amplitude and doppler were included. Details of the data are described in Appendix E.

Test runs were made to determine the amount of processing time necessary for each of the algorithms to be evaluated. Figure 6.0.1, taken from Appendix E, shows the processing times as a function of the number of samples for two typical algorithms - the Bayes classifier and the kNN algorithm. ⁽³⁾

It can be seen from Figure 6.0.1 that even though the kNN algorithm is evaluated for $k = 1, 3, 4, 5, 6, 7, 8, 9$, and 10 in parallel; and therefore, takes longer than a run with only $k = 1$, for example, the processing times are too great to allow all 65,536 samples to be used. Table 6.0.1 summarizes the processing times for 257 samples to be run for each of the algorithms.

<u>Algorithm</u>	Processing Times $n = 257$	1024
	<u>Times (Min)</u>	<u>Times (Min)</u>
Bayes	0.877	2.8
NN	2.63	38.3
Hierarchical Clustering	83.497	--
Minimal Spanning Tree	2.718	--

Table 6.0.1

Based upon these conclusions, runs were made to evaluate the various algorithms using a sample set including up to 1,024 samples. The limited field of view included the region from $x = 1$ through 128, and $y = 128$ through 135 as described in Appendix E. The selection was chosen to give a wide variety of amplitude and doppler values.

15 runs were made for each of 5 sample sizes, (60, 135, 255, 510, and 1005), for the Bayes and kNN algorithms. The π (Rotation) method was used to evaluate the error probabilities - as discussed in Section 5.0. Typical results for these algorithms is shown in Table 6.0.2 and 6.0.3. In these tables, the first column indicates the number in the sample set. The second column indicates the error probability obtained by "testing on the training set." The third column is the expected value of the error probability as obtained by the π method. The fourth column is the result of applying equation (80) to columns 2 and 3. As discussed by Toussaint and Sharpe, this yields a closer estimate of the true average error probability $\frac{2}{21}$. The fifth column in Table 6.0.3 is 1/2 of the fourth column, and represents an estimate of the true Bayesian error probability.

$$P_{\text{total}} [X_k | x_i] = \sum_{j=1}^n P_j [X_j, k | x_{i,j}]^\alpha$$

Bayes Classifier - $\alpha = 0.5$

x, y, Amplitude, Doppler - Doppler Categories

<u>Number of Samples</u>	<u>P_e [R]</u>	<u>E{ $\hat{P}_e [\pi]$ }</u>	<u>$\hat{P}_e$</u>
60	0.0500	0.6167	0.3334
135	0.0296	0.5111	0.2704
255	0.0353	0.4157	0.2255
510	0.1294	0.5725	0.3510
1005	0.4657	0.3771	0.4214

Table 6.0.2. Error Probabilities

1NN Algorithm,

x, y, Amplitude, Doppler - Doppler Categories

<u>Number of Samples</u>	<u>P_e [R]</u>	<u>E{ $\hat{P}_e [\pi]$ }</u>	<u>$\hat{P}_e$</u>	<u>$\hat{P}_e^*$</u>
60	0.1833	0.2167	0.2000	0.1000
135	0.0519	0.1037	0.0778	0.0389
255	0.0784	0.0980	0.0882	0.0441
510	0.0078	0.0098	0.0088	0.0044
1005	0.0010	0.0050	0.0030	0.0015

Table 6.0.3. Error Probabilities

7.0 Tests With Actual Radar Data

W. L. Simkins, Jr. also has supplied a test tape containing samples of actual radar data. The data consists of a number of files containing ρ , ϕ , and Amplitude information. Some runs were made, using the Bayes and kNN algorithms. The results so far obtained were comparable to those obtained with the simulated radar data discussed in Section 6 and Appendix E of this report. However, the number of runs made were insufficient to give definitive information about the data and how each of the algorithms performed.

8.0 Summary, Conclusions and Recommendations

Preliminary runs testing simulated radar data with a number of conventional pattern recognition algorithms indicated that the 1-nearest-neighbor nonparametric algorithm showed promise in producing low error probabilities. As discussed in Appendix E, Figure 6.0.2(b), the errors made were primarily at the transitions between one class and another. This indicates that combining the nearest-neighbor algorithm with a gradient technique to sense the "edges", or boundaries between classes might produce fewer errors.

The times required for the nearest-neighbor algorithm to process only up to 1,024 samples are much too large to yield practical real-time processors in a radar. However, there are techniques which can significantly reduce these times²².

Although runs were made with the Minimal Spanning Tree Algorithm, the results obtained were indifferent at best. One reason for this is that although the technique to produce the minimal spanning tree is an efficient one, the "pruning" of the tree, as the algorithm is presently constituted, does not allow a prior selection of the number of clusters. Thus, the resulting clusters - not being under the control of the program - tend to be different from the natural grouping of the data.

In the light of these preliminary findings, the following recommendations are made for follow-on activity.

- Additional tests using actual radar data comparable in format to the simulated data should be made of at least the nearest-neighbor algorithm.

- Gradient or edge-detection methods should be investigated and incorporated into whatever algorithm is employed.
- Investigation and incorporation of techniques to allow at least two orders of magnitude of data in real-time should be pursued, particularly for the nearest-neighbor algorithm.
- The minimal spanning tree algorithm should be modified to allow the selection of the number of clusters required of the data and tested with simulated and actual radar data.
- A fuzzy k-means algorithm should be incorporated into the evaluation process. This technique offers some promise in the type of problem presented by radar ground clutter²³.

Glossary

c_k	Cluster center of k th cluster
$\bar{d}(i)$	Average distance from pattern i to all cluster centers
$d_k(i)$	Distance between pattern i and cluster center k
e_k	Squared error for cluster k
$k_o(i)$	Cluster center closest to pattern i
$\ell(\bar{v})$	Generalized likelihood ratio
$\binom{m}{k}$	Binomial coefficient indicating the combination of m things taken k at a time
$\bar{o}$	Null vector
$p(\bar{d}, \bar{v} \bar{s})$	Joint conditional probability density function that decision $\bar{d}$ and waveform $\bar{v}$ will occur given that signal $\bar{s}$ has occurred
$p(\bar{n})$	A priori joint probability density function over all noise signals $\bar{n}$ in noise space
$p(\bar{v} \bar{s})$	Conditional probability density function that a particular waveform $\bar{v}$ will occur, given that a signal $\bar{s}$ has occurred
$\overline{p(\bar{v} \bar{s})}_s$	Expectation of $p(\bar{v} \bar{s})$ over all signals $\bar{s}$
$\bar{v}$	All possible joint combinations of signal and noise waveforms within the observation interval in observation space
A	Average amplitude of waveform
$C(\bar{s}, \bar{d})$	Quantitative cost associated with each point $\bar{s}$ in Ω (signal space) and each point $\bar{d}$ in Δ (decision space)
$C_{1-\alpha}$	Cost associated with correctly deciding a signal is present
$C_{1-\beta}$	Cost associated with correctly deciding a signal is absent
C_α	Penalty associated with deciding that signal is present, when there is no signal
C_β	Cost associated with deciding no signal, when there <u>is</u> a signal

D_B	Bayes decision rule
$D(d \bar{v})$	Decision rule leading to a decision d , having observed a waveform $\bar{v}$
D_M	Minimax decision rule
D_{NP}	Neyman-Pearson decision rule
E_K	Squared error for a clustering
H_0	Null hypothesis (i.e., that noise alone is present)
H_1	Composite alternate hypothesis (i.e., that signal plus is present)
K_F	Initial choice of number of clusters
$L_B(\sigma)$	Bayes average loss for an a priori distribution $\sigma(\bar{s})$
$L_c(D \bar{s})$	The mathematical expectation of the loss with respect to all possible decisions $\bar{d}$ for a given $\bar{s}$ and decision rule D
$L(D, \sigma)$	Average loss for a known a priori probability density $\sigma(\bar{s})$ and decision rule D . The sum of costs associated with decisions $\bar{d}$ and inputs $\bar{s}$ weighted according to their joint probability of occurrence
M_k	Number of patterns in cluster k
N_F	Number (user supplied) to eliminate outliers
P, Q	A priori probabilities of signal present and signal absent, respectively
T	Decision threshold
T_F	Threshold (user supplied) for creation of new cluster (number between 0 and 1)
X_i	Set of samples from class i
X^n	Set of n samples

α	False alarm probability
$\bar{p}$	Average missed-detection probability
$\delta \quad (\bar{s}-\bar{o})$	Discrete probability distribution of $\bar{s}$ over space Ω_0 (signal absent region)
ϵ	Probability of rejecting H when it is true (level of test)
$\bar{\theta}$	Parameter vectors determining signal $\bar{s}$
$\hat{\theta}$	Estimate of parameter vector
λ	Lagrange multiplier
$\bar{\mu}_s$	Mean vector of a multivariate probability distribution
$\sigma \quad (\bar{s})$	Joint a priori probability density function over all the points s in signal space
σ^2	Variance of given waveform
τ	Time delay
ϕ	Starting phase
ω_i	Pattern class i
$\omega \quad (\bar{s})$	Probability density of $\bar{s}$ over space Ω_1 (signal plus noise region)
Γ	Observation space
Δ	Decision space
Σ_s	Covariance matrix of a multivariate probability distribution
$(\)^T$	Vector transpose

References

- (1) J. V. DiFranco, W. L. Rubin, "Radar Detection," Prentice-Hall, Inc., 1968
- (2) A. Wald, "Statistical Decision Functions," John Wiley & Sons, 1950
- (3) J. T. Tou, R. C. Gonzalez, "Pattern Recognition Principles," Addison-Wesley, 1974
- (4) J. I. Marcum, "A Statistical Theory of Target Detection by Pulsed Radar," IRE Trans. IT-6: (2), pp. 59-144, April 1960
- (5) P. Swerling, "Probability of Detection for Fluctuating Targets," IRE Trans. IT-6: (2), pp. 269-308, April 1960
- (6) K. S. Fu, "Sequential Methods in Pattern Recognition and Machine Learning," Academic Press, 1968
- (7) R. O. Duda, P. E. Hart, "Pattern Classification and Scene Analysis," John Wiley & Sons, 1973
- (8) K. Abend, "Compound Decision Procedures for Pattern Recognition," Proc. NEC, 22, pp. 777-780, 1966
- (9) J. Raviv, "Decision Making in Markov Chains Applied to the Problem of Pattern Recognition," IEEE Trans., IT-13, pp. 536-551, Oct. 1967
- (10) K. Abend, "Compound Decision Procedures for Unknown Distributions and for Dependent States of Nature," in Pattern Recognition, pp. 207-249, L. Kanal, Ed., Thompson Book Co., Wash. D. C., 1968
- (11) K. Abend, T. J. Harley, L. N. Kanal, "Classification of Binary Random Patterns," IEEE Trans. IT-11, pp. 538-544, Oct. 1965
- (12) E. M. Riseman, R. W. Ehrich, "Contextual Word Recognition Using Binary Diagrams," IEEE Trans, C-20, pp. 397-403, April 1971
- (13) J. R. Slagle, R. C. T. Lee, "Applications of Game Tree Searching Techniques to Sequential Pattern Recognition," Comm. ACM, 14, pp. 103-110, Feb. 1971
- (14) M. Abramowitz, I. A. Stegun (Eds.), "Handbook of Mathematical Functions," NBS Applied Mathematics Series, 55, June 1964

- (15) M.R. Anderberg, "Cluster Analysis for Applications," Academic Press, 1973
- (16) R. Dubes, A.K. Jain, "Clustering Techniques: The User's Dilemma," Pattern Recognition, Vol. 8, pp. 247-260, 1976
- (17) C.T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," IEEE Trans.Comp., Vol. C-20, No. 1, pp. 68-86, Jan. 1971
- (18) R.C. Prim, "Shortest Connection Networks and Some Generalizations," BSTJ, Nov. 1957, pp. 1389-1401
- (19) E.W. Dijkstra, "Some Theorems on Spanning Subtrees of a Graph," Kon. Ned-Akad-Wetensch., Versl. Gewone Vergad, Afd. Natuurk., Series A. Vol. 63, Nov 2; also Indag. Math., Vol. 22, No. 2, pp. 196-199, 1960
- (20) D.L. Duewer, J.R. Koskinen, B.R. Kowalski, "Documentation for Arthur Version 1-8-75," Chemometrics Society Report No. 2, Laboratory for Chemometrics, Dept. of Chemistry, BG-10, University Washington, Seattle, Washington 98195
- (21) G.T. Toussaint, P.M. Sharpe, "An Efficient Method for Estimating the Probability of Misclassification Applied to a Problem in Medical Diagnosis," Comput. Biol. Med. 4, Pergamon Press, pp. 269-278, 1975
- (22) P.E. Hart, "The Condensed Nearest-Neighbor Rule," IEEE Trans. Inf. Theory, IT-14, pp. 515-516, 1968
- (23) W.A. Fordon, J.C. Bezdek, "The Application of Fuzzy Set Theory to Medical Diagnosis," in Advances in Fuzzy Set Theory and Applications, M.M. Gupta, R.K. Ragade, R.R. Yager (Eds.) North Holland Publishing Company, 1979

APPENDIX A - BAYES CLASSIFIER PROGRAM

BAYES

This routine performs an approximate multivariant Bayes rule classification. It also produces the frequency histograms for each feature over each category and over all categories. Since the "true" probability distributions for each feature are presumed to be unknown (if you know them, you may be better off running SPSS and/or BMD), the frequency histograms are used in place of the probability distributions in the Bayes classification. We expect that considerable development of this routine will be required before it is suitable for any but very large data bases.

<u>WORD</u>	<u>DEFAULT</u>	<u>DESCRIPTION</u>
NIN	(I ORIG)	Input unit.
NPNT	0	LE 0 No action GT 0 Histograms produced on line-printer
NPRO	0	LE 0 The <u>a priori</u> probability that a given pattern is a member of a given category is 1.0 for all categories. GT 0 The <u>a priori</u> probability that a given pattern is a member of a given category is (number of patterns in that category) / NPAT.
NRES	0	LE 0 The resolution of the histograms is 1/5 of the number of patterns in the smallest category, rounded to the nearest integer multiple of 10. GT 0 The resolution of the histograms is NRES. (The maximum allowed resolution is NPAT).
LOSS	0	LE 0 The misclassification risk for each category is 1. 0. GT 0 The misclassification risk for each category is user defined. See (1) below.
NPROB	0	LT 0 No classification; only histograms produced.

BAYES, page 2

EQ 0 Default prediction done; individual feature probabilities summed with $\alpha = 0.5, 1.0, 2.0$ and $\sum \ln(\text{prob})$.

GT 0 Prediction summation rules defined by user. See (2) below.

- (1) Misclassification risk: Specify risk associated with each category in the following format:

$i, r\$$ where i = index of the category and r = misclassification risk associated with the i th category.

End risk input with $i=0$. All categories not explicitly defined have a misclassification risk of 1.0.

- (2) Probability summation rules: Specify summation α 's in the following format:

$i, \alpha\$$ where i = dummy index and α = desired summation α ($\alpha = 0$ specifies $\sum \ln(\text{prob})$).

End α input with $i = 0$.

The following example illustrates this option:

To combine the individual feature probabilities using:

$\sum(\text{prob})^1, \sum(\text{prob})^5, \sum(\text{prob})^2, \sum(\text{prob})^{10}, \sum(\text{prob})^1, \sum \ln(\text{prob}) . . .$

1, 0.1\$

1, 0.5\$

1, 2\$

1, 10\$

1, 1\$

1, 0\$

0\$

Prerequisites: Category-type data.

Hints and cautions: Works best on orthogonal features, reduced down to "meaningful" minimum number. See KARLOV and SELECT. Histograms may be obtained on continuous property data.

References: Any numerical statistics text for Bayes Classification Rule.

SECTION II: DEFINITIONS

1. RESOLUTION UNIT

NRES = the number of equal-intervals the features will be divided into.

= user defined

or

$N_{\min} / 5$ (rounded up to nearest even factor of 10)

$N_{\min}$ = the number of patterns in the smallest category

2. MINIMUM

MIN_i = the smallest x_i value in the training set

3. MAXIMUM

MAX_i = the largest x_i value in the training set

4. INCREMENT

$INC_i = (MAX_i - MIN_i) / NRES$

5. PROBABILITY

$PROB_k$ = the a priori probability of a given pattern being a member of category k . (Will either equal 1.0 or $(N_k / NPAT)$, where N_k = number of patterns in category k .)

6. RISK

$RISK_k$ = the risk associated with misclassifying a pattern which is in category k . (Will be 1.0 if not otherwise defined by the user).

7. HISTOGRAMS: NORMALIZED TO THIS SPECTRA MAXIMUM 100 = X

the frequency histogram for the given feature, category has been normalized so the most highly populated interval is plotted full-scale; the actual number of patterns in the full-scale interval is X.

8. HISTOGRAMS: NORMALIZED TO CATEGORY SPECTRA MAXIMUM 100 = Y

the frequency histogram for the given feature, category has been normalized to the most highly populated interval of the NCAT category histograms for the given feature. This normalized plot is presented to allow easy comparison of histograms for all categories of a given feature.

9. HISTOGRAMS: SUMMATION OF ALL CATEGORIES

the frequency histogram for the given feature without regard to possible categories; the over-all feature distribution.

10. HISTOGRAMS: NORMALIZED TO FEATURE SPECTRA MAXIMUM 100 = Z

the over-all feature frequency histogram has been normalized to the most highly populated interval of the NVAR over-all feature histograms. This normalization is presented to allow easy comparison of histograms for all features.

11. PROBABILITY CALCULATED WITH SUM OF . . .

the Bayes Theorem probability that a given pattern is a member of category k , using feature i and its associated probability distribution, is given by:

$$P_j[X_{j,k} \mid x_{i,j}] = \frac{(PROB_k)(RISK_k) P[x_{i,j} \mid X_{j,k}]}{NCAT \sum_{n=1} (PROB_n)(RISK_n) P[x_{i,j} \mid X_{j,n}]}$$

11. continued

$P[x_{i,j} | X_{j,n}]$ = the value of the probability distribution for (feature_j, category_n) at the value of $x_{i,j}$

given that the categories are mutually exclusive and that the probability of a pattern being a member of some category of the training set is 1.0.

The probability distributions as represented by the frequency histograms are not, unfortunately, continuous (there may be completely empty intervals surrounded by high-frequency regions.) This makes for difficulties in attempting a straightforward multiplicative multifeature probability estimate. We have chosen to combine the single feature probabilities using less-sensitive (but empirical) rules:

1. FEATURE PROBABILITIES RAISED TO THE α POWER:

$$P_{\text{total}}[X_k | x_i] = \sum_{j=1}^{\text{NVAR}} P_{j,k} [x_{i,j}]^{\alpha}$$

2. LN FEATURE PROBABILITIES (essentially a multiplicative combination).

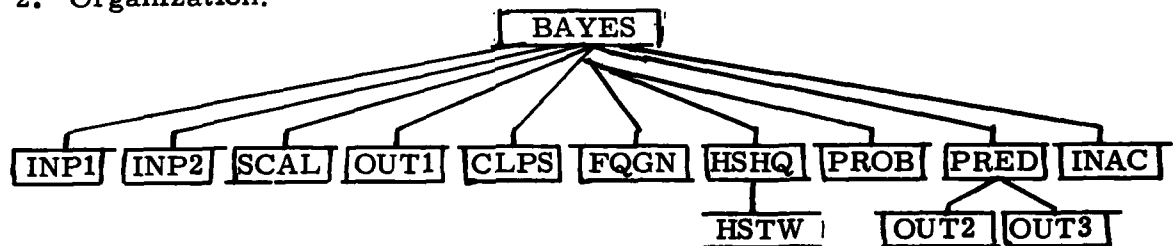
$$P_{\text{total}}[X_k | x_i] = \sum_{j=1}^{\text{NVAR}} \ln(P_{j,k} [x_{i,j}])$$

SECTION III: IMPLEMENTATION

1. Subroutines:

BAYES: driver
INP1BA: input
INP2BA: input, risk and α arrays
SCALBA: scale parameters
OUT1BA: output, working parameters
CLPSBA: continuous feature values to discrete integers
FQGNBA: creates and stores frequency distributions
HSHQBA: driver for histograms
HSTWBA: line-printer histograms
PROBBA: Bayes-rule probabilities for each category
PREDBA: multi-feature probability summations and out
OUT2BA: output, pattern classification results
OUT3BA: output, result summary
INACBA: interactive terminal driver

2. Organization:



APPENDIX B - FORGY/JANCEY PROGRAM - SQUARED-ERROR PROGRAM

Subroutine EXEC

```

C      SUBROUTINE EXEC(X,LIMIT)
C
C      THIS SUBROUTINE READS PARAMETERS, COMPUTES STORAGE AND CALLS MAJOR
C      PROGRAM SEGMENTS NEEDED FOR A NON-HIERARCHICAL CLUSTERING JOB USING
C      ONE OF THE METHODS PROGRAMMED AS A VERSION OF SUBROUTINE "KMEAN".
C
C      EVERY JOB REQUIRES THREE USER SUPPLIED DECK SEGMENTS.
C
C      1. PROGRAM "DRIVER" PERFORMS THE FOLLOWING TASKS.
C          A. ASSIGNS INPUT/OUTPUT UNITS.
C          B. ESTABLISHES THE DIMENSION OF THE "X" ARRAY AND SETS THIS
C             DIMENSION TO "LIMIT".
C          C. CALLS SUBROUTINE "EXEC".
C
C      THE FOLLOWING EXAMPLE WILL SUFFICE IN MOST CASES.
C
C          PROGRAM DRIVER(INPUT,OUTPUT,PUNCH,TAPES=INPUT,TAPE6=OUTPUT,
C          ATAPE7=PUNCH,TAPE1,TAPE2)
C          DIMENSION X(5000)
C          LIMIT=5000
C          CALL EXEC(X,LIMIT)
C          END
C
C      2. SUBROUTINE "USER" IS EMPLOYED TO READ THE COMPLETE SET OF SCORES
C      ON THE VARIABLES FOR ONE DATA UNIT. THE FOLLOWING EXAMPLE
C      ILLUSTRATES VARIOUS POSSIBILITIES FOR MERGING FILES AND
C      TRANSFORMING VARIABLES AS THEY ARE READ.
C
C          SUBROUTINE USER(X)
C          DIMENSION X(8)
C          READ(1,100) X(7),Y
C          READ(2) (X(I),I=1,6)
C          READ(5,200) X(8),Z
C          X(3)=.5*X(3)
C          X(7)=3.6*X(7)
C          X(8)=.4*X(8)+.35*Y+.25*Z*X(8)
C          RETURN
C 100 FORMAT(2F11.3)
C 200 FORMAT(F8.1,F6.3)
C          END
C
C      3. FUNCTION "DIST" COMPUTES THE DISTANCE BETWEEN TWO DATA UNITS OR
C      BETWEEN A DATA UNIT AND A CLUSTER CENTROID. THE USER CAN SPECIFY
C      ANY DESIRED DISTANCE FUNCTION AND WEIGHT THE VARIABLES IN ANY
C      MANNER. THE FOLLOWING EXAMPLE ILLUSTRATES A WEIGHTED SQUARED
C      EUCLIDEAN DISTANCE BETWEEN TWO DATA UNITS DENOTED AS X AND Y.
C      THE PROBLEM INVOLVES 8 VARIABLES AND THE WEIGHTS ARE IN THE
C      "W" ARRAY.
C
C          FUNCTION DIST(X,Y)
C          DIMENSION X(1),Y(1),W(8)
C          DATA (W(I),I=1,8)/3*1.,3.,4.5,2.,2*1.,/
C          DIST=0.
C          DO 10 I=1,8
C          10 DIST=DIST+W(I)*((X(I)-Y(I))**2)
C          RETURN
C          END
C
C      NOTE THAT SCALING AND TRANSFORMATION OF VARIABLES CAN BE
C      ACCOMPLISHED EITHER IN SUBROUTINE "USER" OR IN SUBROUTINE "DIST".
C

```

```

C-----
C INPUT SPECIFICATIONS
C CARD 1 TITLE
C CARD 2 PARAMETER CARD
C COLS 1-5 NC=NUMBER OF ENTITIES (DATA UNITS)
C COLS 6-10 NV=NUMBER OF VARIABLES
C COLS 11-15 NC=NUMBER OF CLUSTERS
C COLS 16-20 NTIN=INPUT UNIT FOR THE DATA SET
C NTIN=5, CARD READER
C NTIN=6,5, TAPE OR DISK FILE
C COLS 21-25 NTOUT=OUTPUT UNIT FOR SAVING CLUSTER MEMBERSHIP LISTS
C NTOUT=7, CARD PUNCH
C NTOUT=6, DO NOT SAVE MEMBERSHIP LISTS
C COLS 26-30 MINREL=TERMINATION PARAMETER. CLUSTERING ENDS WHEN A
C CYCLE THROUGH THE DATA SET RESULTS IN *MINREL*
C OR FEWER CHANGES IN CLUSTER MEMBERSHIPS
C MINREL=6, ITERATE TO COMPLETE CONVERGENCE
C-----

C COLS 31-35 IPART=INITIAL PARTITION PARAMETER
C IPART=1, SEED POINTS ARE SELECTED FROM THE DATA UNITS.
C READ THE SEQUENCE NUMBERS FOR THE CHOSEN DATA
C UNITS FROM CARD(5) 3 IN 2014 FORMAT. IF THE
C DATA SET IS NOT STORED IN CORE, THE LIST OF
C OF SEQUENCE NUMBERS MUST BE IN ASCENDING ORDER
C IPART=2, THE DATA UNITS ARE GROUPED INTO AN INITIAL
C PARTITION IN THE INPUT SEQUENCE WITH THE
C FIRST *NUMBER(1)* IN CLUSTER 1, THE NEXT
C *NUMBER(2)* IN CLUSTER 2 ETC. READ THE
C *NUMBER* ARRAY FROM CARD(5) 3 IN 2014 FORMAT.
C IPART=3, THE SCORE VECTORS FOR THE SEED POINTS ARE
C READ FROM CARD(5) 4 IN FORMAT *FMT* WHICH IS
C READ FROM CARD 3.
C COLS 36-40 METHOD=PARAMETER FOR CHOOSING THE ALGORITHM IN ONE
C VERSION OF SUBROUTINE *KMEAN*.
C METHOD=1, JANCEY ALGORITHM
C METHOD=NE,1, FORGY ALGORITHM
C-----
C ***CARDS 3 AND 4 ARE READ IN SUBROUTINE *KMEAN* ACCORDING TO THE
C ***PROCEDURE SPECIFIED BY THE CHOSEN VALUE OF *IPART*. NOTE THAT THE
C ***BASIC K-MEANS METHOD OF MACQUEEN SIMPLY USES THE FIRST *NC* DATA
C ***UNITS AS CLUSTER SEED POINTS AND THEREFORE IGNORES THE *IPART*
C ***PARAMETER.
C-----
C STORAGE ALLOCATIONS IN THE *X* ARRAY
C X(1) TO X(2)-1 NC*NV WORDS--STORAGE OF THE CENTR ARRAY
C X(2) TO X(3)-1 NC WORDS--STORAGE OF THE NUMBR ARRAY
C X(3) TO X(4)-1 NC WORDS--STORAGE OF THE MEMBR ARRAY
C X(4) TO X(5)-1 NC*NV WORDS--STORAGE OF THE TOTAL ARRAY
C X(5) TO X(6) NV OR NE*NV WORDS--STORAGE OF THE DATA ARRAY
C X(6) TO X(7) NE WORDS--STORAGE OF THE LIST ARRAY IN *RESULT*
C
C DIMENSION X(1),TITLE(20)
C READ(5,1000) TITLE
C READ(5,1100) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD
C WRITE(6,2000) TITLE
C WRITE(6,2100) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD
C N1=1
C N2=N1+NC*NV
C N3=N2+NC
C N4=N3+NE
C N5=N4+NC*NV
C *N6* MAY BE INCREASED IN *KMEAN*.
C N6=N5+NV-1
C N7=N6+NE-1
C N8=N7
C IF (N7.GT.MAX) MAX=N7
C WRITE(6,2200) MAX,LIMIT
C IF (MAX.GT.LIMIT) STOP
C CALL KMEAN(X(1),X(2),X(3),X(4),X(5),N5,NE,NV,NC,NTIN,MINREL,
C IPART,METHOD,LIMIT)
C CALL RESULT(X(1),X(2),X(3),X(4),TITLE,NE,NV,NC,NTOUT)
C RETURN
1000 FORMAT(20A4)
1100 FORMAT(I5)
2000 FORMAT(I5I,20A4)
2100 FORMAT(5HNE =,I4,/,5H NV =,I4,/,5H NC =,I4,/,5H NTIN =,I4,/,
C 5H NTOUT =,I4,/,5H MINREL =,I4,/,5H IPART =,I4,/,5H METHOD =,I4)
2200 FORMAT(19HREQUIRED STORAGE =,I5,6H WORDS,/,
C 4 19HALLOCATED STORAGE =,I5,6H WORDS)
C PND

```

Subroutine RESULT

```

SUBROUTINE RESULT(CENTR,NUMBR,MEMBR,LIST,TITLE,NE,NV,NC,NTOUT)
C THIS SUBROUTINE PRINTS THE RESULTS FROM A CLUSTERING JOB BASED
C ON ANY VERSION OF SUBROUTINE *KMEAN*.
C
C   DIMENSION CENTR(1),NUMBR(1),MEMBR(1),LIST(1),TITLE(20)
C
C AS A CONTINGENCY PRECAUTION WRITE OUT THE RAW MEMBERSHIP LIST.
WRITE(6,2000) TITLE
WRITE(6,2100) (MEMBR(K),K=1,NE)
WRITE(6,2200) (NUMBR(J),J=1,NC)
C INVERT THE *MEMBR* ARRAY AND PUT THE RESULT IN THE *LIST* ARRAY.
C FIRST REVISE THE *NUMBR* ARRAY TO CONTAIN START POINTS IN THE
C *LIST* ARRAY FOR EACH CLUSTER
NUMBR(NC)=NE-NUMBR(NC)+1
JJ=NC
JJ=JJ-1
DO 10 J=2,NC
NUMBR(JJ)=NUMBR(JJ)-NUMBR(JJ-1)
JJ=JJ-1
10 JJ=JJ-1
C BUILD *LIST* ARRAY
DO 20 K=1,NE
MEMBRK=MEMBR(K)
NJ=NUMBR(MEMBRK)
LIST(NJ)=K
NUMBR(MEMBRK)=NUMBR(MEMBRK)+1
20 CONTINUE
C SAVE THE SORTED MEMBERSHIP LIST IF DESIRED
IF(INTOUT.LE.0) GO TO 30
WRITE(INTOUT,3000) TITLE
WRITE(INTOUT,3100) (LIST(K),K=1,NE)
C RESTORE THE *NUMBR* ARRAY
30 JJ=NC
DO 40 J=2,NC
NUMBR(JJ)=NUMBR(JJ)-NUMBR(JJ-1)
JJ=JJ-1
40 NUMBR(1)=NUMBR(1)-1
C PRINT RESULTS FOR EACH CLUSTER
WRITE(6,2000) TITLE
K1=1
DO 50 J=1,NC
WRITE(6,2300) J,NUMBR(J)
J1=(J-1)*NV
WRITE(6,2400) (CENTR(J1+I),I=1,NV)
K2=K1+NUMBR(J)-1
WRITE(6,2500) (LIST(K),K=K1,K2)
K1=K2+1
50 CONTINUE
RETURN
2000 FORMAT(1H1,20A4)
2100 FORMAT(20HRAW MEMBERSHIP LIST,/, (1X,25I5))
2200 FORMAT(14H0CLUSTER SIZES,/, (1X,25I5))
2300 FORMAT(8H0CLUSTER,13,9H CONTAINS,15,11H DATA UNITS)
2400 FORMAT(21H0CENTROID COORDINATES,/, (1X,10E12.4))
2500 FORMAT(16H0MEMBERSHIP LIST,/, (1X,25I5))
3000 FORMAT(20A4)
3100 FORMAT(20I4)
END

```

Subroutine KMEAN

VERSION 1

```

      SUBROUTINE KMEAN(CENTR,NUMBR,MEMBR,TOTAL,DATA,NS,NE,NV,NC,NTIN,
      AMINREL,[PART,METHOD,LIMIT])
C
C-----
C  VERSION 1.  THE DATA SET IS STORED IN CENTRAL MEMORY.
C-----
C
C  THIS SUBROUTINE ITERATIVELY SORTS *NE* DATA UNITS INTO *NC* CLUSTERS
C  USING THE ALGORITHM OF (METHOD,NE,1)
C
C  FORGY, E.W., CLUSTER ANALYSIS OF MULTIVARIATE DATA: EFFICIENCY
C  VERSUS INTERPRETABILITY OF CLASSIFICATIONS, PAPER PRESENTED AT THE
C  BIOMETRIC SOCIETY (WNAF) MEETINGS, RIVERSIDE, CALIFORNIA, JUNE
C  1965. ABSTRACT IN BIOMETRICS, VOLUME 21, NUMBER 3, P 768.
C
C  OR THE ALGORITHM OF (METHOD=1)
C
C  JANCEY, R.C., MULTIDIMENSIONAL GROUP ANALYSIS, AUSTRALIAN JOURNAL
C  OF ROTARY, VOLUME 14, NUMBER 1, APRIL 1966, PP 127-130.
C
C  CENTR(INV*(J-1)+1)=SCORE ON I-TH VARIABLE FOR J-TH CLUSTER CENTROID
C  TOTAL(INV*(J-1)+1)=TOTAL SCORE ON I-TH VARIABLE FOR DATA UNITS THUS
C  FAR ALLOCATED TO THE J-TH CLUSTER
C  NUMBR(J)=NUMBER OF DATA UNITS THUS FAR ALLOCATED TO THE J-TH CLUSTER
C  MEMBR(K)=CLUSTER TO WHICH THE K-TH DATA UNIT CURRENTLY BELONGS
C  DATA(INV*(K-1)+1)=SCORE ON I-TH VARIABLE FOR K-TH DATA UNIT
C
      DIMENSION CENTR(1),TOTAL(1),NUMBR(1),MEMBR(1),DATA(1),FMT(20)
      A,NAME(4)
      DATA (NAME(I),I=1,4)/4H  F,4MORGY,4H  JA,4MNCZY/
      I=1
      IF(METHOD.EQ.1) I=3
      WRITE(6,2000) NAME(1),NAME(1+1)
C  CHECK FOR SUFFICIENT STORAGE
      N6=NS*NE*NV-1
      WRITE(6,2100) N6,LIMIT
      IF(N6.GT.LIMIT) STOP
C  ESTABLISH INITIAL PARTITION
      IF([PART,NE,3]) GO TO 20
C  SEED POINTS ARE READ DIRECTLY FROM CARDS
      READ (5,1000) FMT
      WRITE(6,2200) FMT
      WRITE(6,2300)
      J1=0
      DO 10 J=1,NC
      READ(5,FMT) (CENTR(J),I=1,NV)
      WRITE(6,2400) (CENTR(J),I=1,NV)
10    J1=J1+NV
      GO TO 30
C  [PART=1 OR 2
20    WRITE(6,2500) [PART
      READ(5,1100) (NUMBR(J),J=1,NC)
      WRITE(6,2600) (NUMBR(J),J=1,NC)
C  READ THE DATA SET INTO CENTRAL MEMORY
30    K1=1
      DO 40 K=1,NE
      CALL USER (DATA(K))

```

```

40 K1=K1+NV
   IF (IPART.EQ.3) GO TO 100
C IF *IPART* IS 1 OR 2 SET UP THE SEED POINTS
   IF (IPART.EQ.2) GO TO 60
C IPART=1. THE DATA UNIT WITH SEQUENCE NUMBER *NUMBR(J)* IS USED AS
C THE J-TH SEED POINT
   DO 50 J=1,NC
     NJ=(NUMBR(J)-1)*NV
     J1=(J-1)*NV
     DO 50 I=1,NV
       CENTR(J1+I)=DATA(NJ+I)
50 CONTINUE
   GO TO 100
C IPART=2. THE DATA UNITS ARE GROUPED INTO CLUSTERS WITH THE J-TH
C CLUSTER HAVING *NUMBR(J)* MEMBERS.
60 K=0
   J1=NV
C ACCUMULATE THE TOTAL SCORE ON EACH VARIABLE FOR EACH CLUSTER
   DO 80 J=1,NC
     NJ=NUMBR(J)
     J1=J1+NV
     DO 70 I=1,NV
       TOTAL(J1+I)=0.
70 DO 80 KJ=1,NJ
     K=K+1
     MEMBR(K)=J
     K1=(K-1)*NV
     DO 80 I=1,NV
       J2=J1+I
       TOTAL(J2)=TOTAL(J2)+DATA(K1+I)
80 CONTINUE
C COMPUTE THE CENTROIDS
   J1=0
   DO 90 J=1,NC
     DO 90 I=1,NV
       J1=J1+1
       CENTR(J1)=TOTAL(J1)/NUMBR(J)
90 CONTINUE
   GO TO 115
C INITIALIZE ARRAYS
100 DO 110 K=1,NE
110 MEMBR(K)=0
115 NPASS=1
C BEGINNING OF MAIN LOOP
120 J1=0
   DO 130 J=1,NC
     NUMBR(J)=0
     DO 130 I=1,NV
       J1=J1+1
130 TOTAL(J1)=0.
     MOVES=0
     TDIST=0
C ALLOCATE EACH DATA UNIT TO THE NEAREST CLUSTER CENTROID
     K1=0
     DO 160 K=1,NE
       K2=K1+1
       J2=1
C COMPUTE DISTANCE TO FIRST CLUSTER CENTROID
       DREF=DIST(DATA(K2),CENTR(J2))
       JREF=1
C TEST DISTANCES TO REMAINING CLUSTER CENTROIDS
       DO 140 J=2,NC
         J2=J2+NV
         DTEST=DIST(DATA(K2),CENTR(J2))
         IF (DTEST.GE.DREF) GO TO 140
         DREF=DTEST
         JREF=J

```

```

140 CONTINUE
C ALLOCATE DATA UNIT *K* TO CLUSTER *JREF*
  NUMBR(JREF)=NUMBR(JREF)+1
  TOIST=TOIST+OREF
  IF(JREF.EQ.MEMBR(K)) GO TO 150
C THE DATA UNIT CHANGES ITS MEMBERSHIP
  MOVES=MOVES+1
  MEMBR(K)=JREF
150 J1=(JREF-1)*NV
  DO 160 I=1,NV
    J1=J1+1
    K1=K+1
    TOTAL(J1)=TOTAL(J1)+DATA(K1)
160 CONTINUE
C ALL DATA UNITS ALLOCATED. TEST FOR CONVERGENCE
  WRITE(6,2700) MOVES,NPASS,TOIST
  NPASS=NPASS+1
  JREF=0
  IF(MOVES.GT.MINREL) GO TO 185
  IF(METHOD.NE.1.AND.MOVES.EQ.0) RETURN
  JREF=1
C COMPUTE TRUE CLUSTER CENTROIDS--FORBY UPDATE
170 J1=0
  DO 180 J=1,NC
    DO 180 I=1,NV
      J1=J1+1
180 CENTR(J1)=TOTAL(J1)/NUMBR(J)
  IF(JREF.EQ.1) RETURN
  GO TO 120
185 IF(METHOD.NE.1) GO TO 170
C JANCEY UPDATE
190 J1=0
  DO 200 J=1,NC
    DO 200 I=1,NV
      J1=J1+1
200 CENTR(J1)=2.*TOTAL(J1)/NUMBR(J)-CENTR(J1)
  GO TO 120
1000 FORMAT(20A4)
1100 FORMAT(20I4)
2000 FORMAT(1H0,2A4, 53H METHOD OF CLUSTER ANALYSIS. DATA SET STORED I
  AN CORE)
2100 FORMAT(19H0REQUIRED STORAGE =.15,6M WORDS./,
  A 19H0ALLOTTED STORAGE =.15,6M WORDS)
2200 FORMAT(7H0FORMAT,20A4)
2300 FORMAT( 43H1INITIAL CLUSTER CENTERS READ IN AS FOLLOWS///)
2400 FORMAT(1X,10E12,4)
2500 FORMAT( 9H1 IPART =.12, 30H, NUMBR ARRAY READ AS FOLLOWS///)
2600 FORMAT(1X,10I7)
2700 FORMAT(1H0,15,37M DATA UNITS MOVED ON ITERATION NUMBER,13,/,
  A37M SUMMED DEVIATIONS ABOUT SEED POINTS =.E16,0)
  END

```

VERSION 2

SUBROUTINE KMEAN(CENTR,NUMBR,MEMBR,TOTAL,DATA,NS,NE,NV,NC,NTIN,
AMINREL,IPART,METHOD,LIMIT)

```

C
C -----
C VERSION 2. THE DATA SET IS STORED ON A TAPE OR DISK FILE WHICH IS
C REWOUND AND READ IN ITS ENTIRETY FOR EACH CYCLE.
C -----
C

```

APPENDIX C - HIERARCHICAL CLUSTERING PROGRAM

HIER

This routine produces a "dendrogram" which describes the hierarchical clustering (sometimes known as "Q-mode clustering") of the NPAT training set patterns. The dendrogram connects groups of patterns at levels of similarity. It may be used to group the patterns into a given number of clusters as well as to indicate how many clusters there are at a given similarity level. Similarity is defined from the interpattern distances.

<u>WORD</u>	<u>DEFAULT</u>	<u>DESCRIPTION</u>
NIN	(I ORIG)	Input unit.
IWAIT	0	LE 0 Every pattern is given equal weight in determining the linkage levels, regardless of the size of the group of which it is a member. GT 0 Every group is given equal weight in determining the linkage levels, regardless of how many patterns are contained in the group.
IPULL	0	LE 0 The number of sections in which the dendrogram is printed is determined by the routine. GT 0 The dendrogram is printed in IPULL sections (maximum of 3).

Prerequisites: The distance matrix must be present on NIN. See DIST.

Hints and Cautions: Only the first NPAT patterns will be clustered. Be sure that you've defined the "training set" to include all patterns you are interested in clustering. (The algorithm implemented in this program uses some computational "tricks" to reduce run time. The clusters will be nearly the same as those formed by truly hierarchical clustering, but the levels of similarity may differ.)

SECTION II: DEFINITIONS

1. SIMILARITY

$$S_{i,j} = 1.0 - D_{i,j} / \text{DMAX}$$

DMAX = the largest $D_{i,j}$ in the distance matrix

2. EQUAL SAMPLE WEIGHT PAIR-GROUP METHOD OF CLUSTERING

$$S_{\text{new}} = \frac{(\text{NUM}_{1,\text{old}})(S_{1,\text{old}}) + (\text{NUM}_{2,\text{old}})(S_{2,\text{old}})}{\text{NUM}_{1,\text{old}} + \text{NUM}_{2,\text{old}}}$$

$\text{NUM}_{i,\text{old}}$ = number of patterns grouped into
cluster represented by $S_{i,\text{old}}$

$S_{i,\text{old}}$ = groups chosen to be clustered this cycle

3. EQUAL GROUP WEIGHT PAIR-GROUP METHOD OF CLUSTERING

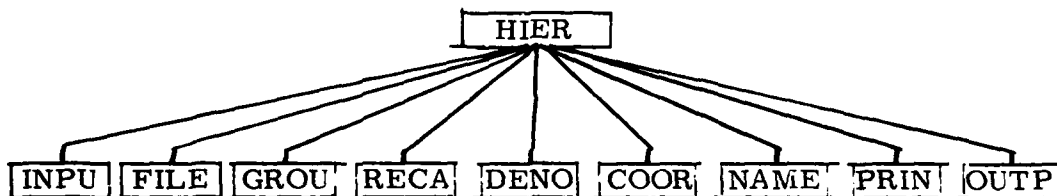
$$S_{\text{new}} = \frac{S_{1,\text{old}} + S_{2,\text{old}}}{2}$$

SECTION III: IMPLEMENTATION

1. Subroutines:

HIER:	driver
INPUHI:	input
FILEHI:	file initialization
GROUHI:	clustering
RECAHI:	distance recalculation
DENOHI:	dendrogram formation
COORHI:	lineprinter coordinates for dendrogram
NAMEHI:	pattern identifiers read into arrays
PRINHI:	output, dendrogram
OUTPHI:	output

2. Organization:



APPENDIX D - MINIMAL SPANNING TREE PROGRAM - GRAPH-THEORETIC METHOD

TREE

This routine generates a minimal spanning tree over the training set patterns. The spanning tree is then evaluated ("pruned") for self-consistent clusters of patterns. The algorithm used is that of Prim and Dijkstra, as implemented by Whitney. The original program was written by Dr. Rex Page, Department of Computer Sciences, Colorado State University.

<u>WORD</u>	<u>DEFAULT</u>	<u>DESCRIPTION</u>
NIN	(I ORIG)	Input unit.
NPNT	0	LE 0 No action.
		GT 0 All nodes of the spanning tree are listed as the tree is constructed. If a diagram of the tree is desired, this information is necessary.
NIT	0	LE 0 The spanning tree will be pruned once, with D=3, FACTOR=2, and SPREAD=0.0.
		GT 0 The spanning tree will be pruned according to user definition of D, FACTOR and SPREAD. (See 1 below)

(1) Pruning parameters . . . Specify the evaluation parameters with the following format:

D, FACTOR, SPREAD\$

where D = the number of edges allowed between patterns for patterns to be "nearby" one another.

FACTOR = Factor times the average length of "nearby" edges for edge to be inconsistent.

SPREAD = Factor times standard deviation of "nearby" edge lengths for edge to be inconsistent.

Prerequisites: None

Hints and Cautions: For unbiased clustering, use autoscaled data.

References: Harry C. Andrews, INTRODUCTION TO MATHEMATICAL TECHNIQUES IN PATTERN RECOGNITION, Wiley-Interscience, New York, 1972.

SECTION II: DEFINITIONS

1. NODE

Pattern.

2. NEIGHBORS

The patterns linked to the given pattern during the construction of the minimal spanning tree.

3. DISTANCE

The Euclidean distance between the given pattern and its given neighbor.

4. CLUSTER (N)

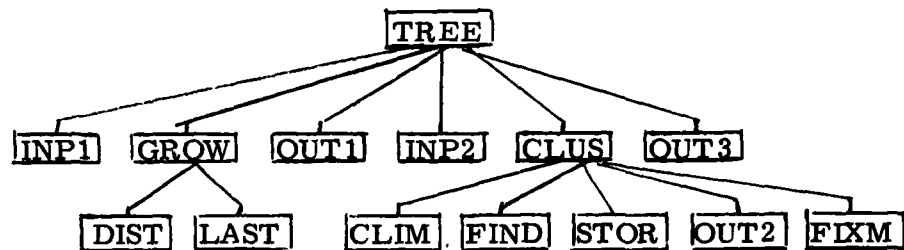
The nth cluster found, searching from "trunk" out, using the given pruning parameters.

SECTION III: IMPLEMENTATION

1. Subroutines:

TREE: driver
INP1TR: input
GROWTR: formation of minimal spanning tree
DISTTR: two-pattern distance calculation
LASTTR: pointer to last-found node
OUT1TR: output, optional intermediate
INP2TR: input, pruning parameters
CLUSTR: prunes tree
CLIMTR: tree search (in conjunction with CLUSTER)
FINDTR: locates node
STORTR: stores cluster
OUT2TR: output, detailed cluster
FIXMTR: puts termination flag into cluster array
OUT3TR: output, compact cluster

2. Organization:



APPENDIX E - "THE COMPARISON OF A BAYESIAN CLASSIFIER AND
A K-NEAREST NEIGHBOR STATISTICAL PATTERN
RECOGNITION TECHNIQUE AS APPLIED TO RADAR
GROUND CLUTTER,"
M.S. Thesis - A. A. Fraser

ABSTRACT

This paper presents a comparison of the Bayes and the k-Nearest Neighbor statistical pattern recognition algorithms. The first half of this presentation is a detailed analysis of both techniques and it also gives a description of the actual algorithms used.

Simulated radar ground clutter information was available for analysis. A description of the data subject to analysis is also presented.

The error rate of these classification algorithms was the chief criterion used for the evaluation of performance. The second half of the paper discusses the various error evaluating techniques that are feasible for evaluation of the performance of the algorithms. Because of economics, time consideration, and other factors, the π -method⁷ was chosen to measure error rate.

Results showed that the nonparametric Nearest Neighbor technique gives a much smaller error rate than the parametric Bayes technique for the given data type. The results are justified in the conclusion.

TABLE OF CONTENTS

1.0	Introduction
1.1	Fundamentals of Statistical Decision Theory
2.0	Introduction to Parametric Classification
2.1	Discriminant Functions
2.2	Parametric Classification
2.3	Classical Techniques
2.4	Bayes Algorithm Used
3.0	Nonparametric Classification
3.1	Nearest Neighbor Pattern Classification
3.2	k-NN Algorithm Used
4.0	Data
5.0	Error Analysis
5.1	Methods for Evaluating the Probability of Misclassification
5.2	Error Estimation Techniques
5.3	Performance Measure Used
6.0	Summary
6.1	Conclusion
6.2	Recommendations for Future Work
	Sub-Appendix E-A - Bayes Implementation
	Sub-Appendix E-B - k-NN Implementation
	Sub-Appendix E-C - $\hat{P}_{ei}[\pi]$ (x, y, amplitude)
	References

LIST OF FIGURES

	<u>Page</u>
1.0.1 The Conceptualized Pattern Recognition Problem	E 3
1.0.2 A Possible Portion of Feature Space	E 5
1.1.1 Reception as a Decision Problem	E 9
2.1.0 A Redundant Decision Surface	E 12
2.1.1 A Typical Classifier	E 13
2.1.2 A Piecewise Linear Discriminant Surface	E 17
2.2.1 Data Sets Having Identical Second-Order Statistics	E 21
2.2.2 Normal Distributions	E 23
2.3.1 Parameter Estimation as a Decision Problem	E 25
2.3.2 Bayes Classifier and a Symmetric Loss Function	E 29
4.0.1 Pseudo-Color Photo of Amplitude Data	E 43
4.0.2 Pseudo-Color Photo of Doppler Data	E 43
4.0.3 Execution Time Versus Number of Samples	E 44a
5.0.1 Bounds on the Error-Rate for the k-Nearest Neighbor Rule	E 50
5.2.1 Error Curves	E 61
6.0.1 Execution Time	E 63c
6.0.2 Bayes (**.5) and 1-NN Error Map	E 63b

LIST OF TABLES

	<u>Page</u>
3.2.1 Some Distance Functions	E 39
4.0.1 Amplitude Range for the Various Features	E 43
4.0.2 Doppler Measure for the Various Features	E 43
5.3.1 Preliminary Results for the Probability of Error (x, y, amplitude, Doppler)	E 56
5.3.2 Layout for the Various Groups of Data Used in Experimentation	E 58
5.3.3 Error Rates Determined From Tests (x, y, Amplitude)	E 59
6.0.1 Memory Allocation to the Two Algorithms	E 63a

Glossary

<u>Symbol</u>	<u>Definition</u>	<u>Units</u>	<u>Page</u>
$C(S_k/S_i)$	The cost of deciding class S_i when S_k is actually present.	--	E 26
$D(\bar{d}/\bar{v})$	$\text{Pr} \{ \text{Making decision } \bar{d} \text{ given } \bar{v} \}$	--	E 7
$E \{ \}$	Expectation operator	--	E 20
$L(x, S_k)$	The average loss associated with class S_k given pattern x .	--	E 26
M_k	The number of samples in class k	--	E 15
$N(\mu, \phi)$	Representation of a normal distribution	--	E 22
Pr	Probability operator	--	
P^*	The Bayesian error rate	--	E 46
$P^*(e/x)$	Error associated with classifying pattern x .	--	E 46
P_s	The probability that " x " falls within hypersphere S .	--	E 46
$P_n(e/x, x_n')$	Complementary probability of error	--	E 47
$\hat{P}_e[H]$	Holdout method error rate	--	E 52
$\hat{P}_e[R]$	Redistribution method error rate	--	E 51
$\hat{P}_e[U]$	U-method error rate	--	E 54
$\hat{P}_e[\pi]$	π -method error rate	--	E 54
$R(S_k)$	Risk associated with deciding class k	--	E 29
$S(\hat{g})$	P_r misclassification	--	E 33

Glossary (continued)

<u>Symbol</u>	<u>Definition</u>	<u>Units</u>	<u>Page</u>
S_k	Pattern class k	--	E 10
TR	Training set	--	E 54
TS	Test set	--	E 54
W	Scalar weight	--	E 14
W^t	Weight vector transpose	--	E 14
W_{N+1}	Augments the weight vector	--	E 14
$X \cdot x(X \cdot x \cdot H)^n$	Euclidean product	--	E 32
$Y_m^{(k)}$	Pattern on m of class k	--	E 15
$\bar{d}$	Vector in decision space	--	E 7
$d(x, y)$	Distance between x and y	--	E 16
$d(x)$	Decision function	--	E 16
$f_i(x)$	Real single valued function of x.	--	E 16
g_a	Class estimator	--	E 32
$g_k(x)$	Discriminant function of class k	--	E 11
k-NN	k-Nearest Neighbor	--	
$\rho(x, S_k)$	Modified conditional average loss	--	E 28
max	Maximum	--	E 16
min	Minimum	--	E 16
$\bar{n}$	Vector in noise and clutter space	--	E 7

Glossary (continued)

<u>Symbol</u>	<u>Definition</u>	<u>Units</u>	<u>Page</u>
$p(\bar{v}/\bar{s})$	Pr $\bar{v}$ given $\bar{s}$	--	E 7
$p(x, S_k)$	Pr x belongs to class k	--	E 22
$\bar{s}$	Vectors in signal space	--	E 7
$\bar{v}$	Vectors in observation space	--	E 7
X_k	Pattern k	--	E 15
Ω	Signal space	--	E 7
Δ	Decision space	--	E 7
Γ	Observation space	--	E 7
ϵ	Is an element of	--	E 15
$\forall$	For all	--	E 15
μ	Sample mean	--	E 20
$[\phi]$	Covariance matrix	--	E 20
λ	Likelihood ratio	--	E 28
$\delta(i-k)$	Kronecker delta function	--	E 27
θ_i	Random variable $i=1, \dots, C$ where C is the number of classes	--	E 33

AD-A111 893

MICHIGAN TECHNOLOGICAL UNIV Houghton

F/S 17/9

STATISTICAL PATTERN RECOGNITION TECHNIQUES AS APPLIED TO RADAR --ETC(U)

DEC 81 W A FORDON, A A FRASER

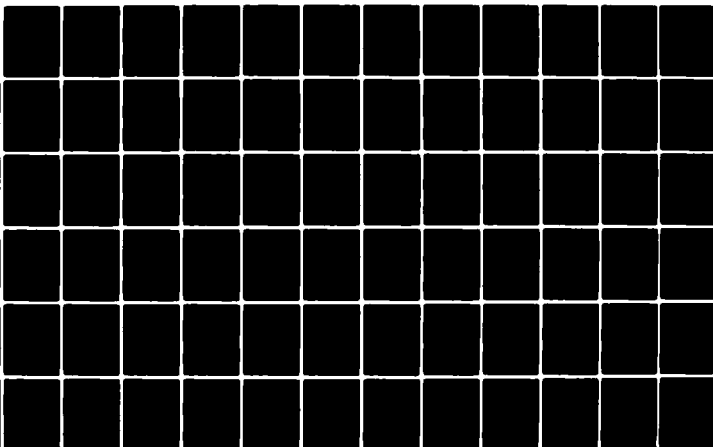
F30602-78-C-0102

ML

UNCLASSIFIED

RADC-TR-81-61

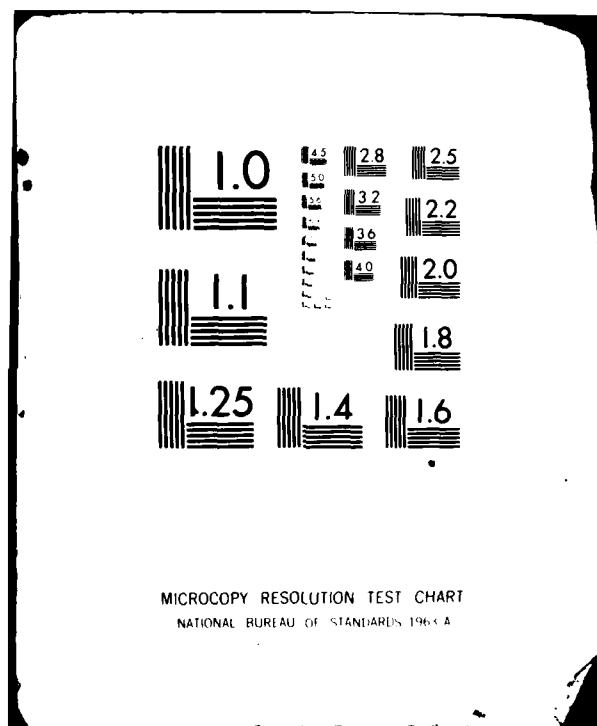
212
201/8/81



END
DATE
FILMED

4-82

NTIC



1.0 Introduction

Detailed investigation in the area of statistical pattern recognition was motivated in this study by the necessity to use mathematical classification algorithms to characterize ground clutter and noise. The ultimate goal is to eventually be able to distinguish between the presence or absence of an object in a background of ground clutter and noise. In general, statistical pattern recognition enhances the capability to develop a machine that will imitate man's perceptive ability. Research towards this end has been carried out in the areas of artificial intelligence, interactive graphics, computer-aided design, and many others. There are some well developed theories behind statistical pattern recognition^{1, 5, 9, 11, 12}. They evolved from all the fields previously presented.

Statistical pattern recognition is the study of mathematical techniques to build machines to aid human perception. The use of computers in this area has its advantage in the fact that it is capable of handling large sets of data.

Pattern recognition's function could be conceptualized in three different states or spaces as indicated in Figure 1.0.1, pattern space, feature space and classification space^{5, 9}. The physical world is sensed by a transducer which inputs its results into pattern space. We may consider the physical world as an infinite-dimensional space of parameters. The transducer describes a representation of the physical world which is in terms of R scalar values where R is typically quite large. R therefore

approaches the dimensionality of pattern space. Since R is quite large and transducers are often defined in terms of cost rather than the specifications of pattern recognition itself, computer time not being insignificant, it is desirable to reduce the dimensionality of R while hopefully minimizing any loss of information. Reducing the dimensionality of R gives us a new N dimensional space known as feature space where $N \ll R$. Classification space is therefore a decision space in which one of k classes is selected for a given sample. It is therefore k dimensional.

Though one may question the necessity of a feature space, it has been contended by many that the greatest advancement that is yet to be made in specific pattern recognition problems will be done when a meaningful pattern to feature space transformation can be determined⁵. This is so because pattern space is always defined by available data sensors which are often defined by convenience rather than for their discriminatory power. Thus, it is not unreasonable to conjecture that there may be linear or highly nonlinear combinations of the convenient parameters of pattern space which might have meaningful classification power. It is also necessary to observe that parameters that may successfully discriminate 'p' from 'q' might not be useful in distinguishing 'p' from 'z'⁵. Hence, feature space should be defined by the inherent discriminatory power of the data that is present in pattern space.

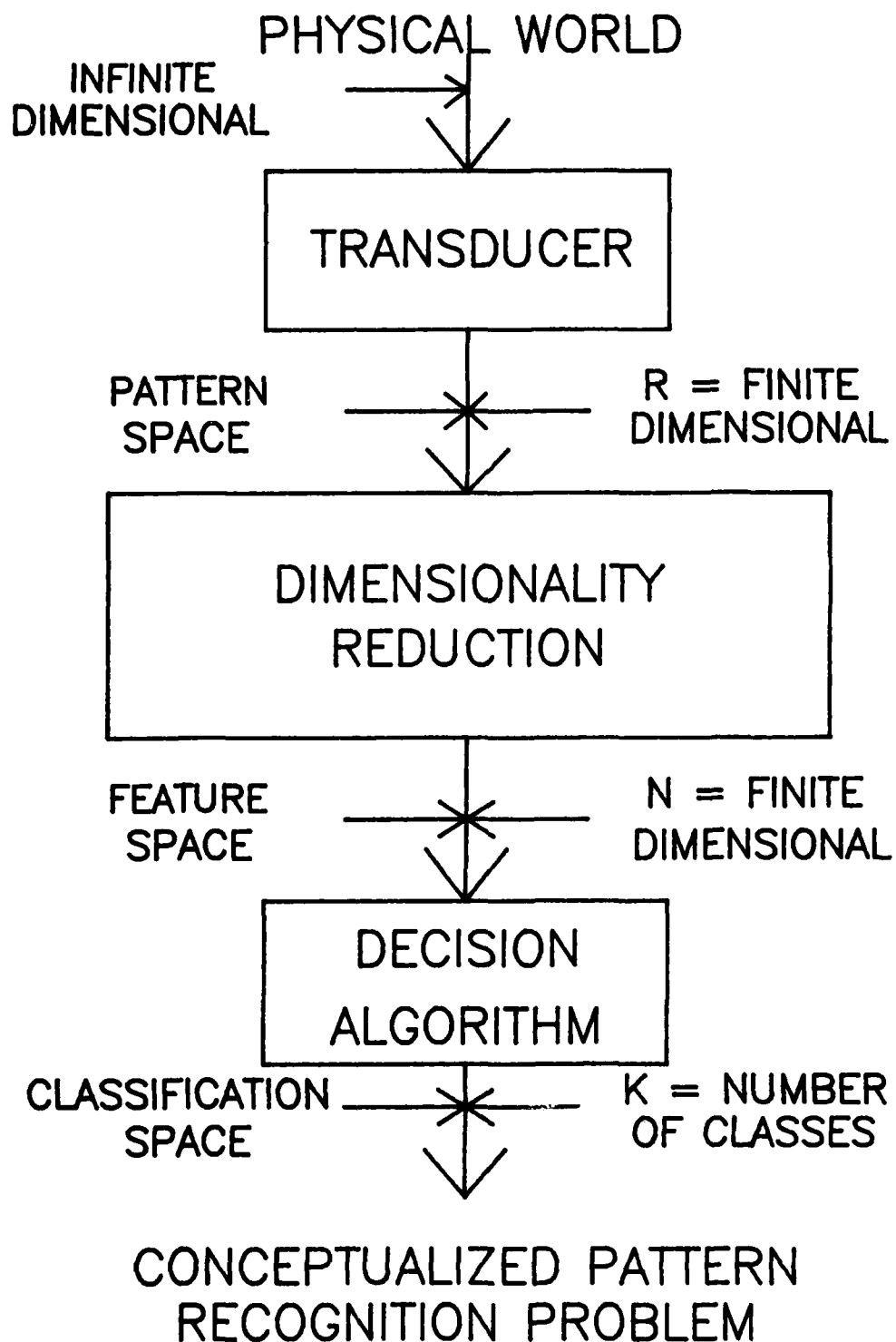


FIGURE 1.0.1

Transducer specifications are certainly of interest, but the core of this paper is classification. It is always good to have a good classifier. However, an ideal feature extractor would lessen the need for an optimum classifier, since classification would become less difficult. In such a case, even a mediocre classifier would do an excellent job. Conversely, with a poor feature extractor, we have more need for an optimum classifier.

The problem of classification involves the partitioning of feature space into regions - one for each category. In general, there is a need to minimize the probability of error by choosing the most appropriate arrangement of partitioning. If some errors are more costly than others, we may wish to reduce the average cost of making an error. In such a case, the problem becomes one of statistical decision theory. Classification space is easiest to describe in the sense that it is k dimensional and it simply contains the decisions implemented by the classification algorithm. Typically, these classification algorithms which define the space partition the N dimensional feature space into disjoint regions - each region associated with only one class. Figure 1.0.2 illustrates the partition of some data in such a manner. The separation surfaces are referred to as hyperplanes in a multidimensional space and are $N-1$ dimensional. Figure 1.0.2 is also ideal.

How well a particular algorithm performs is determined by its ability to minimize the probability of error for a given set of

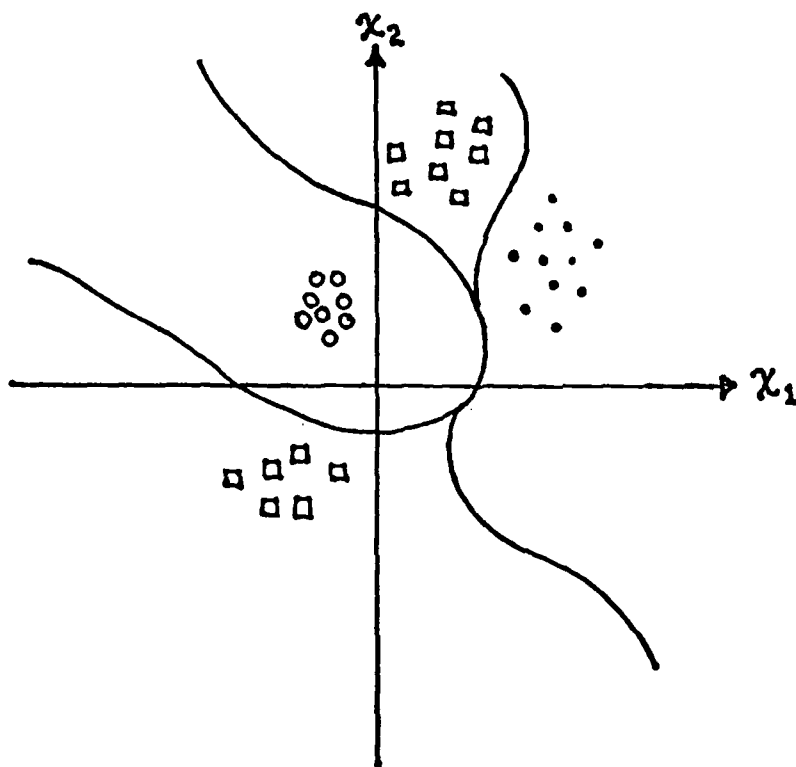


Figure 1.0.2 A Possible Partition of Feature Space.

data. The rest of this paper will be devoted to the analysis and comparison of a parametric and a nonparametric statistical pattern recognition algorithm operating on simulated radar data furnished by the USAF Rome Air Development Center in New York.

Programs of the algorithms used for this analysis were made available by Bruce R. Kowalski from the University of Washington, through Arthur. Arthur is a collection of pattern recognition/general data analysis Fortran programs designed to operate as a flexible, expandable and portable system. "Pattern Recognition," as was embodied in Arthur, is a tool designed to aid in making systematic "educated guess" or analysis of multidimensional data when direct or statistical analysis is not feasible.

1.1 Fundamentals of Statistical Decision Theory

Inherent in radar detection is the problem of parameter estimation. In order to give some significance to the process as it applies to this problem, it is necessary that various signals and spaces associated with radar detection be defined.

Consider the representation of the class of all possible signals as vectors ' $\bar{s}$ ' in a signal space Ω , where each point in the space parameters or feature values. In the case of radar detection, such features may be amplitude, phase, doppler, and so on¹⁵.

In a similar manner, we define noise and clutter space which contain points ' $\bar{n}$ ' that describe all possible waveform realizations of the noise and clutter process within an observation interval¹⁵.

Next, observation space ' T ' which contains points ' $\bar{v}$ '. The ' $\bar{v}$ ' represents all possible joint combinations of signal and noise waveform. The regularity of each point in this set may be represented as an apriori probability distribution function $p(\bar{v}/\bar{s})$. This distribution basically shows the dependence of waveform ' $\bar{v}$ ' on points in signal space¹⁵.

Lastly, let us define ' Δ ', decision space whose points ' $\bar{d}$ ' represent a set of possible decisions in a statistical decision problem. $D(\bar{d}/\bar{v})$ is used to describe the probability density of each decision in decision space for every possible point ' $\bar{v}$ ' in observation space¹⁵.

The diagram shown in Figure 1.1.1 shows parameter estimation
as a decision-making process¹⁵.

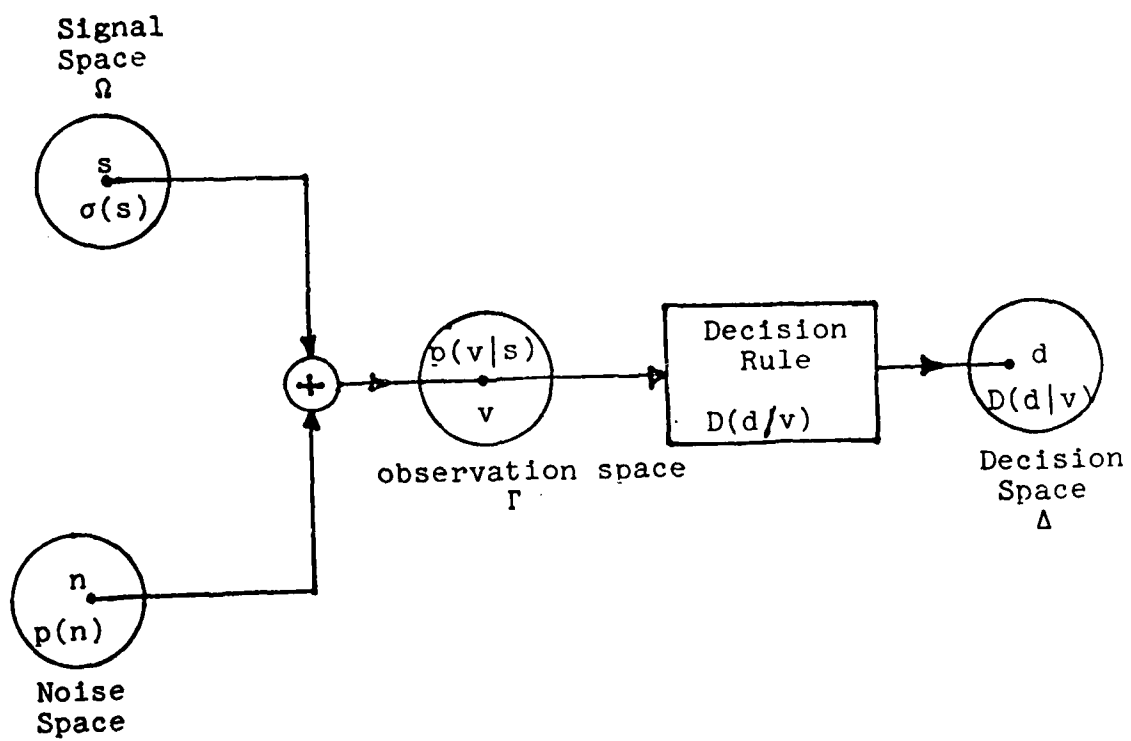


Figure 1.1.1 Reception as a Decision Problem.

2.0 Introduction to Parametric Classification

Parametric classification refers to the development of a statistically defined discriminant function in which the underlying probability density functions are assumed known⁵. The process then simply involves the estimation of a few critical parameters that will define the densities and the corresponding discriminant functions.

Classical techniques in the pattern recognition context provide a basis for studying parametric classification theory which represents the most restrictive of the classification techniques with respect to a priori assumptions on the prototypes and unknown data.

2.1 Discriminant Functions

Let us assume that we have k pattern classes $S_1, S_2, \dots, S_k, \dots, S_K$ with defining prototypes for each pattern $Y_m^{(k)}$, where k is the pattern class and $m = 1, 2, \dots, m_k$, and represents the count of the pattern in class k . Speaking in the context of pattern recognition, what we need ideally is a function which measures each point in pattern or feature space and assigns to that point a certain value which will indicate its membership in any given class. In pattern recognition, such a function is called a discriminant function; in decision theory, it is called a probability density function^{5, 12}. A discriminant function, to be more precise, has the property that it partitions the pattern or feature space into mutually exclusive regions each corresponding to a particular class^{5, 9, 12}. This

function is defined so that for all points x within a given region describing S_k , there exists a function $g_k(x)$ such that $g_k(x) > g_j(x)$ for all $k \neq j$ or:

$$g_k(x) > g_j(x) \quad \forall \quad x \in S_k \quad \text{and} \quad \forall \quad k \neq j \quad (2.1)$$

The hyper-surfaces separating S_k and S_j are given by the expression:

$$g_k(x) - g_j(x) = 0 \quad (2.2)$$

This amounts to the points which have equal discriminant functions for both classes S_k and S_j . There are $k(k-1)/2$ such separating hyper-surfaces in a k class problem^{5, 11}. Often, though, not all surfaces will be significant, and redundant hyper-surfaces will develop⁵. Figure 2.1.0 presents an example of such a situation. Figure 2.1.1 shows a discriminant function classifier and a possible separating surface for a two dimensional space. It should also be pointed out that adding a constant or applying any monotonic nondecreasing function (i. e. logarithm, square, etc.), the discriminant function leaves the decision surface unchanged^{5, 12}. Also, for a two class problem a single discriminant function and a threshold element is sufficient for classification.

$$g(x) = g_1(x) - g_2(x) \quad (2.3)$$

When $g(x)$ is positive, the class chosen is S_1 and when it is negative, S_2 is chosen. $k-1$ discriminant functions are needed to separate k classes.

The adjusting of a discriminant function is referred to as

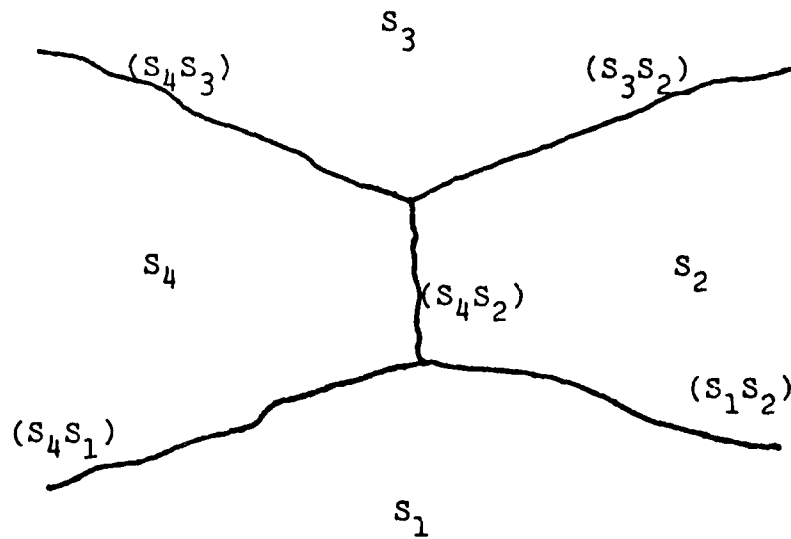
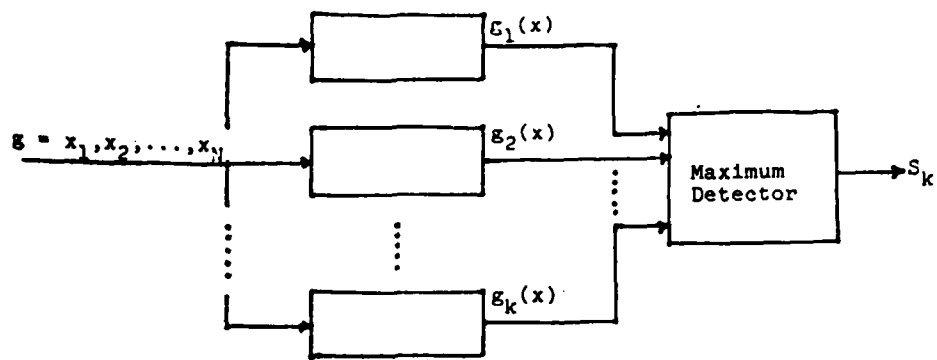
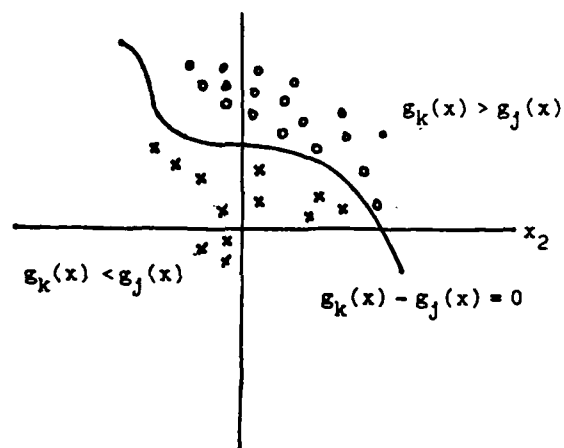


Figure 2.1.0 A Redundant Decision Surface



(a) Classifier



(b) A Decision Surface

Figure 2.1.1 A Typical Classifier

training or learning. If the training is based on known statistics, certain parametric techniques are used. But if it is based on an assumed functional form, for the discriminant function (i. e. linear, quadratic, etc.), distribution free techniques are used.

One of the simplest assumed forms for the discriminant function is known as the linear discriminant function. This function has scalar and vector representation as shown in equations 2.4(a) and (b) below.

$$g_k(x) = W_1^k X_1 + W_2^k x_2 + \dots + W_n^k x_n + W_{n+1}^k \quad (2.4a)$$

or

$$g_k(x) = W_k^t X \quad (2.4b)$$

Where $X = (x_1, x_2, \dots, x_n, 1)$ and $W = (W_1, W_2, \dots, W_n, W_{n+1})$

are the augmented pattern and weight vectors, respectively¹¹.

It should be observed that the scalar term W_{n+1}^k has been added to the discriminant function for a coordinate translation purpose.

This will give the linear discriminant function the capability to pass through the origin of the augmented space when desired. In other words, the surface separating classes S_k and S_j is also linear and may be defined as:

$$g_k(x) - g_j(x) = (W_k^t - W_j^t) X = 0 \quad (2.5)$$

A simple classification algorithm which uses a linear discriminant function is known as a minimum distance classifier⁵.

As an example of such a classifier, let the average point of the patterns defining a given class S_k be given by:

$$Y_k = \frac{1}{M_k} \sum_{m=1}^{M_k} Y_m^{(k)} \quad (2.6)$$

Where M_k represents the number of patterns in class S_k . Then, there exists k such points in pattern space. Let us consider a Euclidean metric for this space and let us assign an unknown point x to that class which has its average value closest to x . The decision rule may then be written as:

$$x \in S_j \text{ if } d(x, Y_j) = \min_k d(x, Y_k) \quad (2.7)$$

however,

$$\begin{aligned} d^2(x, Y_k) &= (x - Y_k)^t (x - Y_k) \\ &= x^t x - 2x^t Y_k + Y_k^t Y_k \end{aligned} \quad (2.8)$$

where x and Y_k are column vectors. According to the properties of a discriminant function, we may subtract the constant $x^t x$ without changing the decision surface⁵. In any case, the algorithm calls for minimum distance. Multiplying by a negative one-half the modified distance squared function becomes a valid discriminant function.

$$g_k(x) = x^t Y_k - \frac{1}{2} Y_k^t Y_k \quad (2.8)$$

In the context of discriminant functions, the elements of Y_k become the linear weights and $-\frac{1}{2} Y_k^t Y_k$ becomes the augmenting property. There exists a set of prototypes Y_m^t assigned to each class S_k . Now, if there exists a linear discriminant function $g_1, \dots, g_k, \dots, g_K$ such that $g_k(Y_m^{(k)}) > g_j(Y_m^{(k)})$ for all $m = 1, \dots, M_k$ and for all $k \neq j$ then, the classes are said to be linearly separable.

The next step in sophistication in defining discriminant functions is given by the piecewise-linear functions. In this case, the separating surface no longer defines a more well behaved region in the pattern and feature space⁵. Therefore, a piecewise-linear machine does not contain the more elegant properties possessed by linear machines⁵. A classic example of a piecewise-linear machine is another form of minimum distance classifier⁵. In this case, the distance of an unknown x for class S_k is given by:

$$d(x, S_k) = \min_{m=1, \dots, m_k} d(x, y_m^{(k)}) \quad (2.9)$$

The distance being considered is actually the smallest distance between all patterns of class S_k and the unknown x . The decision rule becomes:

$$x \in S_j \text{ if } d(x, S_j) = \min_k d(x, S_k) \quad (2.10)$$

The corresponding discriminant function to such an algorithm becomes:

$$g_k(x) = \max \{ x^t Y_m^{(k)} - \frac{1}{2} Y_m^{(k)t} Y_m^{(k)} \} \quad (2.11)$$

A surface of this type is displayed in Figure 2.1.2.

In order to introduce another step up in sophistication for discriminant functions, it is convenient to introduce at this point the concept of a generalized decision function¹¹. It is often seen in the form given by equations 2.12a and 2.12b.

$$d(x) = W_1 f_1(x) + W_2 f_2(x) + \dots + W_k f_k(x) + W_{k+1} \quad (2.12a)$$

or in vector form:

$$d(x) = \sum_{i=1}^{k+1} W_i f_i(x) \quad i = 1, 2, \dots, k \quad (2.12b)$$

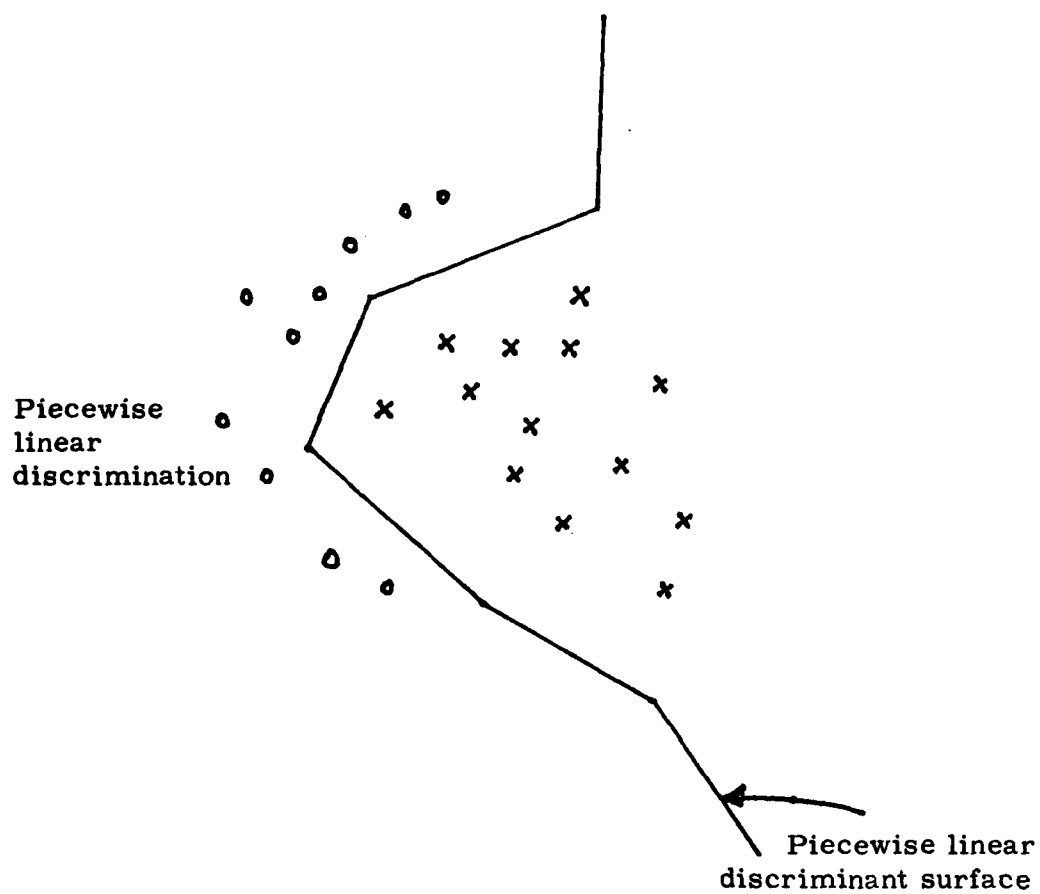


Figure 2.1.2 A Piecewise Linear Discriminant Surface.

Where $\{f_i(x)\}$ are real single valued functions of pattern x , $f_{k+1}(x) = 1$, and $k+1$ are the number of terms used in the expansion. The form of equation 2.12a, b are representative of all discriminant functions^{5, 11}. The various kind of functions may be attained through variation of $\{f_i(x)\}$ and on the number of terms used in the expansion.

Let us define a vector X^* whose elements are $f_i(x)$ so that

$$X^* = \begin{pmatrix} f_1(x) \\ f_2(x) \\ . \\ . \\ f_k(x) \\ 1 \end{pmatrix} \quad (2.13)$$

Now, using equation 2.13 we may express the generalized discriminant function as shown in equation 2.14:

$$g(x) = W X^* \quad (2.14)$$

Where $W = (W_1, W_2, \dots, W_k, W_{k+1})$. Note that x^* is simply a k dimensional vector which has been augmented by one as previously discussed. Hence, equation 2.14 represents a linear function relative to the new patterns X^* . One advantage to this approach is that discussions on discriminant functions may be restricted to the linear type without any loss of generality.

The next step up in sophistication is achieved when $\{f_i(x)\}$ are of polynomial form of second degree, or quadratic. In the

two dimensional case $x = (x_1, x_2)$ and the decision function is of the form:

$$d(x) = W_{11}x_1^2 + W_{12}x_2 + W_{22}x_2^2 + W_1x_1 + W_2x_2 + W_3 \quad (2.15a)$$

This may be expressed in terms of X^* in the linear form as:

$$d(x^*) = W X^* \quad (2.15b)$$

The general quadratic form may be expressed as shown in equation 2.16 if all combinations of components of x which form terms of degree two or less (i. e., if the patterns are n -dimensional). N

$$g_k(x) = \sum_{n=1}^N (W_{nn}^k x_n^2 + W_{nn}^k x_n) + \sum_{n=1}^{N-1} \sum_{j=n+1}^N W_{nj}^k x_n x_j + W_{N+1}^k \quad (2.16a)$$

and in vector form:

$$g_k(x) = x^t A_k x + x^t B_k + W_{W+1}^k \quad (2.16b)$$

2.2 Parametric Classification

Let us consider a set of prototypes in n space with a known distribution. Let us also assume that these points came from a multivariate normal distribution in which case the most we could learn from the data would be contained in its mean vector and sample covariance matrix. The sample mean may be thought of as the point which best represents all the data x in terms of minimization of the sum squared error from all prototypes. The sample covariance matrix gives information on the spread of the data about the mean.

Naturally, if the original assumption about the distribution of the data is incorrect, the statistics are worthless when speaking

in terms of the information they give you about the samples. Obviously, second order statistics would merely be imposing structure on the prototypes rather than revealing its true structure⁹. Figure 2.2.1 displays four different data sets with identical mean and covariance matrix and yet their actual structures are quite different.

Now, with the understanding that a parametric pattern recognition machine will only be as good as the validity of the assumed underlying densities, regardless of mathematical elegance, let us choose a normal distribution for the analysis of this section, simply for the sake of its relative ease of manipulation.

Statistically, we define the sample mean of a set of data points as shown in equation 2.17

$$\mu = E\{x\} \quad (2.17)$$

and in like manner the covariance matrix of equation 2.18.

$$\phi = E\{(x - \mu)(x - \mu)^t\} \quad (2.18)$$

Where the $E\{\}$ represents the expectation operator. The covariance matrix is real, symmetric and positive for real processes⁵. It also has an inverse $[\phi]^{-1}$ and a determinant $|\phi|$. The N-variate normal distribution may then be written as:

$$p(x) = \frac{1}{(2\pi)^{N/2} |\phi|^{1/2}} \exp. \{ -\frac{1}{2}(x - \mu)^t |\phi|^{-1} (x - \mu) \} \quad (2.19)$$

Where $\frac{1}{(2\pi)^{N/2} |\phi|^{1/2}}$ is a normalization constant which makes the area bounded by equation 2.19 of unit value. For convenience, $p(x)$ may be rewritten in the form given by equation 2.19b.

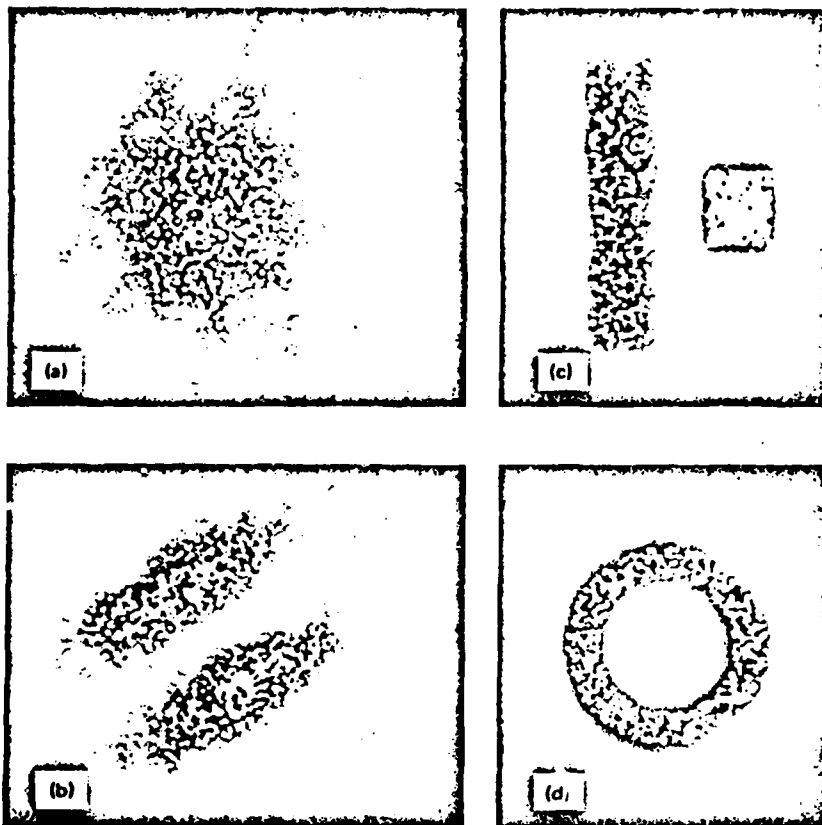


Figure 2.2.1 Data Sets Having Identical Second-Order Statistics.

$$p(x) = N(\mu, [\phi]) \quad (2.19b)$$

When the exponent of equation of 2.19 is constant, the lines of equal probability become hyper-ellipsoidal as displayed in Figure 2.2.2^{5,9}.

In the context of the subject matter being treated, it is of considerable importance that the conditional sensitivity $p(x/S_k)$ be defined. Owing to our knowledge of the correct classification of the known data, we may formulate $p(x/S_k)$ to be of the form given in equation 2.20.

$$p(x/S_k) = \frac{1}{(2\pi)^{N/2} |\phi|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_k)^t [\phi]^{-1} (x - \mu_k) \right\} \quad (2.20)$$

Here the mean and covariance matrix for each class now takes on a significant role. It is intuitively obvious that since we need the first and second order statistics to specify the density, that the mean and covariance matrix take on the values shown in equation 2.21a and 2.21b.

$$\mu_k = E \{ Y_m^{(k)} \} \quad (2.21a)$$

$$[\phi_k] = E \{ (Y_m^{(k)} - \mu_k) (Y_m^{(k)} - \mu_k)^t \} \quad (2.21b)$$

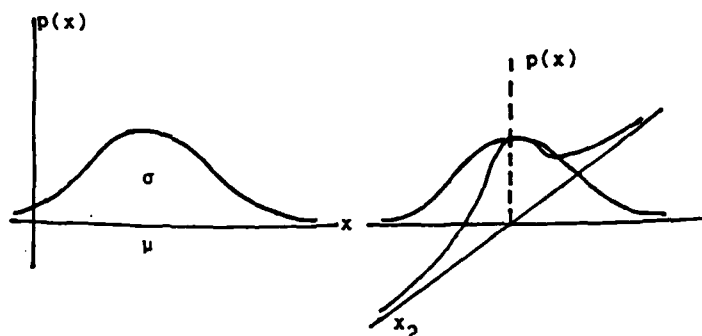
With the use of the previous information of this section, the discriminant function for the symmetric loss function for the Bayes (Classical) technique, which will be treated in the next section, may now be calculated as:

$$g_k(x) = P(S_k) p(x/S_k) \quad (2.22a)$$

or for analytical convenience:

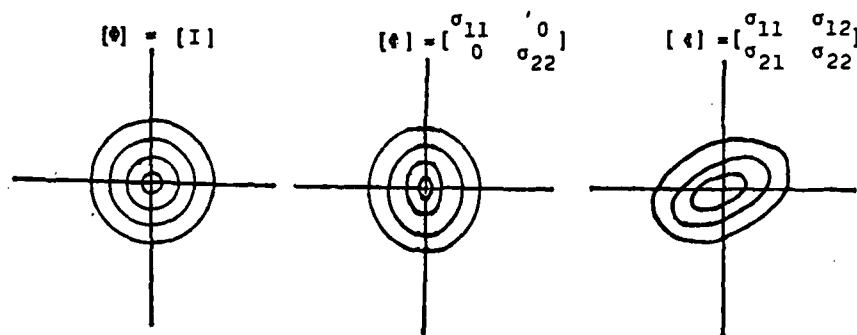
$$g_k(x) = \log \{ P(S_k) p(x/S_k) \} \quad (2.22b)$$

since the log function is monotonic and nondecreasing. Simpli-



(a) One-and two-dimensional normal distribution.

(a) One-and two-dimensional normal distribution.



Note: σ_{nn} is a variance and therefore equivalent to σ_n^2 .

(b) Lines of equal probability ($N = 2$)

Figure 2.2.2 Normal Distribution

Equation 2.22b gives us the following form for $g_k(x)$:

$$g_k(x) = \log P(S_k) - \frac{N}{2} \log 2\pi - \frac{1}{2} \log |\phi_k| - \frac{1}{2} (x - \mu_k)^t [\phi_k]^{-1} (x - \mu_k) \quad (2.23)$$

Eliminating the term which is common to all such discriminant functions, we obtain

$$g_k(x) = -\frac{1}{2} x^t [\phi_k]^{-1} x + x^t [\phi_k]^{-1} \mu_k - \frac{1}{2} \mu_k^t [\phi_k]^{-1} \mu_k + \log P(S_k) - \frac{1}{2} \log |\phi_k| \quad (2.24)$$

In order to proceed with more arguments on this subject, for mathematical simplicity, another simplifying assumption is necessary⁵.

Let's assume that the covariance matrix for each class is the same, since this is a very common occurrence in deterministic communication systems that are perturbed by white Gaussian noise⁵. In this case, the terms $-\frac{1}{2} x^t [\phi_k]^{-1} x$, and $-\frac{1}{2} \log |\phi_k|$ become common to all the discriminant functions and hence may be eliminated from equation 2.24. Its new form is presented in equation 2.25.

$$g_k(x) = x^t [\phi]^{-1} \mu_k - \frac{1}{2} \mu_k^t [\phi]^{-1} \mu_k + \log \{ P(S_k) \} \quad (2.25)$$

where the weight function W and the term which is used for coordinate translation $W_{N+1}^{(k)}$ are given as:

$$W = [\phi]^{-1} \mu_k$$

$$W_{N+1} = -\frac{1}{2} \mu_k^t [\phi]^{-1} \mu_k + \log P(S_k)$$

respectively.

2.3 Classical Technique

The diagram shown in Figure 2.3.1 is of some pertinence to this section since it basically summarizes parameter estimation

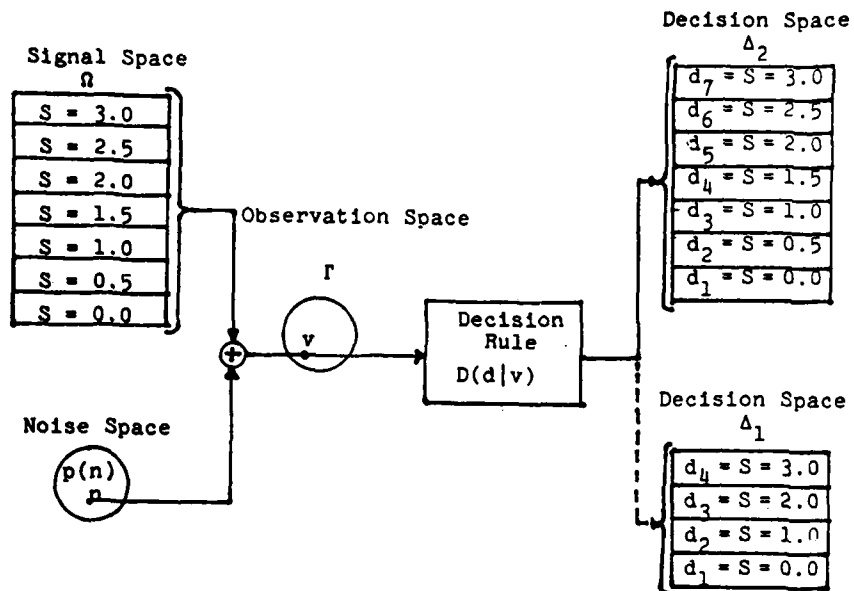


Figure 2.3.1 Parameter Estimation as a Decision Problem.

as a decision making process in terms of the various spaces described previously in Section 1.2.

A decision rule (discriminant function) may be interpreted as an operation which maps points from observation space into decision space. With this in mind, it is quite clear that to optimize the decision process one would like to have an optimum decision. In order that we may have some way to evaluate the performance of these decision rules, let us define a cost function $C(S_i/S_k)$, which will be the loss incurred when a sample pattern x belonging to class S_k is misclassified to class S_i ^{5,9}. This cost or loss function has the advantage of providing the capability to weight specific recognition errors more heavily than others. In order to make use of this function, it is useful to compute a conditional average loss, $L(x, S_k)$ as shown in equation 2.26.

$$L(x, S_k) = \sum_{i=1}^k C(S_i/S_k) p(S_i/x) \quad (2.26)$$

The average loss represents the sum of individual losses weighted by their probability of occurrence. If $L(x, S_k)$ is minimized, then our pattern recognition machine is statistically optimized in the Bayes sense and is often referred to as a Bayes machine⁵. In order to minimize losses, this machine must assign prototype x to category S_k when $L(x, S_k) \leq L(x, S_i)$ for all $i = 1, \dots, k$. This implies that $L(x, S_i)$ must be calculated for each of the k classes. An apparent discriminant function then becomes

$$g_k(x) = -L(x, S_k). \quad (2.27)$$

However, realizing Bayes rule in equation 2.28a,

$$p(S_i/x) = \frac{p(x/S_i)p(S_i)}{p(x)} \quad (2.28a)$$

allows us to rewrite the discriminant function omitting $(p(x))^{-1}$ since it is common to all terms, as shown in equation 2.28b:

$$l(x, S_i) = \sum_{i=1}^k C(S_k/S_i) p(x/S_i) P(S_i). \quad (2.28b)$$

This we will realize as an unconditional average loss while observing the $p(x)$ statistics is missing. Thus far, the conditional average loss has been taken as a value assigned to each class S_k at some point x in pattern space. If this term is integrated over the entire decision space we obtain a risk

$$R(S_k) = \int L(x, S_k) p(x) dx. \quad (2.29)$$

The Bayes rule is then applied to minimize the risk associated with deciding that a particular class is present. The statistics for $p(x)$ are unknown. However, to minimize the risk, we should minimize the maximum worst assumption possible on the distribution of $p(S_k)$ which is uniform $P(S_k) = k^{-1}$ for all $k=1, \dots, k$ classes. This principle is known as the worst criterion on a priori statistics⁵.

For some further illustrations, let us consider the symmetric loss function:

$$C(S_k/S_i) = 1 - \delta(i - k) \quad (2.30)$$

where $\delta(i - k)$ is the Kronecker delta function. Hence,

$$C(S_k/S_i) = \begin{cases} 0 & i = k \\ 1 & i \neq k \end{cases}$$

This basically states that there is zero loss associated with making the correct decision and one unit of loss associated with making the correct decision and one unit of loss associated with making a wrong

decision. This choice of $C(S_k/S_i)$ represents the designer's personal bias since it could be chosen differently. The Bayes decision rule for this loss function is:

$$l(x, S_k) = \sum_{i=1}^k (1 - \delta(i - k))p(x/S_i) P(S_i) \quad (2.31)$$

which may be simplified to

$$l(x, S_k) = p(x) - p(x/S_k) P(S_k) \quad (2.32)$$

Now, to minimize $l(x, S_k)$, we maximize $p(x/S_k)P(S_k)$. The Bayes decision rule becomes: choose S_k if:

$$p(x/S_k)p(S_k) \geq p(x/S_i)p(S_i) \quad (2.33)$$

In terms of a likelihood ratio, we have

$$\lambda = \frac{p(x/S_k)}{p(x/S_i)} \quad (2.34)$$

which simplifies to the choice of category S_k if

$$\lambda \geq \frac{p(S_i)}{p(S_k)} \quad (2.35)$$

which is known as the unconditional maximum likelihood decision.

An obvious discriminant function is given by equation 2.36.

$$g_k(x) = P(S_k) p(x/S_k) \quad (2.36)$$

or for analytical simplicity

$$g_k(x) = \log[P(S_k) p(x/S_k)]$$

The decision surface might also be expressed as seen in equation

2.37a and 2.37b.

$$g_k(x) - g_i(x) = 0 \quad (2.37a)$$

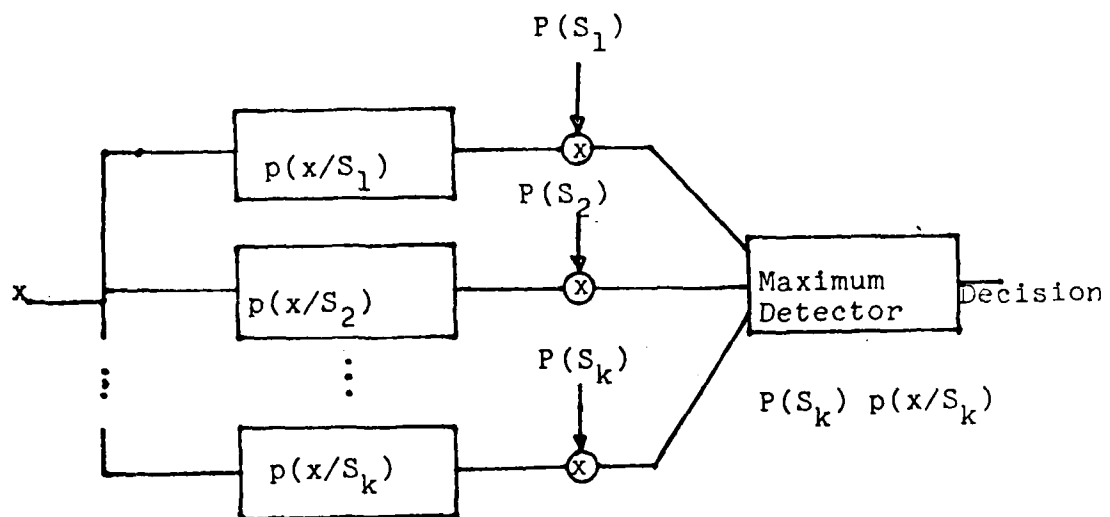


Figure 2.3.2 Bayes Classifier and a Symmetric Loss Function

$$\text{or} \quad \log \left\{ \frac{P(S_k) p(x/S_k)}{P(S_i) p(x/S_i)} \right\} = 0 \quad (2.37b)$$

Figure 2.3.1 shows a block diagram of the representation of a pattern recognition machine for a Bayes classifier.

2.4 Bayes Algorithm Used

This algorithm is an approximation of the Bayes multivariate classification technique. It produces frequency histograms for each feature over each and all categories. At this point, it is important to keep in mind that the accuracy of the results produced by this algorithm will be dependent on how representative are the frequency histograms produced of the true underlying distribution of the various features. The algorithm is considered an approximate Bayes classifier because the true underlying distributions are not known and are only being approximated by the frequency histograms¹⁶.

The loss function used here is:¹⁶

$$C(S_k/S_i) = 1 - \delta(i - k)$$

that was previously discussed. Where

$$C(S_k/S_i) = \begin{matrix} 0 & i = k \\ 1 & i \neq k \end{matrix}$$

The program is quite modular and this could easily be changed but such was not the case.

The decision function for the algorithm is given as the summation over all features of the probability that a given pattern belongs to category k as shown below:

$$g_k(x) = \sum_{j=1}^{NVAR} P_j [x_{j,k}/x_{i,j}]^\alpha$$

where $\alpha = .5, 1$ and 2 and also

$$g(x) = \sum_{i=1}^{NVAR} \ln (P[x_{j,k}/x_{i,j}])$$

For any further information on this algorithm, see Sub-Appendix E-A.

3.0 Nonparametric Classification

Nonparametric techniques in statistical decision models are often resorted to when underlying probability densities are unknown⁵. Nonparametric algorithms are implementable without reference to any specific distribution and are referred to as a "distribution free" technique⁵.

Let us consider a set of patterns and their randomly specified classes $(x_1, \theta_1), (x_2, \theta_2), \dots, (x_n, \theta_n)$; and the problem of classifying some unknown pattern in observation space x_{n+1} , in terms of the known n patterns. Assume that a pattern x takes on some value in observation space and that the θ_1 are random variables which take on values of either zero or one. Let $g(x_{n+1}; (x_1, \theta_1), \dots, (x_n, \theta_n))$ be some arbitrary estimator defined on $X \times (X \times \theta)^n$ which assigns to every X_{n+1} an estimate $g = 0$ or one based on the n training samples. This implies that g partitions the set X into two subsets. Assume once more that $G = \{g_a\}$ is the set of all such decision rules. For example, G may be the set of all k nearest neighbor rules. A major concern is: how does one select the best procedure for the assignment of X_{n+1} ?

As a foundation for further remarks on the topic, either of two basic assumptions on the homogeneity of $(x_1, \theta_1), \dots, (x_n, \theta_n), (x_{n+1}, \theta_{n+1})$ must be made¹.

i) $(x_1, \theta_1), \dots, (x_{n+1}, \theta_{n+1})$ is a collection of $n+1$ independently and identically distributed random variables.

Dependence between x and θ is allowed.

ii) $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n, \tilde{x}_{n+1} \in X$ and $\tilde{\theta}_1, \tilde{\theta}_2, \dots, \tilde{\theta}_n, \tilde{\theta}_{n+1} \in \Theta$ are arbitrary sequences. A permutation π of $1, 2, \dots, n+1$ is chosen at random according to a uniform distribution on the set of $(n+1)!$ permutations. Then an assignment $x_i = \tilde{x}_{\pi(i)}; \theta_i = \tilde{\theta}_{\pi(i)}; i = 1, 2, \dots, n+1$ is made.

Let us assume that $S(\tilde{g})$ is the probability of error associated with making the assignment of x_i 's $i = 1, 2, \dots, n$ to the remaining x_i 's. For a more precise mathematical representation, let σ be a permutation $1, 2, \dots, n$. Also, let $\delta(\tilde{\theta}, \tilde{\theta}) = 1$ or 0 as a direct consequence of $\tilde{\theta} \neq \tilde{\theta}$ or $\tilde{\theta} = \tilde{\theta}$.

We may then define $S(\tilde{g})$ mathematically as shown in equation 3.1.

$$S(\tilde{g}) = 1/n! \sum_{\sigma} \delta[\tilde{\theta}_{\sigma}(i); \tilde{g}(x_{\sigma}(i); (x_{\sigma}(i), \tilde{\theta}_{\sigma}(j))), j = 1, 2, \dots, n, j \neq i] \quad (3.1)$$

For any given $\tilde{g}$, $S(\tilde{g})$ will be a random variable whose distribution is governed by the distribution of (x_i, θ_i) 's.

In general, the classification of x_{n+1} will be formulated in the following way. Firstly, a permutation of σ of $1, 2, \dots, n$ will be chosen according to an equiprobable distribution of the $n!$ permutations. x_{n+1} will then be given the classification as shown in equation 3.2.

$$\hat{\theta}_{n+1} = \hat{g}(x_{n+1}; (x_{\sigma}(1), \theta_{\sigma}(1)), \dots, (x_{\sigma}(n-1), \theta_{\sigma}(n-1))). \quad (3.2)$$

The permutation σ is necessary for bringing symmetry to the data so that the order in which the observation takes place will not be important¹. Now, the risk associated with the classification procedure $R(\hat{g})$ may be expressed as shown in equation 3.3 where

$$R(\hat{g}) = \Pr\{\hat{\theta}_{n+1} \neq \theta_{n+1}\} \quad (3.3)$$

the probability is taken with respect to the distribution of the (x_i, θ_i) 's under either of the previous assumptions as well as the distribution on σ .

One very important point is the fact that $S(\hat{g})$ is an unbiased estimator of the probability of error in using g on x_{n+1} in the sense that

$$R(\hat{g}) = E\{S(\hat{g}) \mid R(\hat{g})\} \quad (3.4)$$

where the expectation is taken over the distribution on $(x, \theta_1), \dots, (x_n, \theta_n)$ and σ . Now, an optimum classifier in $G = g_a$ is the one which minimizes $R(g_a)$. However, since for these non-parametric cases, we do not know $R(g_a)$ we must choose the classifier which minimizes $S(g_a)$ ¹. Notwithstanding the above statement, it is felt that in practical situations, this procedure will develop good decision rules¹.

Let the n samples in the previously defined training set be divided into k disjoint subsets, each containing r samples. Let g be defined on $X \times (X \times \theta)^r$. g will then receive scores $S_1(g), S_2(g), \dots, S_k(g)$ for the errors associated with the various blocks of r patterns. Note that under assumption i, the blocks are independent. $S_i(g)$ $i = 1, 2, \dots, k$ is a set of

independently and identically distributed random variables with common mean $R(g)$. Therefore,

$$S = \sum_{i=1}^k S_i(g)/k \quad (3.5)$$

is an unbiased estimator of $R(g)$ for which the variance approaches zero at a rate $O(1/k)^1$. Let $\{a_n\}$, $\{b_n\}$, $n=1, 2, \dots$, be two sequences of numbers. We say that $\{a_n\}$ is $O(b_n)$, (of the order of b_n), and we write $a_n = O(b_n)$; if $a_n/b_n \xrightarrow{n \rightarrow \infty} 0$.¹⁷

The next section will illustrate a set of decision rules which are pertinent to the context of this discussion and this paper.

3.1 Nearest Neighbor Pattern Classification

For the sake of clarity, let us reassert a few points from the principles of nonparametric pattern classification in order to lay the foundation for the brief principles of the k -Nearest Neighbor rules (k NN).

The domain of nonparametric statistical pattern recognition is rather restrictive in the sense that an optimal decision rule is unattainable on the basis of the underlying statistics of the data under consideration^{1, 9, 14}: This is so because, in cases where the technique is used, knowledge of the underlying distributions are usually unknown except what is inferred from the samples. A decision to classify a point x in observation space into category is allowed to depend only on n correctly classified samples (x_1, θ_1) , $(x_2, \theta_2) \dots, (x_n, \theta_n)$; and a decision procedure which is often by no means a clear cut one¹. The two previous assumptions of section 3.0 still hold, namely, the classified samples (x_i, θ_i) are identically and independently distributed according to the distribu-

tion of (x, θ) ¹.

On this basis, certain heuristic arguments will be made about decision rules for the k-Nearest Neighbor technique. Based on some given measure of similarity, it is fair to say that patterns which are close together will have the same classification, or they should have fairly similar a posteriori probability distributions on their respective classification. Thus, to classify a point in observation space, we could bias our decision on the basis of nearness which provides the basis for one of the simplest and most commonly used decision procedures, the Nearest Neighbor rule (NN) ¹⁴. The first formulation of these Nearest Neighbor rules were made by Fix and Hodges. Surprisingly enough, although simple in concept, it has been shown that in the worst case the k-nearest neighbor rule has a probability of error which is less than twice that of the Bayesian error rate ^{1,9,14}.

Now let us consider a set of n patterns $(x_1, \theta_1), \dots, (x_n, \theta_n)$ where each pattern x_i belongs to category θ_i and takes on in a metric space x upon which is defined a metric d . Consider a new observation (x, θ) where only x is observable and the corresponding category θ is unknown. Based on the information contained in a set of correctly classified patterns, a point $x'_n \in \{x_1, x_2, \dots, x_n\}$ is a nearest neighbor of x if $\min \{ d(x_i, x) = d(x'_n, x) \mid i = 1, 2, \dots, n \}$. This rule will assign x to category θ'_n if its nearest neighbor is x'_n . An apparent error is made when $\theta'_n \neq \theta$.

For the k-Nearest Neighbor, as one might expect, x is classified by assigning it the label most frequently observed among the k nearest samples.

3.2 k-NN Algorithm Used

The choice of k for the k -NN technique to be used in this paper are $k = 1, 3, 7$ and 10 . There is a rule of thumb which suggests that the choice of k should be at most the number of patterns being used divided by five or ten approximately¹⁰. The reason for this is that k is inversely proportional to the probability of misclassification, where the number of samples is much greater than k . Considering the definition of k -NN, one could see that it would be ambiguous to choose k close to or greater than the number of patterns in a given class¹⁴. It will become obvious in Section 5.0 that this procedure is an attempt based on knowledge of a training set to develop a conditional probability distribution, $P(W_i/x)$, where W_i represents class i . We would like to have each data set possessing a fairly high density of patterns because we want all k -NN, x' , of an unclassified pattern, x , to be very close so that $P(W_i/x) \approx P(W_i/x')$. Although large k reduces the probability of error for large sample sizes, we would like to restrict its size so that the chances of x' and x being close to one another are very good^{9,16}.

The criterion for nearness is defined on the basis of interpattern distance. To be specific, the Mahalanobis distance similarity function was used. This function normalizes distance in order to make the analysis invariant to displacement and scale changes⁹. For further details on the k -NN technique used, consult Sub-Appendix E-B.

The use of other similarity measures could cause the algorithms to perform differently. In general, any non-negative real valued function $d(x_i, x_j)$ that satisfies the following requirements may be considered a distance function¹².

- (a) $d(x_i, x_j) \geq 0$ for all x_i and x_j in euclidean space;
- (b) $d(x_i, x_j) = 0$ if and only if $x_i = x_j$;
- (c) $d(x_i, x_j) = d(x_j, x_i)$
- (d) $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$;

where x_i , x_j and x_k are any three vectors in Euclidean space.

The value specified for $d(x_i, x_j)$ represents the distance between data units x_i and x_j .

Table 3. 2. 1 gives a display of some commonly found distance functions¹².

Some Distance Functions

NAME	FORM
1. Euclidean	$d_2(X_i, X_j) = \left[\sum_{k=1}^P (x_{ki} - x_{kj})^2 \right]^{1/2}$
2. l_1 norm	$d_1(X_i, X_j) = \left[\sum_{k=1}^P x_{ki} - x_{kj} \right]$
3. Sup-norm	$d(X_i, X_j) = \sup_{k=1, 2, \dots, P} \{ x_{ki} - x_{kj} \}$
4. l_p norm	$d_p(X_i, X_j) = \left[\sum_{k=1}^P x_{ki} - x_{kj} ^p \right]^{1/p}$
5. Mahalanobis	$D^2(X_i, X_j) = (X_i - X_j)^T W^{-1} (X_i - X_j)$

Table 3.2.1 Some Distance Functions

4.0 Data

Simulated radar ground clutter data furnished by William L. Simkins, Jr., of Rome Air Development Center (RADC) was available for analysis. This data was made available as a result of research sponsored by the USAF/RADC Post-Doctorial program under contract No. CCT-SC-0102-937. This set of data consists of 65,536 samples, each of which may be interpreted as points in a four dimensional space. The four parameters are the x and y coordinates of the region under consideration and a measure of the amplitude and doppler of radar returns from the given x/y coordinates. It will be assumed that varying combinations of each of these parameters is adequate to describe a sample.

The simulated clutter information is on magnetic tape. There is amplitude data in the first file. This amplitude information is displayed in a pseudo-color photo as shown in Figure 4.0.1. There is a color code at the bottom with the lowest amplitude of zero represented by black with each color representing a different category. The size of the field of the various categories may be summarized as shown in Table 4.0.1. The second file contains measurement of the doppler spread of zero mean signal. This feature is displayed in a pseudo-color photo in Figure 4.0.2. There is also a color code at the bottom of this color photo, five of these colors fully describe the Doppler information. Five categories are presented, each of which contain a one-number doppler range as shown in Table 4.0.2.



FIGURE 4. 0. 1 PSEUDO-COLOR PHOTO OF AMPLITUDE DATA



1 (BLACK) .5 0

FIGURE 4. 0. 2 PSEUDO-COLOR PHOTO OF DOPPLER DATA (C_L)

Category	Width of Field/Category (Amp. Range)
1	1
2-12	22
13	13
Total 13	256

Table 4. 0. 1

Category	Doppler Measure (Hz)
1	0
2	49
3	98
4	196
5	147
Total 5	5 bits of doppler

Table 4. 0. 2

With the use of our knowledge of the nature of the data, a standard procedure for obtaining the x/y parameters for each sample was developed. Because of experience with the algorithms that were used to analyse these data, it is impractical to consider working with all 65 K samples. Figure 4.0.3 displays a plot of execution time for both algorithms versus number of samples. Hence, a representative sample was chosen with the help of the pseudo color phototgraphs.

Viewing the pseudo color photographs as a two dimensional x/y graph, the x and y parameters range, in magnitude from zero through 256. The data used for this analysis is bounded by $y = 128-135$ and $x = 1-128$.

The parameters used in this analysis were the x and y coordinates and the amplitude of the radar return. The reason for this was that samples of real radar data whose parameters would be similar to the ones which are being used were expected for comparison and this would provide a basis for comparing how the algorithms work with both real and simulated data.

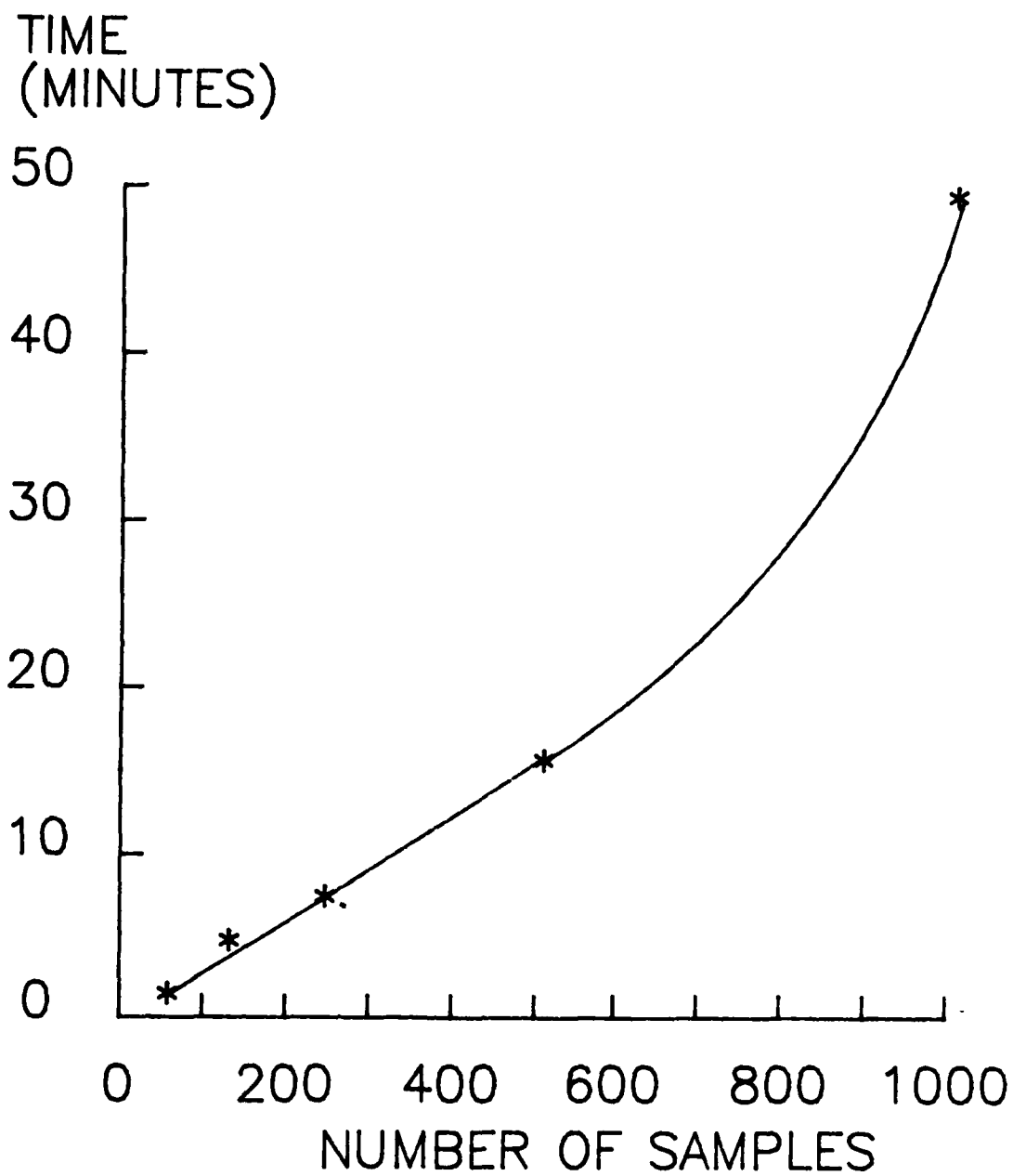


Figure 4.0.3 Execution Time for Both Algorithms Together Versus Number of Samples

5.0 Error Analysis

The purpose of this chapter is to explore the analytical nature of the error bounds of the suboptimal nonparametric pattern recognition classification technique known as the k nearest neighbor. It has been shown that this technique produces an error rate which is greater than the minimum possible P^* and has an optimistic upper bound of approximately $2 P^*$. P^* is also achieved in a practical situation when we have accurate a priori information on the distribution of the data under analysis⁹.

Since the Bayes error rate is in fact the optimum, it will obviously be the lower bound for any other technique including the k -NN. A tight upper bound shall be analytically established for the k -NN technique in order to substantiate expectations of results of analysis carried out on samples of the data previously described.

Let us consider a set of points in observation space x - $\{(x_1, \theta_1), (x_2, \theta_2), \dots, (x_n, \theta_n)\}$ where θ_i is a random variable representation of the category of pattern x_i . θ_i takes on values of W_i $i = 1, \dots, C$, where C = the number of classes. Also, let x'_n be the nearest neighbor of observation x . Recalling the nearest neighbor rule, we see that it would assign observation x to category θ'_n . Now the chances that $\theta'_n = W_i$ may be represented by a conditional probability function $P(W_i/x'_n)$. Assuming that we have a very large sample ($n \rightarrow \infty$), it can be shown that x is close enough to x'_n to that $P(W_i/x'_n)$

$P(W_i/x)$ ⁹. Let us define $W_m(x)$ as the category which optimizes the distribution such that:

$$P(W_m/x) = \max_i P(W_i/x) \quad (5.1)$$

By definition an optimum decision rule is one which selects W_m in all cases. The minimum error associated with classifying an observation x may then be expressed as:

$$P^*(e/x) = 1 - P(W_m/x) \quad (5.2)$$

and hence the minimum unconditional average probability of error over observation space may then be expressed as:

$$P^* = \int P^*(e/x) p(x) dx \quad (5.3)$$

There will be fluctuations in the error rate for different sets of n samples. This will certainly be the case since for each sample used in the classification of observation x , there will be fluctuations in the nearest neighbor vector x'_n . This implies a joint dependence of the n sample error rate, $P_n(e/x, x'_n)$, on both x and x'_n . Averaging over x' yields:

$$P_n(e/x) = \int P_n(e/x, x'_n) P(x'_n/x) dx'_n \quad (5.4)$$

With the previous assumptions on the sample size and the fact that x'_n is the nearest neighbor of x , it is intuitively appealing to choose $p(x'_n/x)$ to be a delta function centered about x , which is, in fact, not a bad assumption^{5,9}. Suppose we call the probability that any sample falls within a hypersphere S centered about x , is some positive number P_s . Then the chance that all of the n independently drawn samples fall outside the hypersphere may be represented as $(1-P_s)^n$, which

approaches zero as n approaches infinity.

Recalling assumption i of section 3.0 and considering the fact that it still holds true, the complementary conditional probability of error may be written as shown in equation 5.5.

$$P(\theta, \theta'_n / x, x'_n) = P(\theta / x) P(\theta'_n / x'_n) \quad (5.5)$$

Since x and x'_n are nearest neighbors $\theta = \theta'_n = W_i$

$$P_n(e/x, x'_n) = 1 - \sum_{i=1}^c P(W_i/x) P(W_i/x'_n) \quad (5.6)$$

In order to obtain an expression for $P_n(e)$ equation 5.6 is substituted into equation 5.4 and this expression is averaged over x . Recall that n approaches infinity and $p(x'/x)$ approaches a delta function. If $P(W_i/x)$ is continuous at x , the equation simplifies to:

$$\begin{aligned} \lim_{n \rightarrow \infty} P_n(e/x) &= \sum_{i=1}^c [1 - \sum_{i=1}^c P(W_i/x) P(W_i/x'_n)] \\ &\quad \delta(x'_n - x) dx'_n \\ &= 1 - \sum_{i=1}^c P^2(W_i/x) \end{aligned} \quad (5.7)$$

The asymptotic nearest neighbor error rate may be developed as shown in equation 5.8.

$$\begin{aligned} P &= \lim_{n \rightarrow \infty} P_n(e) \\ &= \int [1 - \sum_{i=1}^c P^2(W_i/x)] p(x) dx \end{aligned} \quad (5.8)$$

P^* is a lower bound for the error rate of the nearest neighbor technique. Furthermore, it is also fair to say it is a tight lower bound since there is always a set of conditionals and priori probabilities given which P^* is attainable. There-

fore, the problem now lies in the determination of a tight upper bound for P .

In order to find an upper limit on P , we must determine how small $\sum_{i=1}^c P(W_i/x)$ of equation 5.8 can be for a given $P(W_m/x)$. This summation may be minimized subject to the following:

$$(1) P(W_i/x) \geq 0$$

$$(2) \sum_{i \neq m} P(W_i/x) = 1 - P(W_m/x) = P^*(e/x)$$

$\sum_{i=1}^c P^2(W_i/x)$ will be minimized if we choose each $P_{i \neq m}(W_i/x)$ equal to one another. This implies:

$$P(W_i/x) = \begin{cases} \frac{P^*(e/x)}{c-1} & i \neq m \\ 1 - P^*(e/x) & i = m \end{cases} \quad (5.9)$$

Hence

$$\sum_{i=1}^c P^2(W_i/x) \geq (1 - P^*(e/x))^2 + \frac{P^{*2}(e/x)}{c-1}$$

and

$$1 - \sum_{i=1}^c P^2(W_i/x) \leq 2P^*(e/x) - \frac{c}{c-1} P^{*2}(e/x) \quad (5.10)$$

which shows that $P \leq 2P^*$.

The previous developments show that the nearest neighbor error rate is roughly bounded by the minimum possible error rate P^* (Bayes error rate) and $2P^*$ expressed mathematically as

$$P^* \leq P \leq P^* (2 - \frac{c}{c-1} P^*) \quad (5.11a)$$

In order to provide some insight into the error bounds of the other nearest neighbor rules under consideration (3, 7 and 10) the error bounds of the k-Nearest Neighbor rule will now

be considered for cases in which k is greater than one. This rule classifies an observation x by assigning it the label most frequently represented among the k nearest samples^{1,6,9}.

Some basic principles from the nearest neighbor rule still hold for the k -Nearest Neighbor classification scheme. Assuming that k is fixed and that the number of samples are allowed to approach infinity, then all of the k nearest neighbor will converge to x as discussed in Section 5.0 (p. 46)⁹. The labels of each k nearest neighbors are random variables which independently assume the value W_i with probability $p(W_i/x)$ $i = 1, 2$ implying a two class problem. The k nearest neighbor rule will select W_m . The probability of such an occurrence may be expressed as:

$$\sum_{i=(k+1)/2}^K \binom{k}{i} (P(W_m/x))^i [1 - P(W_m/x)]^{k-i}$$

In general, as k increases so does the chance that W_m is selected. With the same arguments that were used in the first nearest neighbor case, it can be shown that if k is odd, the upper bound of the error rate in a two class problem for the k nearest neighbor error rate is given by $C_k(P^*)$, where C_k is defined to be the smallest concave function of P^* greater than

$$\sum_{i=0}^{(k-1)/2} \binom{k}{i} [(P^*)^{i+1} (1-P^*)^{k-1} + (P^*)^{k-1} (1-P^*)^{i+1}] \quad (5.11b)$$

With the evaluation of $C_k(P^*)$, the bounds of the k nearest neighbor error rate are observed to be as shown in Figure 5.0.1. Note that as k approaches infinity, the upper bound on

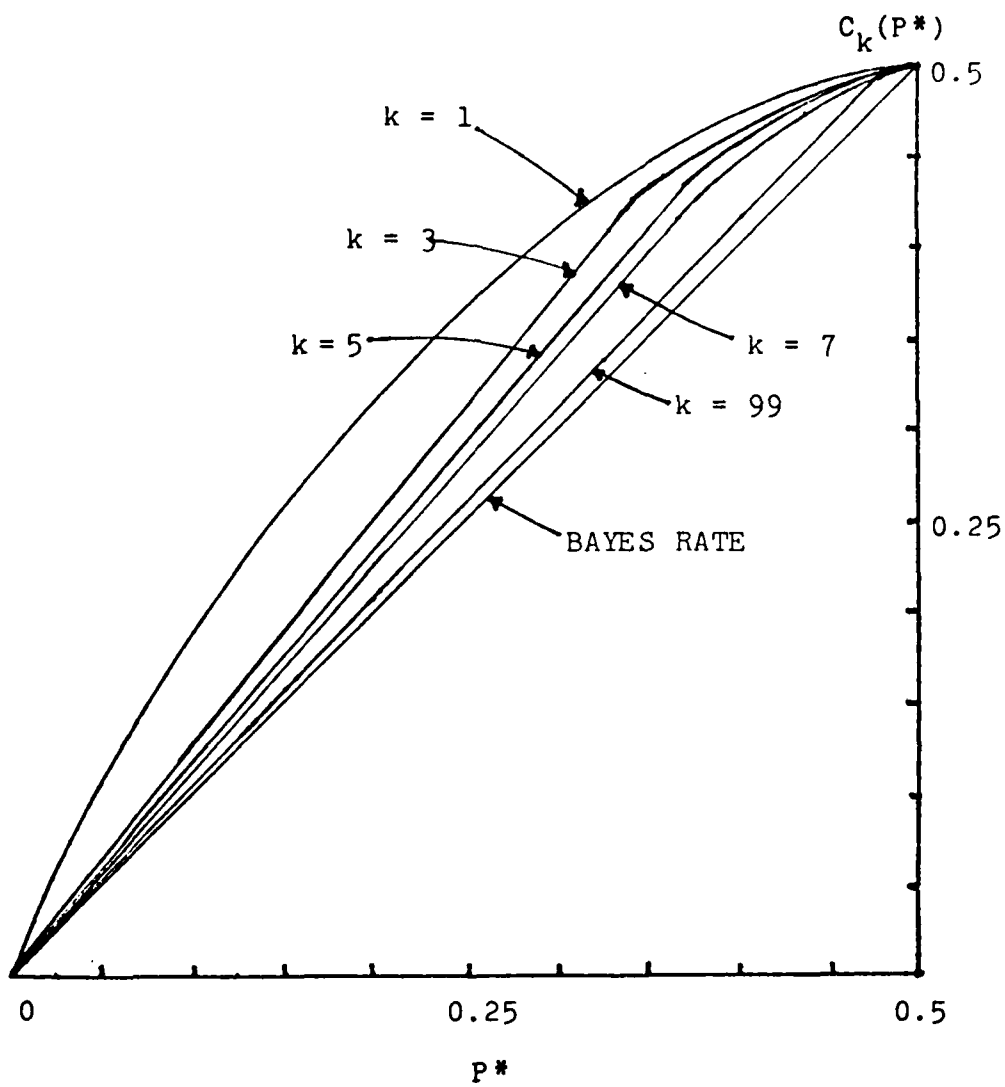


Figure 5.0.1
Bounds on the Error-Rate for the k -Nearest Neighbor Rule

the k nearest neighbor error rate converges to P^* .

5.1 Methods for Evaluating the Probability of Misclassification

At this point the most important development yet to be made involves the development of a reliable technique for estimating the performance of the Bayes and k -NN algorithms. Ideally, what we would desire to have at this point is the actual probability of error P_e obviously cannot be obtained because we do not have accurate information on the underlying distribution, which is a result of the fact that we only have a finite amount of samples to work with^{7,13}. Let P_e be the best estimate of the probability of error P_e which may be obtained when one has an infinite sample size and uses one-half to train and the other half to test the given classifier* P_e also cannot be obtained because by definition, all the sample patterns will be used to train the classifiers and none will be left for testing them. In the next section, some of the more important methods which have been developed and experimentally compared will be discussed. The particular method used in the analysis for this paper will be discussed and substantiated.

5.2 Error Estimation Techniques

Throughout the entirety of this discussion, let $\{x\} = \{x_1, x_2, \dots, x_N\}$ be the set of pattern samples at our disposal. In other words, $\{x\}$ contains N patterns.

The first error measuring technique to be considered here is called the R -method. Its resultant error rate is denoted

as $\hat{P}_e [R]$. R in this context stands for redistribution and its procedure is given in the following steps:

- (i) The classifier is trained on $\{x\}$.
- (ii) The classifier is tested on $\{x\}$.

This technique was developed during the early stages of pattern recognition, but was more or less put aside in the light of inadequacies and developing interest in generalized capability learning machines. This new interest gave rise to the second method to be considered in this section.

This method is called the H or the Holdout-method and its resultant probability of error is denoted as $\hat{P}_e [H]$. Typically in this procedure one-half of the available samples are used for training and the other half for testing the classifiers.

This method may be accomplished through the following steps:

- (i) Partition $\{x\}$ into two mutually exclusive sets

$$\begin{aligned} \{x\}_\alpha \text{ and } \{x\}_\beta \text{ such that} \\ \{x\}_\alpha &= \{x_1, x_2, \dots, x_{N(\alpha)}\} \\ \{x\}_\beta &= x_{N(\alpha)+1}, x_{N(\alpha)+2}, \dots, x_N \end{aligned}$$

Where $N(\alpha) = N/2$.

- (ii) Train the classifier on $\{x\}$.
- (iii) Test the classifier on $\{x\}$.

Although it is commonly the case, $N(\alpha)$ does not have to be $N/2$. In fact, in 1962, W. H. Highleyman presented a paper in which he showed how the set $\{x\}$ may be partitioned for various values of N. However, it has been shown by other researchers that his analysis is only valid for very large N when, in fact,

the problem of estimating $\hat{P}_e$ is mostly concerned with errors associated with small values N . In any case, with rather frequent usage of this technique, frequent observation of discrepancies between $\hat{P}_e[R]$ and $\hat{P}_e[H]$ were reported. In general, observation showed that $\Delta \hat{P}_e(H - R) = \hat{P}_e[H] - \hat{P}_e[R]$ is always positive and inversely proportional to the data size N . As it turns out, $\hat{P}_e[R]$ is an over optimistic estimate of P_e and $\hat{P}_e[H]$ is a pessimistic estimate of P_e where

$$\hat{P}_e[R] \leq P_e \leq \hat{P}_e[H] \quad (5.11)$$

The H method was further developed to increase its data handling efficiency⁷. The data set in this case is divided into mutually exclusive pairs, and $\hat{P}_e[H]$ for each is calculated. The expectation operation is then applied to the set which results in $E \{ \hat{P}_e[H] \}$.

This brings us to yet another method for estimating $\hat{P}_e$. This procedure is called the U-method and its error rate is denoted as $\hat{P}_e[U]$. The method may be accomplished through the following steps:

- (i) Take one pattern sample x_i from $\{x\}$. Then
 $\{x\}' = x_1, x_2, \dots, x_{N-1}$
- (ii) Train the classifier on $\{x\}'$.
- (iii) Test the classifier on x_i . If x_i is correctly classified, set $n_i = 0$, otherwise set $n_i = 1$, where n_i acts as an error indicator.
- (iv) Do steps i, ii, iii for $i = 1, \dots, N$ to obtain values for n_i $i = 1, \dots, N$.
- (v) Estimate $\hat{P}_e[U]$ as:

$$\hat{P}_e[U] = \frac{1}{N} \sum_{i=1}^N n_i \quad (5.13)$$

This procedure is also known as the "leaving one out" method and it may be considered the most efficient error estimation technique since it maximizes the information achievable from the data. In spite of its efficient use of the data, the U-method has one obvious disadvantage. This lies in the fact that for the evaluation of $\hat{P}_e[U]$ we need as many runs as we have samples and this might be quite costly in terms of time and money when we are considering a large sample. As a result of this disadvantage, a procedure was proposed by G. T. Tousseint which reduces the amount of runs necessary and produces an error rate which is an unbiased estimator of $\hat{P}_e[U]$.

This compromise procedure is known as the rotation or π -method and the steps necessary to implement this procedure are as follows:

- (i) Take a small subset of pattern samples $\{x\}_i^{TS} = \{x_1, x_2, \dots, x_p\}$ such that $1 \leq p \leq N$ and N/p is an integer, $p/N < \frac{1}{2}$. Then $\{x\}_i^{TR} = x_1, x_2, \dots, x_{N-p}$
- (ii) Train the classifier on $\{x\}_i^{TR}$.
- (iii) Test the classifier on $\{x\}_i^{TS}$ to obtain an estimate of the error probability denoted by $\hat{P}_e[\pi]_i$.
- (iv) Do steps i, ii, iii, for $i = 1, 2, \dots, N/p$ such that $\{x\}_i^{TS}$ and $\{x\}_j^{TS}$ are disjoint for $i = 1, \dots, N/p$.
- (v) The resulting estimate of $\hat{P}_e$ is computed as:

$$E \{ \hat{P}_e[\pi] \} = \frac{P}{N} \sum_{i=1}^{N/P} \hat{P}_e[\pi]_i \quad (5.14)$$

One interesting observation is the fact that when $P = 1$ the π -method becomes the U-method or when $P = N/2$, the π -method becomes the H-method. This more or less shows that the π -method is a compromise between the U and H method.

The result of the exposition in error estimation may be summarized by the following set of inequalities:

$$E \{ \hat{P}_e[R] \} \leq \hat{P}_e \leq E \{ \hat{P}_e[U] \} \leq E \{ \hat{P}_e[\pi] \} \leq E \{ \hat{P}_e[H] \} \quad (5.15)$$

5.3 Performance Measure Used

As a result of experiences from preliminary runs, the observation of fluctuation in the optimistic redistribution or R-method error rate with sample size and also from previous analysis that showed that there will be fluctuation of the error rate with sample size, it was decided that to estimate the performance of the classifiers under consideration, five different sample sizes would be chosen. The specific sample sizes decided upon 60, 135, 255, 510 and 1005 samples respectively. This choice of data sizes was more or less random, but was chosen with the thought that there would be a greater fluctuation of the error rate for smaller sample sizes. So, the pregression of the data size was chosen to be approximately $2N$. The classifiers to be tested with these samples are the k-Nearest Neighbor and Bayes technique with their various variations. The result of some of these preliminary runs which influenced the decision about the choice of data size are presented in Table 5.3.1.

Bayes Algorithm

1024 Samples

% Correct/Discriminant Function			
Ln(p)	$p^{(.5)}$	$p^{(1)}$	$p^{(2)}$
28.91	25.98	50.68	74.32

(a)

k - NN Algorithm

1024 Samples

% Correct/Discriminant Function								
1-NN	3-NN	4-NN	5-NN	6-NN	7-NN	8-NN	9-NN	10-NN
100	99.71	99.51	99.22	99.02	98.93	98.83	98.83	98.63

(b)

Bayes Algorithm

257 Samples

% Correct/Discriminant Function			
Ln(x)	$x^{(.5)}$	$x^{(1)}$	$x^{(2)}$
41.25	82.88	74.32	71.98

(c)

K-NN Algorithm

257 Samples

The most accurate available method for evaluating the error rates is the U-method^{7,13}. Its application to this experiment, however, would be quite impractical considering the costs that would be involved. Over 1,985 runs would be required per test. Considering the time and cost per run, this technique had to be eliminated. The best available alternative would be the π -method and it was in fact the one that was chosen. As discussed previously, its data handling capability is less efficient than the U-method, however, its error rate $\hat{P}_e[\pi]$ is an unbiased estimator of $\hat{P}_e[U]$ and the number of runs required could be far less than that of the U-method. Recall equation 5.15.

$$E\{\hat{P}_e[R]\} \leq \hat{P}_e \leq E\{\hat{P}_e[U]\} \leq E\{\hat{P}_e[\pi]\} \leq E\{\hat{P}_e[H]\}$$

Where N = the number of samples and P = the number of test samples per run, Table 5.3.2 gives the layout of each group of data used in the experiment. In each case, P was chosen such that the ratio P/N remains the same. This was done so that the efficiency of the handling of the data would stay the same.

Now, for each data set in Table 5.3.2, the error rate was determined according to equation 5.14 and presented in Table 5.3.5.

$$E\{\hat{P}_e[\pi]\} = \frac{1}{15} \sum_{i=1}^{15} \hat{P}_e[\pi]_i,$$

for each variation of the two classifiers. Figure 5.2.1 presents a graphical representation of these error rates. For further details on the results, see Sub-Appendix E-C.

N	P	P/N
1005	67	1/15
510	34	1/15
255	17	1/15
135	9	1/15
60	4	1/15

Table 5.3.2
Layout for the Various Groups of Data Used in Experimentation.

$$E[\hat{P}_e(\pi)]$$

Run #	Ln	P_e for Bayes			P_e for KNN			
		**5	**1	**2	1 st NN	3 rd NN	7 th NN	10 th NN
60	1	.8500	.8333	.8833	.5166	.6333	.6667	.7667
135	2	.7111	.4518	.4518	.0444	.0889	.0963	.1111
255	3	.6823	.3059	.2980	.0706	.0745	.0549	.0549
510	4	.4059	.3392	.4384	.0429	.0427	.0412	.0510
1005	5	.2886	.2336	.3537	.0358	.0328	.0384	.0403
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

Table 5.3.3
Error Rates Determined From Tests.
(x, y, Amplitude)

Contrary to the previous analysis, the k -NN classifier gave far better results than the Bayes technique. There will be further explanation of these results in the final chapter.

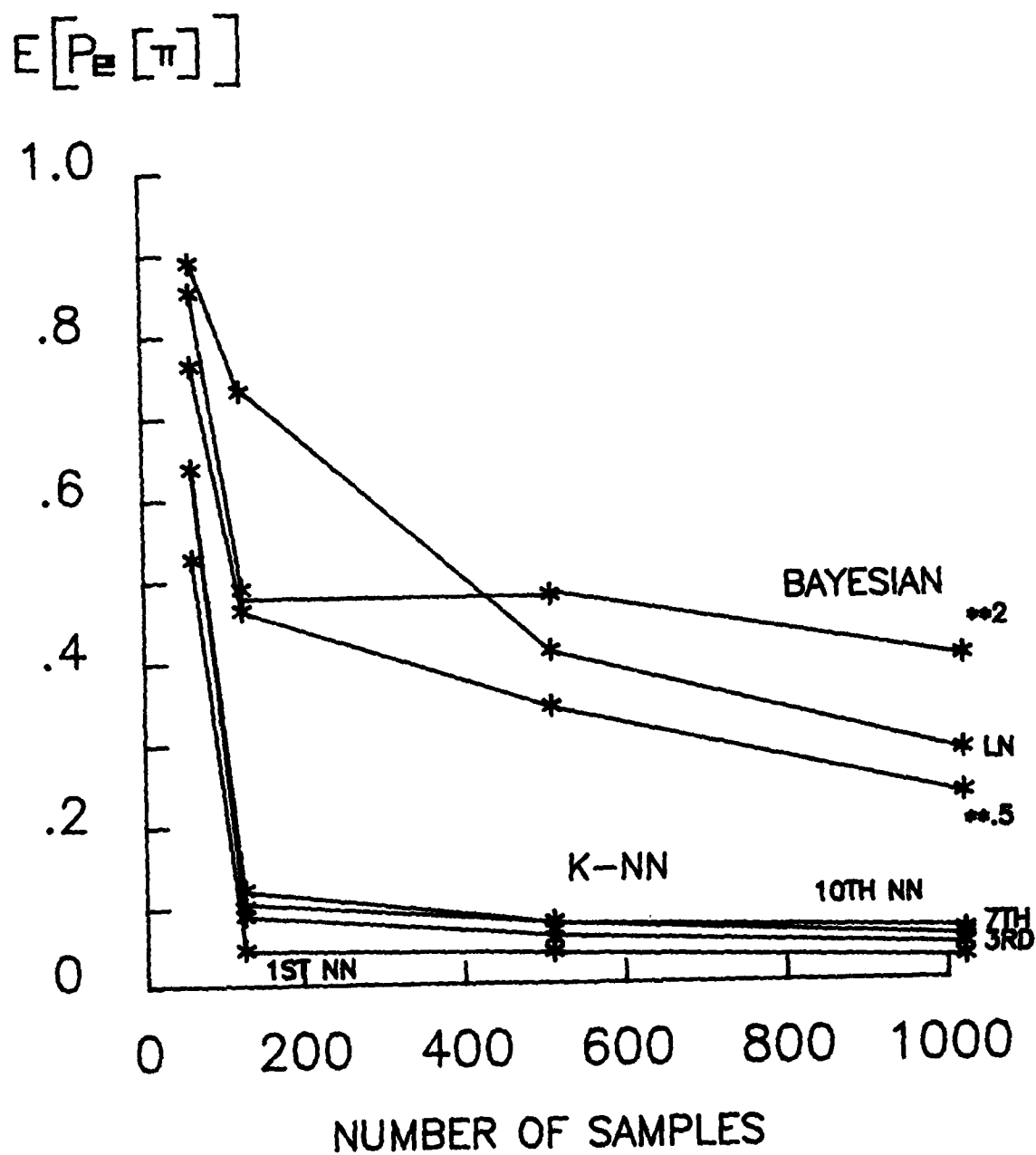


FIGURE 5.2.1 ERROR CURVES

6.0 Summary

Parametric classification refers to the development of a statistically defined discriminant function in which the underlying probability density functions are assumed known⁵. The Bayes rule minimizes the error associated with deciding that a given class is present given an unknown sample x . The accuracy of the results given by the Bayes algorithm is dependent on how representative the frequency histograms are of the true underlying distribution of the data under consideration⁹.

For an infinite data size, the larger k is the more accurate will be the results produced by the k -NN classifier^{9, 14}. For a finite data size choosing k too large could give poor results. A rule of thumb states that k should be at most the number of patterns in the smallest category divided by five or ten¹⁰. Nonparametric statistical pattern recognition is rather restrictive in the sense that an optimal decision rule is unattainable since the underlying statistics of the data under consideration is unknown^{1, 9}. Where P represents the k -NN error rate, its performance is bounded from below by the Bayes error probability and from above by at least twice the Bayes error probability⁹. Note, also, that $P \leq 2P^*$ is only an upper bound for the k -NN rule when $k = 1$. In fact, for an infinite data size as k approaches infinity, both the upper and lower bound of the k nearest neighbor error rate converge to the Bayesian error rate P^* .

The performance of the various error estimation techniques

considered may be summarized by the following equation;

$$E\{\hat{P}_e[R]\} \leq \hat{P}_e \leq E\{\hat{P}_e[U]\} \leq E\{\hat{P}_e[\pi]\} \leq E\{\hat{P}_e[H]\}. \quad (6.1)$$

The data that was used for the analysis of these algorithms is simulated radar ground clutter information. This should pose a very interesting analysis for the algorithms because of the nature of the data. The nature being that the distribution of the radar ground clutter is inescapably dependent upon the background that is being scanned. This diversity in the distribution that may be encountered due to its dependence on the background should present a rather interesting test for the parametric Bayesian technique whose performance is so dependent on data distribution. Also, this data set will present results which will point out some advantages of the heuristic distribution free k-Nearest Neighbor technique.

The results showed that the k nearest neighbor algorithm performed better than the Bayesian algorithm on all accounts for this particular combination of the data. Recall that P^* is used as a standard by which to judge the performance of other algorithms since it represents the ultimate error rate (the Bayes error rate). It seems ironic that since the upper bound on the k-NN error rate is only $2P^*$ that we could now use the nearest neighbor error rate as a standard by which to measure how accurate the assumptions about the underlying distributions actually were in the Bayes algorithm.

Although execution rate and memory allocation were not the prime considerations in the analysis, we consider our findings in the areas to be worthy of some recognition. Table 6.0.1 and Figure 6.0.1 summarize the memory requirements and the execution time of both algorithms. Also, Figure 6.0.2(a) and (b) present a map of the amplitude categories and the location of the

occurrence of errors indicated by a "*" for the best Nearest Neighbor and Bayesian technique with 255 samples.

Algorithms	IBANK	DBANK	COMMON BANK	TOTAL
KNN	2339	2207	72	4618
BAYES	2252	2165	114	4531

Table 6.0.1
Memory Allocation to the Two Algorithms.

6.1 Conclusion

The performance of the k-NN classifiers was directly analogous to the analytical arguments presented in Section 3.2. The first and third nearest neighbors perform the best because the density of these patterns in some categories were quite small in which case choosing k large is somewhat ambiguous.

The various Bayes classifiers used differ only by the fact that different monotonic nondecreasing functions were applied to the discriminant function. There was no reason to suspect why the performance of any particular one of these classifiers should be better than the other. However, the heuristic conclusion drawn about these decision functions is that, based on the results presented from the version of the Bayes algorithm used in this paper and the data under consideration, the decision function raised to the one-half power gives the lowest error rate for this algorithm. In terms of performance it is followed by the logarithmic, the linear and the squared decision function respectively.

As far as a comparison goes between the Bayes and the k-NN classifiers used, the results indicated that the four nearest neighbor classifiers, $k = 1, 3, 7$ and 10 , used give a smaller probability of error than all the variations of the Bayes classifier considered. This is contrary to all previous analysis. This may be explained by the fact that the parametric Bayes classifier considered. This is contrary to all previous analysis. This may be explained by the fact that the parametric Bayes

algorithm is only as good as the underlying assumption about the distribution of the data. The poor performance of the Bayes classifier is an indicator of the fact that the histograms formed in an attempt to approximate the distribution of the data over each class was hardly representative of the true distribution for the various classes.

Estimating the underlying statistics by means of histograms will not necessarily be of much use unless they contain information on the population of all possible samples. On the other hand, if we assume some distribution, there is no guarantee that it will truly symbolize the correct densities and therefore it may give arbitrarily poor results. This presents one more reason to show that the k-NN technique may be a more practical one than the Bayes classifier.

6.2 Recommendations for Future Work

There is a great deal of uncertainty and variability about what is known of the probability distribution of radar ground clutter. However, it is quite obvious that the distribution of radar ground clutter will be highly dependent upon the characteristics of the background.

Because of the very nature of ground clutter, specifically the variability of its distribution due to background characteristics, and equally important the inferior results obtained from the Bayes algorithm in this analysis, the use of the non-parametric k-NN technique over the Bayes classifier is recom-

mended for the characterization of ground clutter. Also, because of the fairly conservative upper bound on the k-NN classifier, I suggest that its results be used as a standard for evaluating assumptions made on the underlying distribution by the parametric Bayes technique.

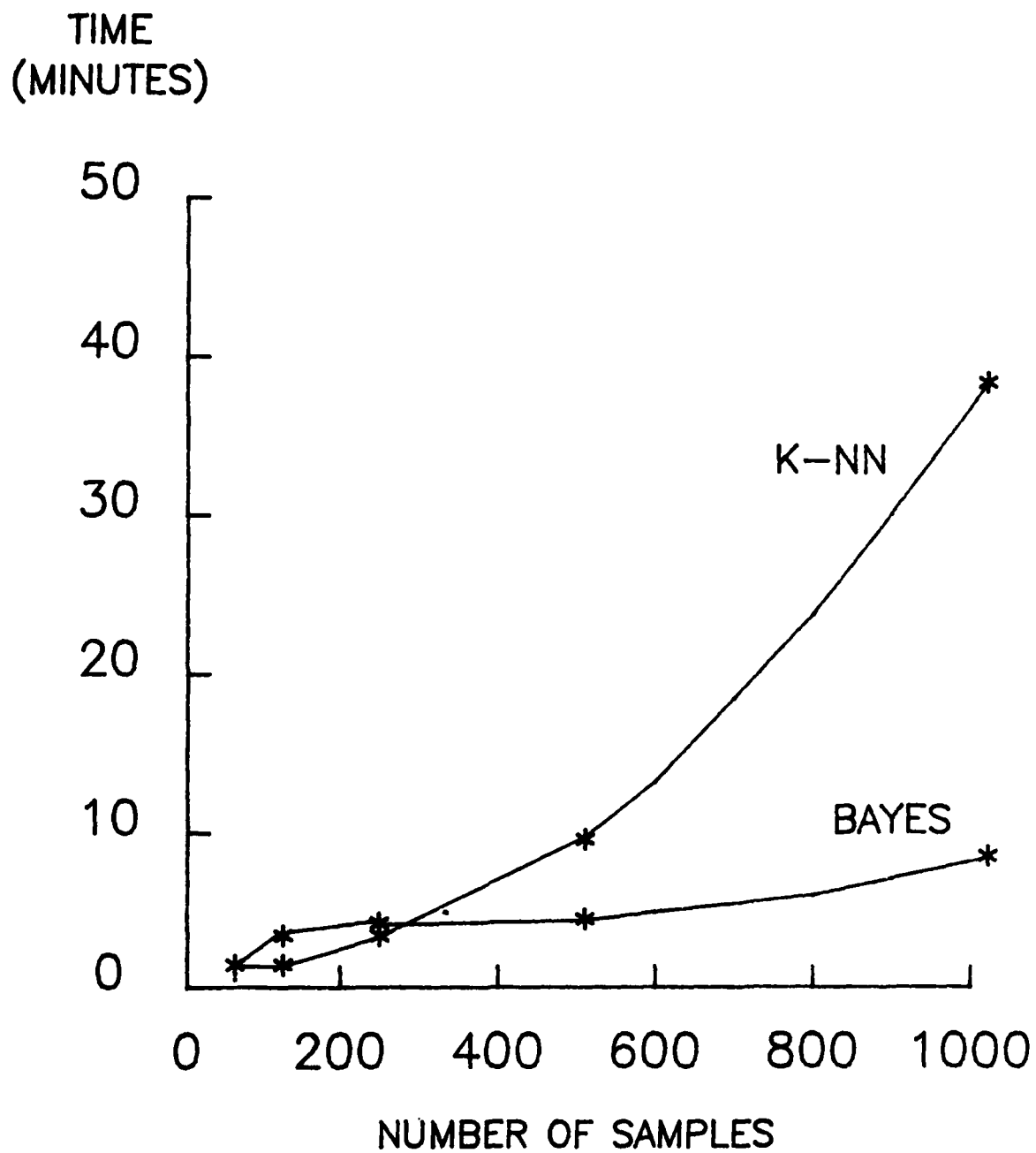


FIGURE 6.0.1 EXECUTION TIME

Figure 6.0.2(a) Bayes (**.5)
Error Map

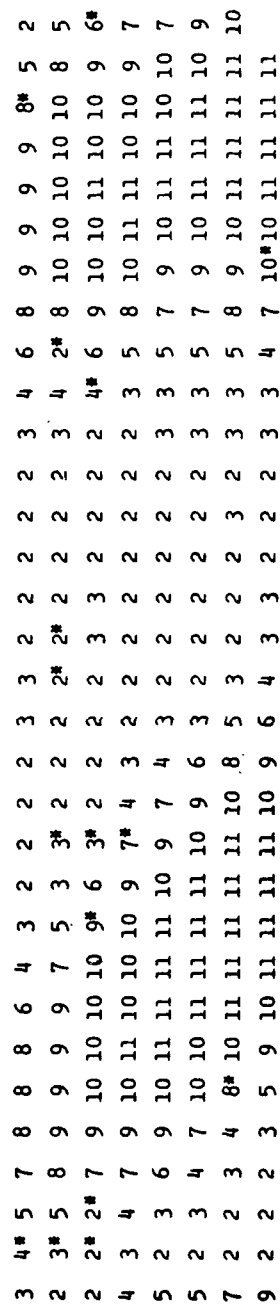


**Figure 6.0.2(b) 1-NN
Error Map**

SUB-APPENDIX E-A

See Appendix A

Bayes Classifier Algorithm

(see pp. 64-69)

SUB-APPENDIX E-B

k-NN Algorithm

KNN

This routine performs the K-Nearest Neighbor classification for category-type data, where $K = 1, 3, 4, 5, 6, 7, 8, 9, 10$. "Nearness" is defined on the basis of the interpattern distances.

<u>WORD</u>	<u>DEFAULT</u>	<u>DESCRIPTION</u>
NIN	(I ORIG)	Input unit.

Prerequisites: The distance matrix must be present on NIN. See DIST. Must have category-type data.

References: T. M. Cover and P. E. Hart, IEEE Trans. on Info. Theory, IT-13, 21 (1967).

DEFINITIONS OF TERMS FOR kNN

1. 1-NN

the category of the pattern closest to the pattern being classified (smallest $D_{i,j}$, $i \neq j$).

2. COMMITTEE VOTES (K-NN, $K=3,4,5,6,7,8,9,10$)

the category which is represented most often in the K-closest patterns to the pattern being classified.

In cases where two or more categories are equally represented, the category which has the smallest sum-of-distances.

3. TOTAL MISSED

a. TRAINING SET:

For the given K-NN, the total number of patterns which were misclassified.

b. TEST/PREDICTION SET:

For the given K-NN, the total number of patterns which were not classified as the category indicated (i. e. , no distinction made between TEST and PREDICTION set patterns.)

4. PERCENT CORRECT

a. TRAINING SET:

$$\% = (\text{NPAT} - \text{\#missed})(100.0) / \text{NPAT}$$

b. TEST/PREDICTION SET:

$$\% = (\text{NTEST} - \text{\#missed})(100.0) / \text{NTEST}$$

IMPLEMENTATION

1. Subroutines:

KNN

INPUKN: input

MAINKN: driver

OUTIKN: output, header

SORTKN: sorts out nearest 10 neighbors

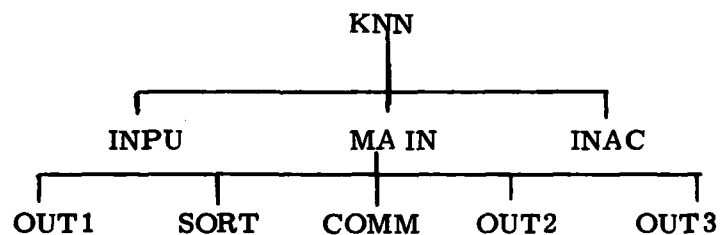
COMMKN: committee votes

OUT2KN: output, pattern results

OUT3KN: output, result summation

INACKN: interactive terminal driver

2. Organization:



SUB-APPENDIX E-C

Tabulation of Error Runs (x, y, Amplitude)

$P_{e_i}(\pi)$ For 60 Samples

Run #	P_e for Bayes				P_e for KNN			
	Ln	** 5	**1	**2	1 st NN	3 rd NN	7 th NN	10 th NN
1	1.0	1.0	1.0	1.0	.75	.75	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	.50	.5000	1.000	1.000	.2500	.5000	.7500	.7500
4	.5000	.5000	.7500	.7500	.2500	.2500	.500	.7500
5	1.000	1.000	.7500	.7500	.5000	.7500	1.000	1.000
6	1.000	1.000	1.000	1.000	.2500	.2500	.7500	1.000
7	.7500	.7500	1.000	1.000	.7500	.7500	.5000	.7500
8	1.000	1.000	1.000	1.000	.000	1.000	.2500	.000
9	1.000	.7500	.7500	.7500	.2500	.2500	.7500	.7500
10	.75	1.000	1.000	1.000	1.000	1.000	.000	.7500
11	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
12	.7500	1.000	.7500	.7500	.7500	.7500	.7500	.7500
13	.7500	2.500	.2500	.2500	.2500	.2500	.2500	.2500
14	.7500	.7500	.7500	1.000	.2500	.5000	.5000	.7500
15	.7500	1.000	1.000	1.000	.5000	.7500	1.000	1.000
	12.75	12.5	13.25	13.25	7.75	9.5	10.0	11.5

$\hat{P}_{e_i} [\pi]$ For 135 Samples

Run #	P_e for Bayes				P_e for KNN			
	Ln	** .5	**1	**2	1 st NN	3 rd NN	7 th NN	10 th NN
1	.7778	.5556	.4444	.5556	.000	.1111	.1111	.3333
2	.5556	.5556	.5556	.5556	.1111	.2222	.2222	.2222
3	.4444	.5556	.4444	.5556	.1111	.2222	.000	.000
4	.8889	.6667	.6667	.7778	.000	.1111	.1111	.1111
5	.4444	.4444	.4444	.4444	.2222	.2222	.2222	.2222
6	.7778	.4444	.4444	.4444	.000	.000	.000	.000
7	.5556	.1111	.2222	.2222	.000	.1111	.2222	.2222
8	.5556	.2222	.3333	.3333	.1111	.1111	.1111	.1111
9	.7778	.8889	.8889	.8889	.000	.000	.000	.000
10	.8889	.4444	.4444	.3333	.000	.000	.000	.000
11	.6667	.5556	.4444	.5556	.000	.000	.1111	.1111
12	.7778	.2222	.2222	.2222	.000	.1111	.1111	.1111
13	.7778	.3333	.3333	.3333	.1111	.1111	.2222	.2222
14	.8889	.4444	.5556	.5556	.000	.000	.000	.000
15	.8889	.3333	.3333	.3333	.000	.000	.000	.000

10.6669 6.7777 6.7775 7.1111 .6666 1.3332 1.4443 1.6665

$\hat{P}_e, [\pi]$ For 255 Samples

Run #	P_e for Bayes				P_e for KNN			
	Ln	** .5	**1	**2	1 st NN	3 rd NN	7 th NN	10 th NN
1	.4706	.2941	.3529	.3529	.0588	.1176	.0588	.588
2	.5294	.2941	.2941	.4706	.2353	.2353	.1176	.1176
3	.5294	.3529	.4118	.4706	.1176	.1765	.1176	.1176
4	.5882	.4118	.4118	.4118	.1765	.2353	.2353	.2353
5	.5882	.3529	.2941	.2941	.1176	.000	.0588	.0588
6	.8824	.2941	.2941	.2941	.1176	.1176	.0588	.0588
7	.7059	.3529	.2941	.3529	.0588	.0588	.000	.000
8	.8235	.5924	.6471	.6471	.000	.000	.000	.000
9	.7647	.1765	.1765	.2353	.000	.000	.000	.000
10	.8235	.2941	.1176	.1176	.000	.000	.0588	.0588
11	.7059	.2353	.2353	.2941	.0588	.0588	.0588	.0588
12	.7647	.2353	.1765	.1765	.0588	.0588	.0588	.0588
13	.7059	.3529	.2941	.2941	.000	.000	.000	.000
14	.5882	.1765	.1765	.1765	.000	.000	.000	.000
15	.7647	.2353	.2941	.3529	.0588	.0588	.000	.000

10.235 4.5881 4.4706 4.9464 1.0588 1.1175 .8233 .8233

$\hat{P}_e [\pi]$ For 510 Samples

Run #	P_e for Bayes				P_e for KNN			
	Ln	**5	**1	**2	1 st NN	3 rd NN	7 th NN	10 th NN
1	.2941	.2353	.3235	.3824	.000	.0588	.0588	.0588
2	.5294	.3529	.5000	.5000	.0882	.0588	.0882	.0882
3	.0882	.0294	.0588	.1471	.0294	.0882	.0882	.0882
4	.1176	.4118	.5000	.4706	.000	.0294	.0294	.1176
5	.4706	.2647	.5294	.6174	.0588	.0296	.000	.0294
6	.2941	.2941	.5882	.6176	.0882	.1176	.1176	.1176
7	.0588	.2353	.2941	.3235	.0294	.0296	.0296	.0296
8	.2059	.4118	.3824	.4412	.0296	.0588	.0296	.0588
9	.2941	.3826	.5588	.7353	.0882	.0588	.0588	.0588
10	.7059	.4706	.6176	.6671	.1176	.0588	.0296	.0294
11	.7941	.7059	.6671	.6671	.0822	.0296	.000	.000
12	.7059	.4706	.5000	.4118	.000	.000	.0588	.0588
13	.6765	.4706	.4412	.4706	.0296	.0296	.0296	.0294
14	.7353	.2059	.4612	.4612	.000	.000	.000	.000
15	.1176	.1471	.1765	.1765	.000	.000	.000	.000

6.0881 5.0884 7.5764 7.1296 .6438 .6408 .6174 .7644

$\hat{P}_e [\pi]$ For 1005 Samples

Run #	P_e for Bayes				P_e for KNN			
	Ln	** .5	**1	**2	1 st NN	3 rd NN	7 th NN	10 th NN
1	.4328	.2537	.3731	.3731	.1196	.1363	.1343	.1343
2	.2090	.1642	.3433	.3433	.0668	.0597	.0746	.0746
3	.3433	.3134	.4328	.4925	.0597	.0597	.0468	.0597
4	.1950	.2985	.2985	.2985	.0448	.0448	.0299	.0448
5	.3731	.3285	.4328	.4776	.1049	.0169	.0229	.0229
6	.1045	.2090	.2836	.3134	.0896	.0766	.0896	.0766
7	.4179	.2537	.5274	.5522	.0448	.0149	.000	.0149
8	.1642	.2388	.2537	.2537	.0299	.0299	.0296	.0468
9	.4030	.2836	.5821	.6567	.0149	.000	.0169	.000
10	.1363	.2985	.2985	.2985	.000	.000	.000	.00
11	.4179	.1306	.4058	.4627	.0299	.0299	.0296	.0299
12	.1940	.1791	.2239	.2537	.0149	.0149	.0468	.0468
13	.4030	.1791	.2388	.3286	.0149	.0169	.0149	.0149
14	.1065	.1960	.2388	.2687	.0149	.000	.0448	.0148
15	.4328	.1791	.3731	.5226	.000	.000	.000	.000
	4.3283	3.5035	5.3062	5.9999	.5374	.4925	.5753	.605

REFERENCES

1. Watanabe, Satoshi, Methodologies of Pattern Recognition, Academic Press, 1969.
2. Fukunaga, K., Introduction to Statistical Pattern Recognition, Academic Press, 1969.
3. Fu, Mendel, Adaptive Learning and Pattern Recognitive Systems, Academic Press, 1970.
4. Fu, K. S., Learning Systems, The American Society of Mechanical Engineers, 1973.
5. Andrews, H. C., Introduction to Mathematical Techniques in Pattern Recognition, Wiley, 1972.
6. Cover, T. M., Hart, P. E., Nearest Neighbor Pattern Classification, IEEE Trans. on Info. Theory, Vol. 13, pp. 21-27, Jan. 1967.
7. Toussaint, G. T., Sharpe, P. M., "An Efficient Method for Estimating the Probability of Misclassification Applied to a Problem in Medical Diagnosis", Comput. Biol. Med., Vol. 4, pp. 269-278, 1975.
8. Dudes, R., Jain, A. K., "Clustering Technique: The Users Dilemma", Pattern Recognition, 8, pp. 247-260, 1967.
9. Duda, R. O., Hart, P. E., Pattern Classification and Scene Analysis, John Wiley and Son, 1973.
10. Fordon, W. A., Computer Aided Differential Diagnosis of Hypertension, Ph. D. Dissertation, Purdue University, School of EE, 1973.
11. Gonzales, R. C., Tou J. T., Pattern Recognition Principles, Addison-Wesley, 1976.
12. Duran, B. S., Odell, P. L., Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 1974.
13. Fukunaga, K., Kessell, D. L., "Estimation of Classification Error", IEEE Trans. Comp., Vol. 20, 1521-1527, 1971.

14. Cover, T. M. , "Estimation by the Nearest Neighbor Rule", IEEE Trans. Info. Theory, Vol. 14, pp. 50-55, 1968.
15. Difranto, J. V. , Rubin, W. L. , Radar Detection, Prentice-Hall, Inc. , 1968.
16. Liedtke, C. E. , Eggers, D. , Arthur, University of Minnesota, 1975.
17. Roussas, G. G. , A First Course in Mathematical Statistics, Addison-Wesley Publishing Company, 1973.

*MISSION
of
Rome Air Development Center*

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C³I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

**DATA
FILM**