130 Title

COMPONENT PART NOTICE

THIS PAPER IS A COMPONENT PART OF THE FOLLOWING COMPILATION REPORT:

TITLE: Computing Science and Statistics: Proceedings of the Symposium on Interface Critical Applications of Scientific Computing (23rd): Biology, Engineering, Medicine, Speech Held in Seattle, Washington on 21-24 April 1991.

TO ORDER THE COMPLETE COMPILATION REPORT, USE AD-A252 938.

THE COMPONENT PART IS PROVIDED HERE TO ALLOW USERS ACCESS TO INDIVIDUALLY AUTHORED SECTIONS OF PROCEEDING, ANNALS, SYMPOSIA, ETC. HOWEVER, THE COMPONENT SHOULD BE CONSIDERED WITHIN THE CONTEXT OF THE OVERALL COMPILATION REPORT AND NOT AS A STAND-ALONE TECHNICAL REPORT.

THE FOLLOWING COMPONENT PART NUMBERS COMPRISE THE COMPILATION REPORT:

AD#: AD-P007 096 thru AD#: AD-P007 225

AD#: AD#:

AD#: AD#:

Exact Stratified Linear Rank Tests for Binary Data

Cyrus R. Mehta† Nitin Patel‡ Pralay Senchaudhuri†

† Department of Biostatistics, Harvard School of Public Health
and
‡ Indian Institute of Management, Ahmadebad, India

Abstract

We present an efficient network algorithm for generating exact permutational distributions for linear rank tests defined on stratified $2 \times c$ contingency tables. The algorithm can evaluate exact one and two sided p-values, and compute exact confidence intervals for trend parameters arising from certain loglinear and logistic models embedded in these contingency tables. It is especially efficient for highly imbalanced categorical data, a situation where the asymptotic theory is unreliable. Part of the algorithm can be adapted to evaluating the conditional maximum likelihood and its derivatives for the logistic regression model, with grouped data. We illustrate the techniques with an analysis of two data sets; the leukemia data on the Hiroshima atomic bomb survivors, and data from a clinical trial of bone marrow transplant.

1 Introduction

Linear rank tests play a major role in nonparametric inference. The Chernoff-Savage theorem (1958) ensures the asymptotic normality of these tests, and indeed, for continuous data the asymptotic results work very well. By the time the sample size is around 30, there is very little difference between the asymptotic distribution of a linear rank test statistic and its exact permutational distribution. However this is not the case for categorical data. Here the rate of convergence to asymptotic normality depends on more than just sample size. The number of ties in each category, the group imbalance, and the choice of rank scores, all affect the shape of the permutation distribution in complicated ways, making it difficult to predict a priori whether the asymptotic results for a given data set are reliable. It is important therefore to develop efficient numerical algorithms to supplement existing asymptotic results for the categorical case. These algorithms serve both the data analyst concerned about the validity of the inference in small, sparse, or imbalanced data sets, and the theoretical statistician developing new asymptotic methods and wishing to confirm that the theory is accurate.

This paper develops a very fast algorithm for generating exact permutation distributions for linear rank tests defined on stratified $2 \times c$ contingency tables. The permutational problem is formulated very precisely in Section 2. A network algorithm for solving the problem is presented in Section 3. A major strength of the algorithm is that its limits of computational feasibility increase with the degree of imbalance between the groups being compared. This is precisely where it is needed most, since the reliability of asymptotic results decrease as the imbalance increases. In another paper we analyze some case-control data in which the total sample size is 99,960. Yet, because of the severe imbalance between cases and controls, the asymptotic results differ from the exact ones. The algorithm developed here performs exact permutational inference on the data set with no difficulty whatsoever, despite its enormous sample size.

The inference techniques discussed in this paper are conditional. This is true both for the exact as well as the asymptotic inference. Exact methods for parameter estimation naturally require strong numerical algorithms. But it is not generally recognized that conditional inference places a heavy computational burden on the maximum likelihood estimation as well. A by-product of the algorithmic development in Section 3 is its applicability to the problem of estimating model parameters by maximizing a conditional likelihood function and evaluating its first two derivatives. Without our algorithm, evaluating the conditional likelihood, even though it only yields asymptotic estimates, would be almost as difficult as the

exact inference.

2 Statistical Formulation

In this section we formulate a general permutation problem whose solution will make exact statistical inference possible for a rich class of linear rank tests, defined on ordered categorical or binary data. The computational difficulties encountered with the permutation problem are discussed, setting the stage for the development of an efficient numerical algorithm, in Section 3.

2.1 Tabular Representation of the Data

The data can be represented as a collection of s $2 \times c$ contingency table consisting of 2 rows, c columns, and s strata. A specific collection, or three way table, of this type, denoted by $\underline{\mathbf{x}} \equiv (\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_s)$, is displayed below:

$\mathbf{x}_1 =$

Stratum 1					
Rows	Col_1	Col_2	...	Col_c	Row-Total
Row_1	x_{11}	x_{21}	...	x_{c1}	m_1
Row_2	x'_{11}	x'_{21}	...	x'_{c1}	m'_1
Col-Total	n_{11}	n_{21}	...	n_{c1}	N_1
Col-Score	w_1	w_2	...	w_c	

$\mathbf{x}_2 =$

Stratum 2					
Rows	Col_1	Col_2	...	Col_c	Row-Total
Row_1	x_{12}	x_{22}	...	x_{c2}	m_2
Row_2	x'_{12}	x'_{22}	...	x'_{c2}	m'_2
Col-Total	n_{12}	n_{22}	...	n_{c2}	N_2
Col-Score	w_1	w_2	...	w_c	

$\vdots$

$\mathbf{x}_s =$

Stratum s					
Rows	Col_1	Col_2	...	Col_c	Row-Total
Row_1	x_{1s}	x_{2s}	...	x_{cs}	m_s
Row_2	x'_{1s}	x'_{2s}	...	x'_{cs}	m'_s
Col-Total	n_{1s}	n_{2s}	...	n_{cs}	N_s
Col-Score	w_1	w_2	...	w_c	

The above tabular representation accommodates both the comparison of two multinomial populations and the comparison of k binomial populations. In either case we may adjust for possible covariate effects by stratification. Unstratified data may be regarded as a special case with $s = 1$.

Two Multinomial Populations The two rows of stratum k represent two independent multinomial populations. Each observation falls into one of c ordinal response categories. Thus x_{jk} is the number of stratum k observations, out of a total of m_k, falling into ordered category j for population 1, and x'_{jk} is the number of stratum k observations, out of a total of m'_k, falling into ordered category j for population 2. The stratum invariant scores, $w_1, w_2, \ldots w_c$, are numerical values assigned to the c ordered multinomial response categories.

Several Binomial Populations The c columns of stratum k represent c independent binomial populations with row 1 representing successes and row 2 representing failures. For population j and stratum k there are x_{jk} successes and x'_{jk} failures in n_{jk} independent Bernoulli trials. The stratum invariant scores, $w_1, w_2, \ldots w_c$ typically represent doses, or levels of exposure, affecting the success rates of the c binomial populations.

2.2 Exact Conditional Inference

Define the reference set for the kth stratum, Γ_k, as all possible $2 \times c$ contingency tables whose row and column margins are fixed at the corresponding values of the observed $2 \times c$ table, $\mathbf{x}_k$:

$$\Gamma_k = \{\mathbf{y}_k\text{: } \mathbf{y}_k \text{ is } 2 \times c;\ y_{jk} + y'_{jk} = n_{jk}, \forall j;$$

$$\sum_{j=1}^{c} y_{jk} = m_k, \sum_{j=1}^{c} y'_{jk} = m'_k\} \ .$$

Define the full reference set as the cartesian product of the reference sets across all s strata:

$$\Theta = \Gamma_1 \times \Gamma_2 \times \ldots \times \Gamma_s = \{\underline{\mathbf{y}}\text{: } \mathbf{y}_k \in \Gamma_k, k = 1, 2, \ldots s\ \} \ .$$

The test statistic, T, is defined as a sum of linear rank statistics over the s strata:

$$T = T_1 + T_2 + \ldots + T_s \ ,$$

where each T_k can only take on the values t_k of the form

$$t_k = \sum_{j=1}^{c} w_j y_{jk} \ ,$$

for some $\mathbf{y}_k \in \Gamma_k$, and a fixed set of scores, $w_1, w_2, \ldots w_c$. By a suitable choice of scores one can obtain a very rich class of linear rank tests. The distribution of the test

statistic, T, is derived by limiting the sample space to $\underline{y} \in \Theta$.

Under the null hypothesis of no row and column interaction the conditional probability distribution of T_k given $\mathbf{y}_k \in \Gamma_k$ is

$$f_k(t_k) = \frac{\sum_{\mathbf{y}_k \in \Gamma_{k,t_k}} \prod_{j=1}^{c} \binom{n_{jk}}{y_{jk}}}{\binom{N_k}{m_k}}, \qquad (2.1)$$

where

$$\Gamma_{k,t_k} = \{\mathbf{y}_k \in \Gamma_k : \sum_{j=1}^{c} w_j y_{jk} = t_k\} .$$

Then by convolution, the conditional probability distribution of T, given $\underline{y} \in \Theta$, is

$$f(t) = \frac{\sum_{\underline{y} \in \Theta_t} \prod_{k=1}^{s} \prod_{j=1}^{c} \binom{n_{jk}}{y_{jk}}}{\prod_{k=1}^{s} \binom{N_k}{m_k}}, \qquad (2.2)$$

where

$$\Theta_t = \{\underline{y} \in \Theta : \sum_{k=1}^{s} \sum_{j=1}^{c} w_j y_{jk} = t\} .$$

Notice that (2.2) is a sum of generalized hypergeometric probabilities and is free of all unknown parameters. This enables us to compute exact p-values for all the linear rank tests listed above. We can also compute the first two moments of T and thereby perform asymptotic inference by appealing to the Chernoff-Savage theorem.

2.3 Parameter Estimation

For data arising from two multinomial distributions or c binomial distributions, we can specify loglinear and logistic models, respectively, for the data generating process. Let π_{jk} be the probability that a subject from stratum k is classified as falling into row 1 and column j. Let π'_{jk} be the probability that a subject from stratum k is classified as falling into row 2 and column j. If the two rows of each stratum represent data from two multinomial populations, the above probabilities must satisfy the constraints

$$\sum_{j=1}^{c} \pi_{jk} = \sum_{j=1}^{c} \pi'_{jk} = 1 ,$$

for $k = 1, 2, \ldots s$. If the c columns of each stratum represent data from c binomial populations, the above probabilities must satisfy the constraints

$$\pi_{jk} + \pi'_{jk} = 1 ,$$

for $j = 1, 2, \ldots c$, and $k = 1, 2, \ldots s$. In either case we assume that there is no three-factor interaction so that the $c - 1$ odds ratios

$$\Psi_j = \frac{\pi_{jk}\pi'_{1k}}{\pi_{1k}\pi'_{jk}} ,$$

$j = 2, 3, \ldots c$, do not depend on k. Next we model these odds ratios as a function of the scores. If the data have been generated from two stratified multinomial populations, it is natural to derive the odds ratios from a log-linear model with a linear by linear row times column association (Agresti, 1990, page 275, equation (8.11)). In the present context the linear by linear model specifies the following expected cell counts on the logarithmic scale:

$$\log(m_k \pi_{jk}) = \alpha_{jk} + \beta w_j$$

for row 1, and

$$\log(m'_k \pi'_{jk}) = \alpha_{jk}$$

for row 2.

If the data have been generated from c stratified binomial populations it is natural to derive the odds ratios from a logistic regression model (Cox, 1970):

$$\log \frac{\pi_{jk}}{\pi'_{jk}} = \alpha_k + \beta w_j .$$

Both models yield the relationship

$$\log \Psi_j = \beta(w_j - w_1) , \qquad (2.3)$$

where β is an unknown parameter to be estimated from the data. It can be shown that T is a sufficient statistic for β under both the linear by linear association model and the logistic regression model. Moreover, the conditional distribution of T, given $(\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_s) \in \Theta$, depends only on β, other (nuisance) parameters being eliminated by the conditioning. This conditional distribution is given by

$$f(t|\beta) = \frac{f(t)\exp(\beta t)}{\sum_u f(u)\exp(\beta u)}, \qquad (2.4)$$

where the denominator of equation (2.4) is simply the normalizing constant obtained by summing over all possible values of T. When $\beta = 0$ we obtain the null distribution (2.2).

The conditional maximum likelihood estimate (cmle) of β is obtained by finding the value of $\hat{\beta}$ that maximizes the conditional probability (2.4) at the observed value $T = a_0$. To obtain the variance of the cmle we need the second derivative of the log likelihood, evaluated at the cmle. Both the cmle and its variance may be rapidly evaluated by repeated backward induction on a network, as discussed in detail in Section 3. We can then use these estimates to perform asymptotic hypothesis tests or compute asymptotic confidence intervals for β.

To obtain an exact confidence interval for β we need the coefficients $f(t)$ for all values of T in the tails its distribution. A network algorithm for this computation is described in Section 3. Once these coefficients have been computed, the conditional tail probabilities, $T \geq a_0$, or $T \leq a_0$, for any value of β, may be derived from equation (2.4). Exact confidence bounds for β are then obtained by inverting corresponding UMP unbiased tests for β, as shown in Cox (1970). For example, a 100(1 - α)% lower confidence bound for β, say $\beta(a_0)$, would be obtained as the solution to

$$\sum_{t=a_0}^{t_{max}} f(t|\beta(a_0)) = \alpha \ . \qquad (2.5)$$

The solution to equation (2.5) may be rapidly evaluated by a simple binary search because, as shown in (2.4), $f(t)$ and β are separable in the expression for $f(t|\beta)$.

2.4 Computational Issues

From the above discussion it is clear that a broad class of exact linear rank tests and parameter estimates can be obtained if we are able to compute truncated distributions of the form

$$\Omega = \{(t, f(t)): \ t \geq a_0\} \ . \qquad (2.6)$$

Exhaustive enumeration of all the tables in Θ for generating Ω would be computationally explosive. Consider the simple case of a single stratum, no ties, and $m = m' = N/2$. The number of tables in the reference set Θ for various values of N is

Sample Size (N)	Tables in Reference Set (Γ)
20	1.8×10^5
30	1.5×10^8
40	1.4×10^{11}
50	1.3×10^{14}
100	1.0×10^{29}

If there were s strata, the size of the corresponding reference set would be raised to the sth power. It is clear that even in the very powerful computing environment available today, explicit enumeration of all the tables in the reference set Θ rapidly becomes computationally infeasible. However much recent research, for example, Mehta et. al. (1984) (1985) (1988), Pagano and Tritchler (1983), Tritchler (1984), Streitberg and Rohmel (1986), and Hollander and Pena (1988), has focused on implicit enumeration of the tables in Θ, thereby considerably extending the size of problem for which exact inference is possible.

Mehta, Patel and Tsiatis (1984), and Mehta, Patel and Wei (1988), developed a network algorithm for implicit enumeration of all the $2 \times c$ contingency tables in the reference set Γ, defined for a single stratum ($s = 1$). Mehta, Patel and Gray (1985) developed a network algorithm for implicit enumeration of s 2×2 contingency tables (where $s > 1$). The present paper generalizes the earlier work to s independent $2 \times c$ contingency tables, a considerably more difficult problem. An alternative method would be to treat the s $2 \times c$ problem as a special case of conditional logistic regression and directly use the exact algorithm of Hirji, Mehta and Patel (1988). However that would not exploit the special structure of the problem in the way that the present algorithm does. We conjecture that the algorithm presented here is the fastest one currently available for categorical data, with unequally spaced w_j scores. In another paper we perform exact inference on some rather large data sets, to illustrate how powerful the algorithm is, and to set up a benchmark against which competing algorithms may be evaluated.

A second contribution of this paper is to provide an efficient numerical algorithm for computing the cmle for β (equation 2.3) and its standard error. A previous algorithm for this problem, in the more general conditional logistic regression setting, was developed by Gail, Lubin, and Rubenstein (1981). Our algorithm is equivalent to theirs for data with no ties, but is considerably more efficient for categorical data. In another paper, we show that the Gail et. al., algorithm, as implemented in the EGRET (1988) software package, is unable to compute conditional maximum likelihood estimates for a large heavily tied data set, whereas our algorithm, obtains the required estimates very rapidly.

3 Numerical Algorithms

We provide numerical algorithms for two problems; generating the truncated permutation distribution Ω, defined by (2.6), and computing the cmle for β, say $\hat{\beta}$, along with its standard error, $\hat{\sigma}$. Both problems are solved within one unified framework wherein the reference set Θ is represented as a network. We will see that processing the network in the forward direction yields Ω, while processing the same network in the backward direction yields $\hat{\beta}$ and its standard error.

3.1 Generating an Overall Truncated Permutation Distribution

Our goal is to generate the truncated permutation distribution Ω for T, the sum of linear rank statistics across all the strata. Our strategy will be to generate s independent stratum specific truncated permutation distributions of the form

$$\Omega_k = \{(t_k, f_k(t_k)) : t_k \geq a_k\} ,$$

at the cut-off points

$$a_k = a_0 - \sum_{i \neq k} t_{i,max} ,$$

for, $k = 1, 2, \ldots s$. Here $t_{k,max}$ is the maximum value of the random variable T_k, and is easily evaluated as part of the backward induction step discussed below. We will perform pairwise convolutions on these stratum specific distributions until the overall distribution is obtained. Thus there are two steps to be performed repeatedly; a distribution generation step, and a convolution step. These steps are described next in separate subsections.

3.1.1 Generating Stratum Specific Truncated Permutation Distributions

Suppose we wish to generate the truncated permutation distribution Ω_k, for the kth stratum. In principle this involves enumerating all the $2 \times c$ contingency tables $\mathbf{y}_k \in \Gamma_k$, computing the value of $t_k = \sum_{j=1}^{c} w_j y_{jk}$ for each one, and summing the hypergeometric probabilities of all the tables $\mathbf{y}_k \in \Gamma_{k,t_k}$, as shown in (2.2). We do this enumeration implicitly rather than explicitly, by representing the reference set Γ_k as a network of nodes and arcs, and then processing the network in a recursive stage-wise fashion.

Network Representation of Γ_k

The network representation of the reference set, Γ_k, is constructed in $c+1$ stages labelled $0, 1, \ldots c$, where stage j corresponds to the jth column of a typical $2 \times c$ table in Γ_k. At stage j there exist a set of nodes of the form (j, m_{jk}), where each $m_{jk} = \sum_{l=1}^{j} y_{lk}$ corresponds to one distinct partial sum of the first j columns of the tables $\mathbf{y}_k \in \Gamma_k$. Arcs emanate from each node (j, m_{jk}) and connect it to successor nodes of the form $(j+1, m_{j+1,k})$. These successor nodes may be specified explicitly as the set

$$\mathbf{R}(j, m_{jk}) = \{(j+1, m_{j+1,k}): \max(m_{jk}, m_k - \sum_{l=j+2}^{c} n_{lk}) \leq m_{j+1,k} \leq \min(m_{jk} + n_{j+1,k}, m_k)\} . \quad (3.7)$$

Starting at stage 0 with initial node $(0,0)$, and applying (3.7) successively to the nodes at stages $1, 2, \ldots c-1$, we automatically end up with the unique terminal node (c, m_k). In this construction each path, or sequence of connected arcs of the form

$$(0,0) \rightarrow (1, m_{1k}) \rightarrow \cdots \rightarrow (c, m_k) \quad (3.8)$$

corresponds to one and only one table $\mathbf{y}_k \in \Gamma_k$, with $y_{jk} = m_{jk} - m_{j-1,k}$, for $j = 1, 2, \ldots c$. Thus the tables in Γ_k are in one-to-one correspondence with the paths through the network.

To complete the network representation we assign to each arc

$$(j-1, m_{j-1,k}) \rightarrow (j, m_{jk})$$

a **rank length**

$$r_{jk} = w_j(m_{jk} - m_{j-1,k})$$

and a **probability length**

$$p_{jk} = \binom{n_{jk}}{m_{jk} - m_{j-1,k}} \exp(\beta r_{jk}) . \quad (3.9)$$

The rank length of a complete path of the form (3.8) connecting the initial node to the terminal node is defined as the sum of rank lengths of the individual arcs constituting that path. Its probability length is the product of probability lengths of the individual arcs constituting that path. The distribution of T_k is then the same as the distribution of rank lengths of all the paths in Γ_k.

Backward Induction on Γ_k

We can obtain much useful information about the distribution of T_k very quickly, by a single backward pass through the network Γ_k. At any node (j, m_{jk}) define

the sub-network, $\Gamma_k(j, m_{jk})$, to be the set of all possible paths from (j, m_{jk}) to the terminal node (c, m_k). In other words $\Gamma_k(j, m_{jk})$ consists of all possible values of the entries in columns $(j+1, j+2, \ldots, c)$ of the $2 \times c$ contingency tables in Γ_k whose first j columns sum to m_{jk}. Now define the length of the longest path in $\Gamma_k(j, m_{jk})$ by

$$LP(j, m_{jk}) = \max_{\Gamma_k(j,m_{jk})} \{ \sum_{l=j+1}^{c} r_{lk} \} , \tag{3.10}$$

the length of the shortest path in $\Gamma_k(j, m_{jk})$ by

$$SP(j, m_{jk}) = \min_{\Gamma_k(j,m_{jk})} \{ \sum_{l=j+1}^{c} r_{lk} \} , \tag{3.11}$$

and the sum of probability lengths of all the paths in $\Gamma_k(j, m_{jk})$ by

$$TP(j, m_{jk}) = \sum_{\Gamma_k(j,m_{jk})} \prod_{l=j+1}^{c} p_{lk} . \tag{3.12}$$

The values of LP, SP, and TP can be rapidly obtained by backward induction. We illustrate how this is done for LP. Set $LP(c, m_k) = 0$. Now suppose that $LP(j+1, m_{j+1,k})$ is known for every node at stage $j+1$. Move backwards to stage j, select a node $(j, m_{j,k})$, and compute

$$LP(j, m_{jk}) = \max_{\mathbf{R}(j,m_{jk})} \{ r_{j+1,k} + LP(j+1, m_{j+1,k}) \} . \tag{3.13}$$

Repeat this process for every node at stage j and then move back one more stage. Proceeding in this manner we reach stage $(0, 0)$ having evaluated the LP values for all the nodes of the network. The other nodal quantities may be obtained similarly.

Processing Γ_k in the Forward Direction

Starting with the initial node (0,0), we process the network in the forward direction, stage by stage, in such a way that by the time we reach the terminal node, (c, m_k), we will have generated the desired truncated distribution Ω_k. First we introduce some notation. At any node (j, m_{jk}) define the sub-network, $\Upsilon_k(j, m_{jk})$, to be the set of all possible paths from the starting node (0,0) to (j, m_{jk}). In other words, $\Upsilon_k(j, m_{jk})$ consists of all possible values of the entries in columns $(1, 2, \ldots j)$ of the $2 \times c$ contingency tables in Γ_k whose first j columns sum to m_{jk}. (Notice that this set differs from $\Gamma_k(j, m_{jk})$, which specifies the last $c-j+1$ columns of these tables.) Denote a generic path,

$$(0,0) \to (1, m_{1k}) \to \cdots \to (j, m_{jk})$$

in $\Upsilon_k(j, m_{jk})$ by τ. The rank length of τ is

$$r(\tau) = \sum_{l=1}^{j} r_{lk} ,$$

and its probability length is

$$p(\tau) = \prod_{l=1}^{j} p_{lk} .$$

There will typically be several paths, $\tau \in \Upsilon_k(j, m_{jk})$, each having the same rank length, $r(\tau) = u$. Let $c(u)$ be the sum of probability lengths of all these paths. That is,

$$c(u) = \sum_{\{\tau \in \Upsilon_k(j,m_{jk}):\ r(\tau)=u\}} p(\tau) .$$

We now provide a recursive procedure for processing the network in the forward direction. Suppose we have reached stage j of the network in such a way that at each of its nodes, (j, m_{jk}), we are carrying a set of records

$$\Lambda(j, m_{jk}) = \{(u, c(u)):\ u = r(\tau), u + LP(j, m_{j,k}) \geq a_k,$$
$$\tau \in \Upsilon_k(j, m_{jk})\} .$$

The following five-step algorithm is used to update these sets and thereby move forward to stage $j+1$.

Step 1: Select a record $(u, c(u)) \in \Lambda(j, m_{jk})$.

Step 2: Transmit a copy of this record to each successor node $(j+1, m_{j+1,k})$, where the successors are identified by (3.7).

Step 3: At each successor node, $(j+1, m_{j+1,k})$, transform the transmitted record to (u^*, c^*), where $u^* = u + r_{j+1,k}$, and $c^* = c(u)p_{j+1,k}$.

Step 4: Insert (u^*, c^*) into $\Lambda(j+1, m_{j+1,k})$ as follows:

1. If $u^* + LP(j+1, m_{j+1,k}) < a_k$, drop this record from further consideration, and go to Step 5. Otherwise continue with the insertion as described below. (The value of LP is available from the backward induction on Γ_k.)
2. If there already exists a record $(u, c(u)) \in \Lambda(j+1, m_{j+1,k})$ such that $u = u^*$, then merge the two records by replacing $(u, c(u))$ with $(u, c(u) + c^*) \in \Lambda(j+1, m_{j+1,k})$.
3. If no record currently in $\Lambda(j+1, m_{j+1,k})$ has $u = u^*$, then augment $\Lambda(j+1, m_{j+1,k})$ by adding $(u, c(u))$ to it, as a new record.

The technique of hashing (Sedgewick 1983, page 201) is used to search for matches and either merge or augment records in $\Lambda(j+1, m_{j+1,k})$. This ensures an optimum trade-off between efficient use of available memory and fast search.

Step 5: Return to Step 1.

The above 5-step algorithm continues until every record in $\Lambda(j, m_{jk})$ has been processed. Then another node at stage j is selected, and all its records are processed in accordance with the above 5 steps. When all nodes at stage j have been exhausted, repeat Steps 1 through 5 for stage $j+1$. Starting with $\Lambda(0,0) = \{(0,1)\}$ and moving through stages $0, 1, \ldots c-1$ by repeatedly carrying out Steps 1 through 5, we process the entire Γ_k network, ending up at its terminal node with the set of records $\Lambda(c, m_k)$. These records are really the same as the desired truncated probability distribution Ω_k, except that the probability lengths, $c(u)$, have to be normalized by dividing by their sum. That is,

$$f_k(t_k) = \frac{c(t_k)}{\sum_u c(u)} .$$

3.1.2 Pairwise Convolution of the Stratum Specific Truncated Distributions

We restrict our discussion to the convolution of Ω_1 with Ω_2. The resultant distribution may be convolved with Ω_3 in exactly the same manner. We can go on with this pairwise convolution until we obtain Ω.

First sort the records of Ω_1 in ascending order of t_1, and the records of Ω_2 in descending order of t_2. Set $i = 1$, $j = 1$. Now proceed with the following 3-step algorithm:

Step 1: Select record i from Ω_1. Denote it by $(t_1^i, f_1(t_1^i))$. Select record j from Ω_2. Denote it by $(t_2^j, f_2(t_2^j))$.

Step 2: If

$$t_1^i + t_2^j + \sum_{k=3}^{s} t_{k,max} \geq a_0 ,$$

set $j = j + 1$, and return to Step 1. But if

$$t_1^i + t_2^j + \sum_{k=3}^{s} t_{k,max} < a_0 , \qquad (3.14)$$

convolve record i from Ω_1 with each of the first $j-1$ records from Ω_2.

Step 3: Set $i = i + 1$, and return to Step 1.

There are many ways to perform the convolution at Step 2, if the inequality (3.14) holds. We use hashing to club records having the same value of $t_1 + t_2$. The details are similar to Step 4.2 of the 5-step algorithm for forward processing of Γ_k. A considerable efficiency gain is achieved because we need not consider records from Ω_2 located at positions j or below. The inequality (3.14) ensures that they can never contribute to the final set of records in Ω, since the maximum to which they could be augmented is less than a_0. This is analogous to the record elimination achieved at Step 4.1 of the 5-step algorithm for forward processing of Γ_k.

3.2 Evaluating $\hat{\beta}$ and its Variance

To obtain $\hat{\beta}$, the cmle for β, we must maximize the logarithm of the likelihood (2.4). Then the second derivative of the log likelihood, evaluated at $\hat{\beta}$, yields the desired variance. But direct evaluation of the log likelihood is not an easy task, given the complicated expression for the denominator of (2.4). In fact if one attempted to evaluate this denominator directly, it would require the enumeration of all the s $2 \times c$ tables in Θ. This would make the asymptotic inference as computationally complex as the exact inference. Fortunately there is an easier approach that works well up to extremely large sample sizes. Notice that the denominator of (2.4) is the same as $TP(0,0)$, summed over all the strata. We can easily set up recursions like (3.13) for TP, its first derivative, TP', and its second derivative, TP'', and rapidly evaluate all three quantities during the backward induction

of Γ_k. For example,

$$TP'(j, m_{jk}) = \sum_{\mathbf{R}(j,m_{jk})} p_{j+1,k}[TP(j+1, m_{j+1,k}) + TP'(j+1, m_{j+1,k})]$$

It is easy to show by successive differentiation of the logarithm of (2.4) that the second derivative of the contribution to the log likelihood of the kth stratum is

$$[TP(0,0)]^{-2}[TP'(0,0)]^2 - [TP(0,0)]^{-1}[TP''(0,0)] \qquad (3.15)$$

Evaluating (3.15) at the cmle of β, summing across strata, and equating the resultant second derivative to zero, yields the desired asymptotic variance.

4 Concluding Remarks

The following technical features of the network algorithm were responsible for its extraordinary success:

- The network representation takes advantage of the categorical nature of the data by requiring only as many stages as there are discrete categories.
- The number of nodes in the Γ_k network is determined $\min(m_k, m'_k)$. Thus the greater the imbalance between the two row sums, the smaller the network, and the easier the processing.
- The preliminary backward induction pass through the network provides valuable information about the 'future' for each stage of the forward processing. This enables us to generate a truncated permutation distribution directly at the forward pass, rather than generating the full permutation distribution and then truncating it as needed. In effect, substantially fewer records are carried along at each stage of the forward pass, as records not satisfying the *LP* criterion get eliminated.
- The network representation enables us to generate the distribution of each T_k recursively in a stagewise forward pass through the network. During this forward pass paths having the same rank length up to some node are 'clubbed' together. We thus deal only with paths having distinct rank lengths up to each node, rather than all the paths up to that particular node.
- The backward induction step enables us to rapidly evaluate the denominator of (2.4), and its first and second derivatives. This greatly facilitates the conditional maximum likelihood inference.

References

Agresti A (1990). *Categorical Data Analysis*, Wiley & Sons, New York.

Chernoff H, Savage IR (1958). Asymptotic normality and efficiency of certain nonparametric test statistics. *Ann Math Stat* 29:972-994.

Cox DR (1970). *The Analysis of Binary Data.* Methuen, London.

Egret (1988). *State of the Art Epidemiological Computing.* Statistics and Epidemiological Research Corporation, Seattle, WA.

Gail MH, Lubin JH, Rubinstein LV (1981). Likelihood calculations for matched case-control studies and survival studies with tied death times. *Biometrika* 68:703-707.

Gail MH, Mantel N (1977). Counting the number of $r \times c$ contingency tables with fixed margins. *JASA* 72:859-862.

Hirji KF, Mehta CR, Patel NR (1988). Exact inference for matched case-control studies. *Biometrics* 44:803-814.

Hollander M, Pena D (1988). Nonparametric tests under restricted treatment assignment rules. *JASA* 83(404):1144-1151.

Landis R, Heyman ER, Koch GG (1978). Average partial association in three-way contingency tables: a review and discussion of alternative tests. *Int. Stat. Rev.* 46:237-254.

Mehta CR, Patel NR, Tsiatis AA (1984). Exact significance testing to establish treatment equivalence for ordered categorical data. *Biometrics* 40:819-825.

Mehta CR, Patel NR, Gray R (1985). On computing an exact confidence interval for the common odds ratio in several 2×2 contingency tables. *JASA* 80(392):969-973.

Mehta CR, Patel NR, Wei LJ (1988). Computing exact significance tests with restricted randomization rules. *Biometrika* 75(2):295-302.

Pagano M, Tritchler D (1983). On obtaining permutation distributions in polynomial time. *JASA* 78:435-441.

Sedgewick R (1983). *Algorithms.* Addison-Wesley, Reading, MA.

StatXact (1991). *A Statistical Package for Exact Nonparametric Inference: Version 2.* Cytel Software Corporation, Cambridge, MA.

Streitberg B, Rohmel R (1986). Exact distributions for permutation and rank tests. *Statistical Software Newsletter* 12:10-17.

Tritchler D (1984). An algorithm for exact logistic regression. *JASA* 79:709-711.