Sequential Monte Carlo Methods for Statistical Analysis of Tables

Yuguo CHEN, Persi DIACONIS, Susan P. HOLMES, and Jun S. LIU

We describe a sequential importance sampling (SIS) procedure for analyzing two-way zero–one or contingency tables with fixed marginal sums. An essential feature of the new method is that it samples the columns of the table progressively according to certain special distributions. Our method produces Monte Carlo samples that are remarkably close to the uniform distribution, enabling one to approximate closely the null distributions of various test statistics about these tables. Our method compares favorably with other existing Monte Carlo-based algorithms, and sometimes is a few orders of magnitude more efficient. In particular, compared with Markov chain Monte Carlo (MCMC)-based approaches, our importance sampling method not only is more efficient in terms of absolute running time and frees one from pondering over the mixing issue, but also provides an easy and accurate estimate of the total number of tables with fixed marginal sums, which is far more difficult for an MCMC method to achieve.

KEY WORDS: Conditional inference; Contingency table; Counting problem; Exact test; Sequential importance sampling; Zero–one table.

1. INTRODUCTION

1.1 Darwin's Finch Data

In ecology, researchers are often interested in testing theories about evolution and the competition among species. The zero–one table shown in Table 1 is called an *occurrence matrix* in ecological studies. The rows of the matrix correspond to species; the columns, to geological locations. A "1" or "0" in cell (i, j) represents the presence or absence of species i at location j. The occurrence matrix in Table 1 represents 13 species of finches inhabiting 17 islands of the Galápagos Islands (an archipelago in the East Pacific). The data are known as "Darwin's finches," because Charles Darwin collected some of these species when he visited the Galápagos. Darwin suggests in *The Voyage of the Beagle* that his observation of the striking diversity in these species of finches started a train of thought that culminated in his theory of evolution. [However, Sullaway (1982) showed that Darwin did not realize the significance of the finches until years after he visited the Galápagos.] Cook and Quinn (1995) cataloged many other occurrence matrices that have been collected. The ecological importance of the distribution of species over islands was described by Sanderson (2000) as follows: "Birds with differing beaks may live side by side because they can eat different things, whereas similarly endowed animals may not occupy the same territory because they compete with one another for the same kinds of food. Ecologists have long debated whether such competition between similar species controls their distribution on island groups or whether the patterns found simply reflect chance events in the distant past."

From a statistical standpoint, the null hypothesis that the pattern of finches on islands is the result of chance rather than competitive pressures can be translated to the statement that the observed zero–one table is a "typical" sample drawn uniformly from the set of all tables with the observed row and column marginal sums. The number of islands that each species inhabits (the row sums) and the number of species on each island (the column sums) are kept fixed under the null hypothesis to reflect the fact that some species are naturally more widespread than others and some islands are naturally more accommodating to a wide variety of species than others (Manly 1995; Connor and Simberloff 1979). For testing whether there is competition between species, Roberts and Stone (1990) suggested the test statistic

$$\bar{S}^2 = \frac{1}{m(m-1)} \sum_{i \neq j} s_{ij}^2, \tag{1}$$

where m is the number of species, $\mathbf{S} = (s_{ij}) = \mathbf{A}\mathbf{A}^T$, and $\mathbf{A} = (a_{ij})$ is the occurrence matrix. The null hypothesis is rejected if $\bar{S}^2$ is too large. Sanderson (2000) used the number of instances of two specific species living on the same island as the test statistic, which corresponds to focusing on two rows and counting the number of columns in which both of these rows contain a 1. More test statistics have been discussed by Connor and Simberloff (1979), Wilson (1987), Manly (1995), Sanderson, Moulton, and Selfridge (1998), and Sanderson (2000). Our methods apply to all of these approaches.

A difficult challenge in carrying out these tests is that there are no good analytic approximations to the null distributions of the corresponding test statistics. We show how to simulate the zero–one tables nearly uniformly, then adjust the samples using importance weights. We can thus obtain a good approximation to the null distribution of any test statistic, as well as an estimate of the total number of the zero–one tables that satisfy marginal constraints. Although several methods for generating tables from the uniform distribution conditional on marginal sums have been proposed in the literature, most of them are inefficient and some are incorrect (see Sec. 6.2).

1.2 Problem Formulation

From magic squares to Darwin's theory of evolution, problems of counting the total number of tables and testing hypotheses about them arise in many fields, including mathematics, statistics, ecology, education, and sociology. Although fixing the marginal sums makes these problems much more difficult,

Yuguo Chen is Assistant Professor, Institute of Statistics and Decision Sciences, Duke University, Durham, NC 27708 (E-mail: *yuguo@stat.duke.edu*). Persi Diaconis is Mary V. Sunseri Professor and Susan P. Holmes is Associate Professor (E-mail: *susan@stat.stanford.edu*), Department of Statistics, Stanford University, Stanford, CA 94305. Jun S. Liu is Professor, Department of Statistics and Department of Biostatistics, Harvard University, Cambridge, MA 02138 (E-mail: *jliu@stat.harvard.edu*). This work was supported in part by National Science Foundation grants DMS-02-03762, DMS-02-04674, and DMS-02-44638. The authors thank Arnab Chakraborty, Dylan Small, Charles Stein, the associate editor, and two referees for helpful discussions and valuable suggestions.

Journal of the American Statistical Association
March 2005, Vol. 100, No. 469, Theory and Methods
DOI 10.1198/016214504000001303

Table 1. Occurrence Matrix for Darwin's Finch Data

	Island																
Finch	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*
Large ground finch	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
Medium ground finch	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
Small ground finch	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
Sharp-beaked ground finch	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1
Cactus ground finch	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	0
Large cactus ground finch	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
Large tree finch	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0
Medium tree finch	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Small tree finch	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0
Vegetarian finch	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
Woodpecker finch	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0
Mangrove finch	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Warbler finch	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

NOTE: Island name code: A = Seymour, B = Baltra, C = Isabella, D = Fernandina, E = Santiago, F = Rábida, G = Pinzón, H = Santa Cruz, I = Santa Fe, J = San Cristóbal, K = Española, L = Floreana, M = Genovesa, N = Marchena, O = Pinta, P = Darwin, Q = Wolf.

it is important to do in many applications. For statistical applications in which the subjects are not obtained by a sampling scheme but are the only ones available to the researcher, conditioning on the marginal sums of the table creates a probabilistic basis for a test (Lehmann 1986, chap. 4.7). In some other applications, such as those related to the Rasch (1960) model, the marginal sums are sufficient statistics under the null hypothesis. Conditioning on the marginal sums is a way to remove the effect of nuisance parameters on tests (Lehmann 1986, chap. 4; Snijders 1991).

Because the interactions among the row and column sum constraints are complicated, no truly satisfactory analytical solutions or approximations are available for distributions of various test statistics (Snijders 1991). The table-counting problem is slightly more approachable analytically, although it is more challenging algorithmically. Several asymptotic methods have been developed for approximating the count of zero–one or contingency tables with fixed marginal sums (Békéssy, Békéssy, and Komlos 1972; Gail and Mantel 1977; Good and Crook 1977); however, these formulas are usually not very accurate for tables of moderate size. Wang (1988) provided an exact formula for counting zero–one tables, which was further improved by Wang and Zhang (1998). However, their exact formula is very complicated, and both of those authors (by personal communication) think that the formula would take too long to compute for Table 1, which is only of moderate size among our examples.

From a practical standpoint, if we can simulate tables from the uniform or a nearly uniform distribution, then we can both estimate the total count of the tables and approximate the distribution of any test statistic that is a function of the table. Several algorithms for generating uniform zero–one tables have been proposed (Connor and Simberloff 1979; Wilson 1987; Besag and Clifford 1989; Rao, Jana, and Bandyopadhyay 1996; Sanderson et al. 1998; Sanderson 2000; Cobb and Chen 2003), and an importance sampling idea has been suggested by Snijders (1991). Algorithms for generating contingency tables from the uniform distribution have also been developed (Balmer 1988; Boyett 1979; Patefield 1981), including a recent Markov chain Monte Carlo (MCMC) method by Diaconis and Gangolli (1995). Forster, McDonald, and Smith (1996) and Smith, Forster, and McDonald (1996) suggested a Gibbs sampling approach to sample multiway contingency tables when the underlying distribution is Poisson or multinomial. Holmes and Jones (1996) used the rejection method both to sample contingency tables from the uniform distribution and to estimate the total number of such tables with fixed margins. In our experience, however, all of these methods encounter difficulties for large, sparse tables and are especially vulnerable or ineffective when used to estimate the total number of tables.

We describe a sequential importance sampling (SIS) approach for approximating statistics related to the uniform distribution on zero–one and contingency tables with fixed margins. The distinctive feature of the SIS approach is that the generation of each table proceeds *sequentially* column by column and the partial importance weight is monitored along the way. Section 2 introduces the basic SIS methodology and the rules for evaluating the accuracy and efficiency of our estimates. Section 3 describes how we apply conditional-Poisson sampling together with the SIS for generating zero–one tables. Section 4 proposes a more delicate SIS method that is guaranteed to always generate proper tables. Section 5 generalizes the SIS method from zero–one tables to contingency tables. Section 6 shows some applications and numerical examples, including statistical evaluation of Table 1 and a count of the number of tables with the same row and column sums as Table 1, and Section 7 concludes with a brief discussion on the method.

2. SEQUENTIAL IMPORTANCE SAMPLING

Given the row sums $\mathbf{r} = (r_1, r_2, \ldots, r_m)$ and the column sums $\mathbf{c} = (c_1, c_2, \ldots, c_n)$, we let $\Sigma_{\mathbf{rc}}$ denote the set of all $m \times n$ (zero–one or contingency) tables with row sums $\mathbf{r}$ and column sums $\mathbf{c}$ (assuming that $\Sigma_{\mathbf{rc}}$ is nonempty). Let $p(\mathbf{T}) = 1/|\Sigma_{\mathbf{rc}}|$ be the uniform distribution over $\Sigma_{\mathbf{rc}}$. If we can simulate a table $\mathbf{T} \in \Sigma_{\mathbf{rc}}$ from a "trial distribution" $q(\cdot)$, where $q(\mathbf{T}) > 0$ for all $\mathbf{T} \in \Sigma_{\mathbf{rc}}$, then we have

$$E_q\left[\frac{1}{q(\mathbf{T})}\right] = \sum_{\mathbf{T}\in\Sigma_{\mathbf{rc}}} \frac{1}{q(\mathbf{T})} q(\mathbf{T}) = |\Sigma_{\mathbf{rc}}|.$$

Hence, we can estimate $|\Sigma_{\mathbf{rc}}|$ by

$$\widehat{|\Sigma_{\mathbf{rc}}|} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{q(\mathbf{T}_i)}$$

from N iid samples $\mathbf{T}_1, \ldots, \mathbf{T}_N$ drawn from $q(\mathbf{T})$. Furthermore, if we are interested in evaluating $\mu = E_p f(\mathbf{T})$, then we can use the weighted average,

$$\hat{\mu} = \frac{\sum_{i=1}^N f(\mathbf{T}_i)(p(\mathbf{T}_i)/q(\mathbf{T}_i))}{\sum_{i=1}^N (p(\mathbf{T}_i)/q(\mathbf{T}_i))} = \frac{\sum_{i=1}^N f(\mathbf{T}_i)(1/q(\mathbf{T}_i))}{\sum_{i=1}^N (1/q(\mathbf{T}_i))}, \quad (2)$$

as an estimate of μ. For example, if we let

$$f(\mathbf{T}) = \mathbb{1}_{\{\chi^2 \text{ statistic of } \mathbf{T} \geq s\}},$$

then (2) estimates the p value of the observed chi-square statistic s.

The standard error of $\hat{\mu}$ can be simply estimated by further repeated sampling. A more analytical method is to approximate the denominator of $\hat{\mu}$ by the δ-method so that

$$\operatorname{std}(\hat{\mu}) \approx \left(\operatorname{var}_q \left\{ f(\mathbf{T}) \frac{p(\mathbf{T})}{q(\mathbf{T})} \right\} + \mu^2 \operatorname{var}_q \left\{ \frac{p(\mathbf{T})}{q(\mathbf{T})} \right\} - 2\mu \operatorname{cov}_q \left\{ f(\mathbf{T}) \frac{p(\mathbf{T})}{q(\mathbf{T})}, \frac{p(\mathbf{T})}{q(\mathbf{T})} \right\} \right)^{1/2} \Big/ N^{1/2}.$$

However, because this standard deviation is dependent on the particular function of interest, it is also useful to consider a "function-free" criterion, the *effective sample size* (ESS) (Kong, Liu, and Wong 1994) to measure the overall efficiency of an importance sampling algorithm,

$$\text{ESS} = \frac{N}{1 + cv^2},$$

where the *coefficient of variation* (CV) is defined as

$$cv^2 = \frac{\operatorname{var}_q\{p(\mathbf{T})/q(\mathbf{T})\}}{E_q^2\{p(\mathbf{T})/q(\mathbf{T})\}},$$

which is equal to $\operatorname{var}_q\{1/q(\mathbf{T})\}/E_q^2\{1/q(\mathbf{T})\}$ for the current problem. The cv^2 is simply the chi-square distance between the two distributions p and q; the smaller it is, the closer the two distributions are. Heuristically, the ESS measures how many iid samples are equivalent to the N weighted samples. Throughout the article, we use cv^2 as a measure of efficiency for an importance sampling scheme. In practice, the theoretical value of the cv^2 is unknown, so its sample counterpart is used to estimate cv^2, that is,

$$cv^2 \approx \frac{\sum_{i=1}^N \{1/q(\mathbf{T}_i) - [\sum_{j=1}^N 1/q(\mathbf{T}_j)]/N\}^2/(N-1)}{\{[\sum_{j=1}^N 1/q(\mathbf{T}_j)]/N\}^2},$$

where $\mathbf{T}_1, \ldots, \mathbf{T}_N$ are N iid samples drawn from $q(\mathbf{T})$.

A central problem in importance sampling is constructing a good trial distribution $q(\cdot)$. Because the target space $\Sigma_{\mathbf{rc}}$ is rather complicated, it is not immediately clear what proposal distribution $q(\mathbf{T})$ can be used. Note that

$$q\big(\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_n)\big) = q(\mathbf{t}_1) q(\mathbf{t}_2|\mathbf{t}_1) q(\mathbf{t}_3|\mathbf{t}_2, \mathbf{t}_1) \cdots q(\mathbf{t}_n|\mathbf{t}_{n-1}, \ldots, \mathbf{t}_1), \quad (3)$$

where $\mathbf{t}_1, \ldots, \mathbf{t}_n$ denote the configurations of the columns of $\mathbf{T}$. This factorization suggests that it is perhaps a fruitful strategy to generate the table sequentially, column by column, and use the partially sampled table to guide the sampling of the next column. More precisely, the first column of the table is sampled conditional on its marginal sum c_1. Conditional on the realization of the first column, the row sums are updated, and the second column is sampled in a similar manner. This procedure is repeated until all of the columns are sampled. The recursive nature of (3) gives rise to the name *sequential importance sampling*. A general theoretical framework for SIS was given by Liu and Chen (1998). SIS is in fact just an importance sampling algorithm, but the design of the proposal distribution is adaptive in nature.

3. SAMPLING ZERO–ONE TABLES: THEORY AND IMPLEMENTATION

To avoid triviality, we assume throughout the article that none of the row or column sums is 0, none of the row sums is n, and none of the column sums is m. For the first column, we need to choose c_1 of the m possible positions to put 1's in. Suppose that the c_1 rows that we choose are $i_1, \ldots, i_{c_1}$. Then we need consider only the new $m \times (n-1)$ subtable. The row sums of the new table are updated by subtracting the respective numbers in the first column from the original row sums. Then the same procedure can be applied to sample the second column.

For convenience, we let $r_j^{(l)}, j = 1, \ldots, m$ denote the updated row sums after the first $l-1$ columns have been sampled. For example, $r_j^{(1)} = r_j$, and, after sampling the positions $i_1, \ldots, i_{c_1}$ for the first column, we have

$$r_j^{(2)} = \begin{cases} r_j^{(1)} - 1, & \text{if } j = i_k \text{ for some } 1 \leq k \leq c_1 \\ r_j^{(1)}, & \text{otherwise.} \end{cases} \quad (4)$$

Let $c_j^{(l)}, j = 1, \ldots, n-(l-1), l = 1, \ldots, n$, denote the updated column sums after we have sampled the first $l-1$ columns. That is, after sampling the first $l-1$ columns, we update the lth column in the original table to the first "current column" so that $(c_1^{(l)}, \ldots, c_{n-(l-1)}^{(l)}) = (c_l, \ldots, c_n)$.

A naive way to sample the c_1 nonzero positions for the first column (and, subsequently, the other columns) is from the uniform distribution, which can be rapidly executed. But this method turns out to be very inefficient; the cv^2 routinely exceeds 10,000, making the effective sample size very small. Although it is perhaps helpful to apply the resampling idea (Liu and Chen 1998), a more direct way to improve efficiency is to design a better sampling distribution. Intuitively, we want to put a "1" in position i if the row sum r_i is very large, and a "0" otherwise. To achieve this goal, we adopt here the conditional-Poisson (CP) sampling method described by Brewer and Hanif (1983) and Chen, Dempster, and Liu (1994).

3.1 Sampling From the Conditional Poisson Distribution

Let

$$\mathbf{Z} = (Z_1, \ldots, Z_m) \quad (5)$$

be independent Bernoulli trials with probability of successes $\mathbf{p} = (p_1, \ldots, p_m)$. Then the random variable

$$S_{\mathbf{Z}} = Z_1 + \cdots + Z_m$$

is said to follow the *Poisson-binomial* distribution. In the next section we provide some choices of the p_i for the zero–one table

simulation. The conditional distribution of $\mathbf{Z}$ given $S_{\mathbf{Z}}$ is called the CP distribution. If we let $w_i = p_i/(1-p_i)$, then it is easy to see that

$$P(Z_1 = z_1, \ldots, Z_m = z_m | S_{\mathbf{Z}} = c) \propto \prod_{k=1}^{m} w_k^{z_k}. \tag{6}$$

Chen et al. (1994) and Chen and Liu (1997) provided five schemes to sample from the CP distribution; we adopt their drafting sampling method here. Sampling from the CP distribution, as defined in (6), can be described through sampling c units without replacement from the set $\{1, \ldots, m\}$ with probability proportional to the product of each unit's "weight" w_i. Let A_k $(k = 0, \ldots, c)$ denote the set of selected units after k draws. Thus $A_0 = \emptyset$, and A_c is the final sample that we obtain. At the kth step of the drafting sampling $(k = 1, \ldots, c)$, a unit $j \in A_{k-1}^c$ is selected into the sample with probability

$$P(j, A_{k-1}^c) = \frac{w_j R(c-k, A_{k-1}^c \backslash j)}{(c-k+1) R(c-k+1, A_{k-1}^c)},$$

where

$$R(s, A) = \sum_{B \subset A, |B| = s} \left(\prod_{i \in B} w_i \right).$$

Most of the computing time required by this sampling procedure is spent on calculating $R(s, A)$ through the recursive formula $R(s, A) = R(s, A \setminus \{s\}) + w_s R(s-1, A \setminus \{s\})$, and the whole process is of order $O(s|A|)$. (See Chen et al. 1994; Chen and Liu 1997 for more details on CP sampling and its applications.)

3.2 Justification of the Conditional-Poisson Sampling

The following theorem, the proof of which is given in Appendix A, provides some insight on why the CP distribution is desirable in the sequential sampling of zero–one tables.

Theorem 1. For the uniform distribution over all $m \times n$ zero–one tables with given row sums $r_1, \ldots, r_m$ and first column sum c_1, the marginal distribution of the first column is the same as the conditional distribution of $\mathbf{Z}$ [defined by (6)] given $S_{\mathbf{Z}} = c_1$ with $p_i = r_i/n$.

Because the desired (true) marginal distribution for the first column $\mathbf{t}_1$ is $p(\mathbf{t}_1) = P(\mathbf{t}_1 | r_1, \ldots, r_m, c_1, \ldots, c_n)$, it is natural to let the sampling distribution of $\mathbf{t}_1$ be $q(\mathbf{t}_1) = P(\mathbf{t}_1 | r_1, \ldots, r_m, c_1)$, which is exactly the CP distribution with $p_i = r_i/n$. Suppose that we have sampled the first $l-1$ columns during the process; we then update the current number of columns left, $n-(l-1)$, and the current row sums $r_i^{(l)}$, and generate column l with the CP sampling method using the weights $r_i^{(l)}/[n-(l-1)-r_i^{(l)}]$. Because the CP distribution $q(\mathbf{t}_1)$ is not exactly the same as the true marginal distribution of $\mathbf{t}_1$, we may want to adjust the weights to make $q(\mathbf{t}_1)$ closer to $p(\mathbf{t}_1)$; see Section 7 for further discussion. In our experience, however, CP sampling without any adjustment of the weights already exhibited very good performance (see the examples in Sec. 6). During the sampling process, if any row sum equals 0 (or the number of rows left), then one can fill that row by 0 (or 1) and remove it from further consideration.

One requirement for an importance sampling algorithm to work is that the support of the proposal distribution $q(\cdot)$ must contain the support of the target distribution $p(\cdot)$. It is easy to see that for any zero–one table $\mathbf{T}$ that satisfies the row and column sum constraints, its first column $\mathbf{t}_1$ has a nonzero sampling probability $q(\mathbf{t}_1)$. The same argument applies recursively to $q(\mathbf{t}_2|\mathbf{t}_1)$, and so on, which shows that $q(\mathbf{T}) > 0$. In fact, the support of $q(\cdot)$ is larger than $\Sigma_{\mathbf{rc}}$ (see Sec. 4), and a more delicate SIS algorithm is provided in Section 4 guaranteeing that the support of the proposal distribution is the same as $\Sigma_{\mathbf{rc}}$.

The asymptotic analysis of Good and Crook (1977) provided another intuition for the use of CP sampling. In particular, they gave the following approximation to the number of zero–one matrices with fixed row sums $\mathbf{r} = (r_1, r_2, \ldots, r_m)$ and column sums $\mathbf{c} = (c_1, c_2, \ldots, c_n)$:

$$|\Sigma_{\mathbf{rc}}| \sim \Delta_{\mathbf{rc}} \equiv \frac{\prod_{i=1}^{m} \binom{n}{r_i} \prod_{j=1}^{n} \binom{m}{c_j}}{\binom{mn}{M}}, \tag{7}$$

where $M = \sum_{i=1}^{m} r_i = \sum_{j=1}^{n} c_j$. Let $\mathbf{v}(i_1, \ldots, i_{c_1})$ be the zero–one vector of length m that has i_kth component equal to 1 for $1 \le k \le c_1$ and all other components equal to 0. For a particular configuration of the first column, $\mathbf{t}_1 = \mathbf{v}(i_1, \ldots, i_{c_1})$, we let $\mathbf{r}^{(2)} = (r_1^{(2)}, \ldots, r_m^{(2)})$ and $\mathbf{c}^{(2)} = (c_2, \ldots, c_n)$ be the updated row and column sums as defined in (4). Then, by approximation (7), we have

$$p\big(\mathbf{t}_1 = \mathbf{v}\big(i_1, \ldots, i_{c_1}\big)\big) \approx \frac{\Delta_{\mathbf{r}^{(2)}\mathbf{c}^{(2)}}}{\Delta_{\mathbf{rc}}} \propto \prod_{k=1}^{c_1} \frac{r_{i_k}}{n - r_{i_k}}.$$

Thus this approximation also suggests that we should sample the first column according to the CP distribution with weights proportional to $r_i/(n-r_i)$.

Békéssy et al. (1972) gave another asymptotic result for $|\Sigma_{\mathbf{rc}}|$,

$$|\Sigma_{\mathbf{rc}}| \sim \Delta_{\mathbf{rc}}^* \equiv \frac{M!}{\prod_{i=1}^{m} r_i! \prod_{j=1}^{n} c_j!} e^{-\alpha(\mathbf{r}, \mathbf{c})}, \tag{8}$$

where

$$\begin{aligned} \alpha(\mathbf{r}, \mathbf{c}) &= 2 \frac{[\sum_{i=1}^{m} \binom{r_i}{2}][\sum_{j=1}^{n} \binom{c_j}{2}]}{(\sum_{i=1}^{m} r_i)(\sum_{j=1}^{n} c_j)} \\ &= \frac{1}{2M^2} \sum_{i=1}^{m} (r_i^2 - r_i) \sum_{j=1}^{n} (c_j^2 - c_j). \end{aligned}$$

This approximation has been proven to work well for large and sparse zero–one matrices. By (8), we have

$$p\big(\mathbf{t}_1 = \mathbf{v}\big(i_1, \ldots, i_{c_1}\big)\big) \approx \frac{\Delta_{\mathbf{r}^{(2)}\mathbf{c}^{(2)}}^*}{\Delta_{\mathbf{rc}}^*} \propto \prod_{k=1}^{c_1} r_{i_k} e^{-\alpha(\mathbf{r}^{(2)}, \mathbf{c}^{(2)})}.$$

We note that

$$\alpha\big(\mathbf{r}^{(2)}, \mathbf{c}^{(2)}\big) = \frac{\sum_{j=2}^{n} (c_j^2 - c_j)}{(M - c_1)^2} \sum_{i=1}^{m} \binom{r_i^{(2)}}{2}$$

and $\sum_{i=1}^{m} \binom{r_i^{(2)}}{2} = \sum_{i=1}^{m} (r_i^2 - r_i)/2 + c_1 - \sum_{k=1}^{c_1} r_{i_k}$. Hence,

$$\frac{\Delta_{\mathbf{r}^{(2)}\mathbf{c}^{(2)}}^*}{\Delta_{\mathbf{rc}}^*} \propto \prod_{k=1}^{c_1} r_{i_k} e^{d r_{i_k}},$$

where $d = \sum_{j=2}^{n} (c_j^2 - c_j)/(M - c_1)^2$. Thus another CP sampling distribution can be conducted with the weights proportional to $r_i e^{d r_i}$.

Although it is not clear whether the approximations (7) or (8) are accurate for a given table, we observed that these two CP-based SIS methods performed similarly well in all of the settings that we have tested and were extremely accurate when the marginal sums do not vary much. For the rest of the article, we focus on the CP sampling strategy based on approximation (7) (see Chen 2001 for further discussion of different sampling strategies for zero–one tables).

4. A MORE DELICATE SEQUENTIAL IMPORTANCE SAMPLING METHOD

Although the SIS procedure described in the previous section is already very effective, we found that sometimes the sampling cannot proceed after a few columns have been generated, because no valid zero–one table can be produced. For example, suppose that we want to sample tables with row sums 4, 4, 2, and 1 and column sums 3, 3, 3, 1, and 1. If we happen to draw the first column as $(1, 0, 1, 1)^T$ and the second column as $(1, 1, 1, 0)^T$, then we would have no way to sample the third column. In the following, we show that there exists an easy-to-check condition that guarantees the existence of subtables with the updated row and column sums. This condition helps us develop a more delicate SIS procedure for sampling more efficiently from $\Sigma_{\mathbf{rc}}$. Before we describe the procedure, we first provide some background (see Marshall and Olkin 1979 for more details).

Definition 1. For any $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{R}^n$, we let $x_{[1]} \geq \cdots \geq x_{[n]}$ denote the components of $\mathbf{x}$ in decreasing order. For $\mathbf{x}, \mathbf{y} \in \mathcal{R}^n$, we define $\mathbf{x} \prec \mathbf{y}$ if

$$\begin{cases} \displaystyle\sum_{i=1}^{k} x_{[i]} \leq \sum_{i=1}^{k} y_{[i]}, & k = 1, \ldots, n-1 \\ \displaystyle\sum_{i=1}^{n} x_{[i]} = \sum_{i=1}^{n} y_{[i]}. \end{cases} \tag{9}$$

When $\mathbf{x} \prec \mathbf{y}$, $\mathbf{x}$ is said to be *majorized* by $\mathbf{y}$ ($\mathbf{y}$ majorizes $\mathbf{x}$).

Lemma 1. Suppose that $(j_1, \ldots, j_n)$ is a permutation of $(1, \ldots, n)$. Then $\mathbf{x} \prec \mathbf{y}$ implies that

$$\begin{cases} \displaystyle\sum_{i=1}^{k} x_{j_i} \leq \sum_{i=1}^{k} y_{[i]}, & k = 1, \ldots, n-1 \\ \displaystyle\sum_{i=1}^{n} x_{j_i} = \sum_{i=1}^{n} y_{[i]}. \end{cases} \tag{10}$$

Proof. Because $x_{[1]} \geq \cdots \geq x_{[n]}$ are the components of $\mathbf{x}$ in decreasing order and $(j_1, \ldots, j_n)$ is a permutation of $(1, \ldots, n)$, we have

$$\begin{cases} \displaystyle\sum_{i=1}^{k} x_{j_i} \leq \sum_{i=1}^{k} x_{[i]}, & k = 1, \ldots, n-1 \\ \displaystyle\sum_{i=1}^{n} x_{j_i} = \sum_{i=1}^{n} x_{[i]}. \end{cases} \tag{11}$$

$\mathbf{x} \prec \mathbf{y}$ implies (9). The lemma follows immediately from (11) and (9).

Definition 2. Let $a_1, a_2, \ldots, a_n$ be nonnegative integers, and define

$$a_j^* = \#\{a_i : a_i \geq j\}, \qquad j = 1, 2, \ldots.$$

The sequence $a_1^*, a_2^*, a_3^*, \ldots$ is said to be conjugate to $a_1, a_2, \ldots, a_n$. Note that the conjugate sequence $\{a_i^*\}$ is always nonincreasing and is independent of the order of the a_i.

Lemma 2 (Gale 1957; Ryser 1957). Let $r_1, \ldots, r_m$ be nonnegative integers not exceeding n, and $c_1, \ldots, c_n$ be nonnegative integers not exceeding m. A necessary and sufficient condition for the existence of an $m \times n$ zero–one table with row sums $r_1, \ldots, r_m$ and column sums $c_1, \ldots, c_n$ is that

$$\mathbf{c} \equiv (c_1, \ldots, c_n) \prec (r_1^*, \ldots, r_n^*) \equiv \mathbf{r}^*,$$

or, equivalently, $\mathbf{r} \equiv (r_1, \ldots, r_m) \prec (c_1^*, \ldots, c_m^*) \equiv \mathbf{c}^*$.

Because the size of $\Sigma_{\mathbf{rc}}$ does not depend on the order of the row sums, we can arrange that $r_1 \geq \cdots \geq r_m$ without loss of generality. Let the conjugate of $(c_1^{(1)}, \ldots, c_n^{(1)}) = (c_1, \ldots, c_n)$ be $(c_1^{(1)*}, \ldots, c_n^{(1)*})$. The conjugate of $(c_1^{(2)}, \ldots, c_{n-1}^{(2)})$, denoted by $(c_1^{(2)*}, \ldots, c_{n-1}^{(2)*})$, is

$$c_j^{(2)*} = \begin{cases} c_j^{(1)*} - 1, & 1 \leq j \leq c_1 \\ c_j^{(1)*}, & j > c_1. \end{cases}$$

From Lemma 2, we know that a necessary and sufficient condition for the existence of an $m \times (n-1)$ zero–one table with row sums $r_1^{(2)}, \ldots, r_m^{(2)}$ and column sums $c_1^{(2)}, \ldots, c_{n-1}^{(2)}$ is that

$$\mathbf{r}^{(2)} \equiv \left(r_1^{(2)}, \ldots, r_m^{(2)}\right) \prec \left(c_1^{(2)*}, \ldots, c_m^{(2)*}\right) \equiv \mathbf{c}^{(2)*};$$

that is,

$$\begin{cases} \displaystyle\sum_{i=1}^{k} r_{[i]}^{(2)} \leq \sum_{i=1}^{k} c_i^{(2)*}, & k = 1, \ldots, m-1 \\ \displaystyle\sum_{i=1}^{m} r_{[i]}^{(2)} = \sum_{i=1}^{m} c_i^{(2)*}, \end{cases}$$

where $r_{[i]}$ denotes the components of $\mathbf{r}$ in decreasing order. From Lemma 1, we know that $\mathbf{r}^{(2)} \prec \mathbf{c}^{(2)*}$ implies that

$$\begin{cases} \displaystyle\sum_{i=1}^{k} r_i^{(2)} \leq \sum_{i=1}^{k} c_i^{(2)*}, & k = 1, \ldots, m-1 \\ \displaystyle\sum_{i=1}^{m} r_i^{(2)} = \sum_{i=1}^{m} c_i^{(2)*}. \end{cases} \tag{12}$$

Thus (12) is clearly a necessary condition for the existence of the subtable with new row sums and column sums. We prove in the following theorem that it is also a sufficient condition.

Theorem 2. Let $\mathbf{a}' = (a_1', \ldots, a_n')$, $\mathbf{b} = (b_1, \ldots, b_n)$. Suppose that $a_1' \geq \cdots \geq a_n'$ and $b_1 \geq \cdots \geq b_n$ are all nonnegative integers and there are $d \geq 1$ nonzero components in $\mathbf{b}$. Pick any d', $1 \leq d' \leq d$, nonzero components from $\mathbf{b}$, say $b_{k_1}, \ldots, b_{k_{d'}}$. Define $\mathbf{b}' = (b_1', \ldots, b_n')$ as

$$b_j' = \begin{cases} b_j - 1, & \text{if } j = k_i \text{ for some } 1 \leq i \leq d' \\ b_j & \text{otherwise} \end{cases}$$

and suppose that $\mathbf{b}'$ satisfies

$$\begin{cases} \sum_{i=1}^{k} b_i' \le \sum_{i=1}^{k} a_i', & k = 1, \ldots, n-1 \\ \sum_{i=1}^{n} b_i' = \sum_{i=1}^{n} a_i'. \end{cases} \tag{13}$$

Then $\mathbf{b}'$ is majorized by $\mathbf{a}'$.

The proof of Theorem 2 is given in Appendix B. The reason that this result is not entirely trivial is that $\mathbf{b}'$ is not necessarily ordered. For example, if $\mathbf{a}' = (4, 4, 2, 1)$, $\mathbf{b} = (4, 4, 3, 1)$, and $d' = 1$, then $\mathbf{b}'$ might be $(3, 4, 3, 1)$. To see that the theorem implies that condition (12) is necessary and sufficient, we let $\mathbf{a}' = \mathbf{c}^{(2)*}$, $\mathbf{b} = \mathbf{r}^{(1)}$ $(= \mathbf{r})$, and $\mathbf{b}' = \mathbf{r}^{(2)}$ and let condition (12) hold. Theorem 2 implies that $\mathbf{b}' \prec \mathbf{a}'$ or, equivalently, $\mathbf{r}^{(2)} \prec \mathbf{c}^{(2)*}$, which, according to Lemma 2, guarantees that there exists some zero–one subtable having the new row sums $\mathbf{r}^{(2)}$ and column sums $\mathbf{c}^{(2)}$.

Although we do not know $\mathbf{r}^{(2)}$ before we sample the first column, we can restate condition (12) from the current $\mathbf{r}^{(1)}$ and $\mathbf{c}^{(2)*}$. For each $1 \le k \le m$, we compare $\sum_{i=1}^{k} r_i$ and $\sum_{i=1}^{k} c_i^{(2)*}$:

- If $\sum_{i=1}^{k} r_i > \sum_{i=1}^{k} c_i^{(2)*}$, then we must put at least $\sum_{i=1}^{k} r_i - \sum_{i=1}^{k} c_i^{(2)*}$ 1's at or before the kth row in the first column. For convenience, we may call k a knot.
- If $\sum_{i=1}^{k} r_i \le \sum_{i=1}^{k} c_i^{(2)*}$, then there is no restriction at the kth row.

These two conditions can be summarized by two vectors, one vector recording the positions of the knots, denoted by $(k[1], k[2], \ldots)$, and the other vector recording how many 1's we must put at or before those knots, denoted by $(v[1], v[2], \ldots)$. To make the conditions easier to implement, we eliminate some redundant knots:

a. If $v[j] \le v[i]$ for some $j > i$, then we ignore knot $k[j]$.

b. If $v[j] - v[i] \ge k[j] - k[i]$ for some $j > i$, then we ignore knot $k[i]$. If the restriction on knot $k[j]$ is satisfied, then it will guarantee that the restriction on knot $k[i]$ is also satisfied.

Using the foregoing conditions, we design the following, more delicate, CP sampling strategy.

- We are required to place at least $v[1]$ but no more than $\min(k[1], c_1)$ 1's at or before row $k[1]$. So we assign equal probability to these choices, that is,

$$q_1\{(\text{number of 1's at or before row } k[1]) = i\} = \frac{1}{\min(k[1], c_1) - v[1] + 1}$$

  for $v[1] \le i \le \min(k[1], c_1)$.
- After the number of 1's o_1 at or before row $k[1]$ is chosen according to the foregoing distribution, we pick the o_1 positions between row 1 and row $k[1]$ using the CP sampling with weights $r_i/(n - r_i)$. (See Sec. 7 for other choices of weights. Sampling uniformly instead of using the CP distribution for this step can reduce the algorithm's efficiency by several orders of magnitude.)
- After o_1 positions have been chosen for knot 1, we consider knot 2 conditional on the 1's that we have already placed at or before knot 1. Because we are required to place at least $v[2]$ 1's at or before row $k[2]$, the number of 1's o_2 that we could put between knot 1 and knot 2 ranges from $\max(v[2] - o_1, 0)$ to $\min(k[2] - k[1], c_1 - o_1)$. We assign equal probability to all of these choices for o_2. Then we pick the o_2 positions between row $k[1]$ and $k[2]$ once again using CP sampling.
- We continue the procedure until all of the knots in column 1 have been considered.
- After we have completed the first column, we record the probability $q(\mathbf{t}_1)$ of getting such a sample for the first column, update the row sums, rearrange the updated row sums in decreasing order, and repeat the procedure with the second column.

The foregoing, more delicate CP sampling strategy improves on the basic CP sampling by checking the existence of subtables with the updated row and column sums when sampling each column. The support of the basic CP sampling strategy proposed in Section 3.2 is larger than $\Sigma_{\mathbf{rc}}$ (the set of all $m \times n$ zero–one tables with row sums $\mathbf{r}$ and column sums $\mathbf{c}$). Lemma 2 and Theorem 2 guarantee that we can always have a valid table in $\Sigma_{\mathbf{rc}}$ by the more delicate CP sampling. Therefore, the support for the more delicate CP sampling strategy is the same as $\Sigma_{\mathbf{rc}}$. This allows us to sample more efficiently from $\Sigma_{\mathbf{rc}}$, without generating any invalid tables. The reader may want to look ahead to Section 6 for examples.

5. SAMPLING CONTINGENCY TABLES

Sampling from contingency tables is much easier to implement than sampling from zero–one tables, because there are fewer restrictions on the values that each entry can take. For a contingency table, given positive row sums $r_1, \ldots, r_m$ and column sums $c_1, \ldots, c_n$, the necessary and sufficient condition for the existence of a contingency table with such row and column sums is

$$r_1 + r_2 + \cdots + r_m = c_1 + c_2 + \cdots + c_n \equiv M.$$

This is much simpler than the Gale–Ryser condition, which makes the whole procedure much simpler to implement.

We still sample column by column as we did for zero–one tables. Suppose that the element at the ith row and the jth column is a_{ij}. We start from the first column. We have that a_{11} must satisfy

$$0 \le a_{11} \le r_1,$$
$$c_1 - \sum_{i=2}^{m} r_i = c_1 + r_1 - M \le a_{11} \le c_1.$$

So, combining the two equations, we have

$$\max(0, c_1 + r_1 - M) \le a_{11} \le \min(r_1, c_1).$$

It is also easy to see that this is the only condition that a_{11} needs to satisfy. Recursively, suppose that we have chosen $a_{i1} = a_{i1}'$

for $1 \le i \le k-1$. Then the only restriction on a_{k1} is

$$\max\left(0, \left(c_1 - \sum_{i=1}^{k-1} a'_{i1}\right) - \sum_{i=k+1}^{m} r_i\right) \le a_{k1} \le \min\left(r_k, c_1 - \sum_{i=1}^{k-1} a'_{i1}\right).$$

Thus we need to consider only the strategy for sampling a_{11}, and can apply the same strategy recursively to sample other cells.

If we collapse columns 2 to m and rows 2 to n to form a 2×2 table with a_{11} as the only variable, then the uniform distribution on all tables implies that a_{11} is uniform in its range $[\max(0, c_1 + r_1 - M), \min(r_1, c_1)]$. However, if we consider both a_{11} and a_{21} simultaneously (i.e., the original table is collapsed into a 3×2 table), then for each $a_{11} = x$, the choices of a_{21} range from $\max(0, c_1 + r_1 + r_2 - M - x)$ to $\min(r_2, c_1 - x)$. Thus, if our goal is to sample a 3×2 table uniformly, then we should have

$$P(a_{11} = x) \propto \min(r_2, c_1 - x) - \max(0, c_1 + r_1 + r_2 - M - x) + 1.$$

An analog of conditional Poisson sampling could be developed. Our examples in Section 6 show, however, that the simple uniform sampling of a_{11} seems to have already worked very well.

6. APPLICATIONS AND SIMULATIONS

In the examples in this section, we generated zero–one tables by the more delicate CP sampling with weights proportional to $r_i^{(l)}/[n - (l-1) - r_i^{(l)}]$ (see Sec. 4), which we call the CP sampling for abbreviation. Contingency tables are generated by the SIS algorithm proposed in Section 5. All examples were coded in C language and run on an Athlon workstation with a 1.2-GHz processor.

6.1 Counting Zero–One Tables

Here we apply the SIS procedure described in Section 4 to estimate the number of zero–one tables with given row sums $r_1, r_2, \ldots, r_m$ and column sums $c_1, c_2, \ldots, c_n$. Because the ordering of the column or row sums does not affect the total number of tables, in the following examples we attempt to arrange the rows and columns in such a way that the cv^2 is made small. We discuss some heuristic rules for arranging the rows and columns to achieve a low cv^2 in Section 7.

We first tested our method on counting the number of 12×12 zero–one tables with all marginal sums equal to 2, which is a subset of the 12×12 "magic squares." For CP sampling, the cv^2 of the weights was .04. It took about 1 second to obtain 10,000 tables and their weights using the delicate SIS procedure described in Section 4. The average of the weights gives rise to an estimate of $(2.196 \pm .004) \times 10^{16}$, where the number after the "±" sign is the standard error. For this table, the exact answer of 21,959,547,410,077,200 was given by Wang and Zhang (1998). Although Wang and Zhang's formula provides a fast answer to this problem, it is often difficult to quickly compute their formula for larger zero–one tables.

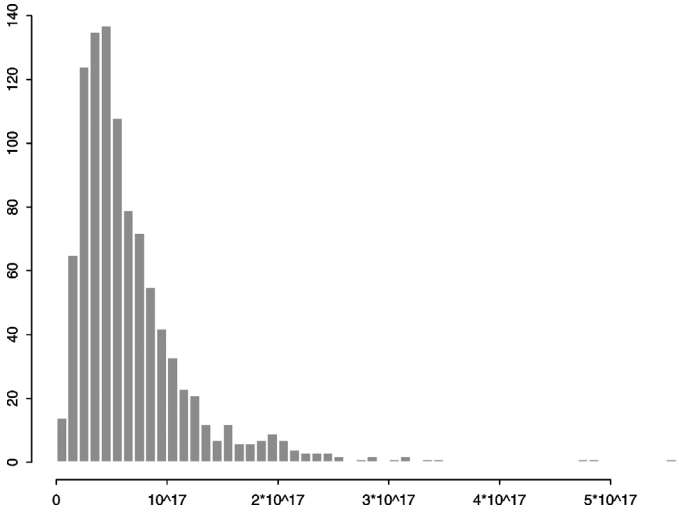

Figure 1. Histogram of 1,000 Importance Weights.

Counting the number of tables with the same marginal sums as the finch data (see Table 1) is a more challenging exercise. The last row of the original table is removed, because it consists of all 1's and will not affect the counting. We ordered the 17 column sums from the largest to the smallest and applied the CP sampling, which gives a cv^2 of around 1. With 10,000 samples, which took about 10 seconds, we estimated the total number of zero–one tables to be $(6.72 \pm .07) \times 10^{16}$. As a verification, we obtained a more accurate estimate of 6.7150×10^{16} based on 10^8 samples. Here the exact answer was computed for us by David desJardins using a clever divide-and-conquer algorithm. His program (confirmed by an independent check) gives exactly 67,149,106,137,567,626 tables. We see that the SIS algorithm gives a very accurate approximation. Figure 1, a histogram of 1,000 importance weights, shows that the weights are tightly distributed in a relatively small range. The ratio of the maximum weight over the median weight is about 10.

To further challenge our method, we randomly generated a 50×50 table for which the probability for each cell to be 1 is .2. The row sums of the table are

10, 8, 11, 11, 13, 11, 10, 9, 7, 9, 10, 16, 11, 9,
12, 14, 12, 7, 9, 10, 10, 6, 11, 8, 9, 8, 14, 12,
5, 10, 10, 8, 7, 8, 10, 10, 14, 6, 10, 7, 13, 4, 6,
8, 9, 15, 11, 12, 10, 6,

and the column sums are

9, 6, 12, 11, 9, 8, 8, 11, 9, 11, 13, 7, 10, 8, 9, 7,
8, 3, 10, 11, 13, 7, 5, 11, 10, 9, 10, 13, 9, 9, 7, 7,
6, 8, 10, 12, 8, 12, 16, 12, 15, 12, 13, 13, 10, 7,
12, 13, 6, 11.

We ordered the column sums from largest to smallest and used CP sampling, which gave a cv^2 of around .03. Based on 100 samples, which took a few minutes to generate, we estimated that the total number of zero–one tables with these marginal sums was $(7.7 \pm .1) \times 10^{432}$.

Because our method generally works well when the marginal sums do not vary much, we tried another example for which the marginal sums were forced to vary considerably. We generated a 50×50 table with cell (i, j) being 1 with probability $\exp(-6.3(i+j-2)/(m+n-2))$, which gave rise to the row sums

14, 14, 19, 18, 11, 12, 12, 10, 13, 16, 8, 12,
6, 15, 6, 7, 12, 1, 12, 3, 8, 5, 9, 4, 2, 4, 1, 4,
4, 5, 2, 3, 3, 1, 1, 1, 2, 1, 1, 2, 1, 3, 3, 1, 3,
2, 1, 1, 1, 2

and the column sums

14, 13, 14, 13, 13, 12, 14, 8, 11, 9, 10, 8,
9, 8, 4, 7, 10, 9, 6, 7, 6, 5, 6, 8, 1, 6, 6, 3,
2, 3, 5, 4, 5, 2, 2, 2, 3, 2, 4, 3, 1, 1, 1, 3, 2,
2, 3, 5, 2, 5.

With the same SIS method as in the previous case, we had a cv^2 of .2. Based on 1,000 samples, we estimated the total number of zero–one tables with these margins as $(8.9 \pm .1) \times 10^{242}$. Based on 10,000 samples, the estimate was improved to $(8.78 \pm .05) \times 10^{242}$. Finally, we estimated the total number of 100×100 zero–one tables with all marginal sums equal to 2 to be $(2.96 \pm .03) \times 10^{314}$ based on 100 Monte Carlo samples. The cv^2 in this case was .008, showing again that the SIS approach is extremely efficient. In comparison, we know of no MCMC-based algorithm that can achieve a comparable accuracy for counting tables of these sizes with a reasonable amount of computing time.

6.2 Testing Zero–One Tables in Ecology

For the finch data, the observed test statistic (1) as suggested by Roberts and Stone (1990) is 53.1. We applied the CP-based SIS to approximate the p value of this statistic. Our algorithm took about 10 seconds to generate 10,000 tables, based on which we estimated the p value as $(4 \pm 2.8) \times 10^{-4}$. A longer simulation of 1,000,000 SIS samples gave an estimate of $(3.96 \pm .36) \times 10^{-4}$, which took about 18 minutes. Thus there is strong evidence against the null hypothesis of a uniform distribution conditional on the marginal sum. The null distribution of the test statistic in the form of a histogram (computed using the weighted samples) is given in Figure 2.

The following MCMC algorithm has been proposed to simulate from the uniform distribution of the tables and to estimate the p value (Besag and Clifford 1989; Cobb and Chen 2003). Pick two rows and two columns at random; if the intersection of the two rows and two columns are one of the following two types,

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \quad \text{or} \quad \begin{matrix} 0 & 1 \\ 1 & 0, \end{matrix}$$

then switch to the other type; otherwise, stay at the original table. Within 18 minutes, this MCMC scheme generated 15,000,000 samples, giving an estimated p value of $(3.56 \pm .68) \times 10^{-4}$. Thus the SIS algorithm is about four times more efficient than the MCMC algorithm (to achieve the same standard error) on this example. We note that for more complex functionals, say $E(\bar{S}^2)$, the relative efficiency of SIS can be even higher.

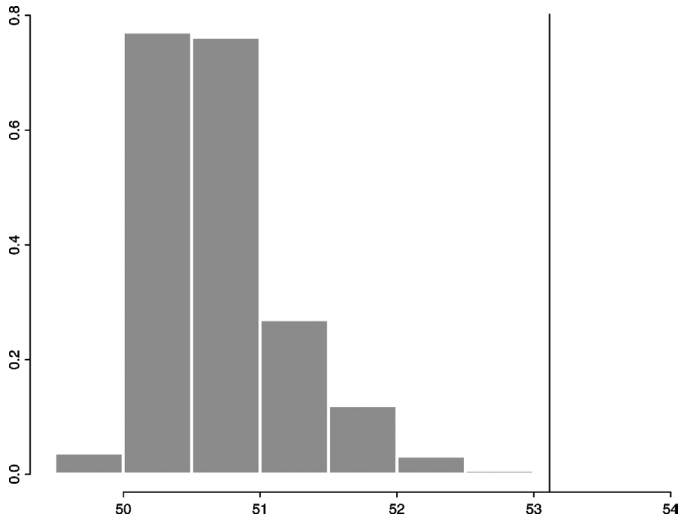

Figure 2. Approximated Null Distribution of the Test Statistic Based on 10,000 Weighted Samples. The vertical line indicates the observed $\bar{S}^2(T_0)$.

Sanderson (2000) described a method for generating zero–one tables with fixed margins, which he applied to the finch data with an implied belief that the tables obtained are uniformly distributed. His method does not produce uniformly distributed tables, however. For example, for the set of 3×3 tables with marginal sums (2, 2, 1) for both the columns and the rows, we found that the probability for Sanderson's method of generating one of the five possible configurations is 332/1,512, but that of generating each of the remaining configurations is 295/1,512. Because Sanderson's sampling method does not generate tables uniformly, the conclusion of his statistical hypothesis testing is questionable.

Many other occurrence matrices describing the distribution of birds, reptiles, and mammals on oceanic archipelagoes or mountain ranges have been collected (see, e.g., Cook and Quinn 1995). To compare the SIS method with the MCMC algorithm, we analyzed another dataset of the distribution of 23 land birds on the 15 southern islands in the Gulf of California (Cody 1983). The occurrence matrix has row sums 14, 14, 14, 12, 5, 13, 9, 11, 11, 11, 11, 11, 7, 8, 8, 7, 2, 4, 2, 3, 2, 2, and 2, and column sums 21, 19, 18, 19, 14, 15, 12, 15, 12, 12, 12, 5, 4, 4, and 1. Based on 100,000 sampled tables using the SIS algorithm, which took about 5 minutes, we estimated that the p value for the test statistic (1) is $.053 \pm .003$. The MCMC algorithm took about 6 minutes to generate 2,000,000 samples and estimated a p value is $.052 \pm .006$ (using 500,000 samples as burn-in). This shows that the SIS algorithm is more than four times faster than the MCMC algorithm for this example. A long simulation of 1,000,000 samples based on the SIS method gave an estimate of .0527 for the p value.

To compare the SIS and MCMC methods on a larger table, we randomly generated a 100×10 table for which the probability for each cell to be 1 is .3. The row sums of the table are 27, 33, 31, 25, 35, 32, 31, 29, 21, and 26, and the column sums are 2, 2, 3, 4, 4, 4, 3, 3, 4, 3, 4, 1, 0, 3, 3, 5, 5, 5, 1, 6, 3, 1,

1, 5, 3, 2, 4, 1, 2, 1, 3, 2, 3, 3, 0, 3, 4, 5, 1, 4, 3, 2, 1, 1, 1, 6, 3, 2, 4, 0, 2, 3, 4, 2, 2, 5, 1, 3, 2, 2, 3, 3, 3, 5, 4, 3, 5, 4, 5, 4, 4, 2, 6, 6, 5, 2, 3, 2, 0, 3, 4, 3, 5, 4, 2, 3, 1, 3, 3, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, and 3. Based on 10,000 sampled tables using the SIS method, which took about 35 seconds, we estimated that the p value is $.8894 \pm .0027$. The MCMC algorithm took about 35 seconds to generate 200,000 samples and estimated a p value of $.8913 \pm .0298$ (using 50,000 samples as burn-in). This shows that the SIS algorithm is more than 100 times faster than the MCMC algorithm on this example. A long simulation of 1,000,000 samples based on the SIS method gave an estimated p value of .8868.

We note that the foregoing comparison between SIS and MCMC focuses only on their efficiency in approximating p values (i.e., the expectation of a step function). The results may differ if the expectation of another function is considered. For example, the SIS method estimates $E(\bar{S}^2)$ even more efficiently than MCMC.

6.3 Testing the Rasch Model

Rasch (1960) proposed a simple linear logistic model to measure a person's ability based on his or her answers to a dichotomous response test. Suppose that n persons are asked to answer m questions (items). We can construct a zero–one matrix based on all of the answers. A 1 in cell (i, j) means that the ith person answered the jth question correctly, and a 0 means otherwise. The Rasch model assumes that each person's ability is characterized by a parameter θ_i, each item's difficulty is characterized by a parameter β_j, and

$$P(x_{ij} = 1) = \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}, \tag{14}$$

where x_{ij} is the ith person's answer to the jth question. The responses x_{ij} are assumed to be independent. The numbers of items answered correctly by each person (the column sums) are minimal sufficient statistics for the ability parameters, and the numbers of people answering each item correctly (the row sums) are minimal sufficient statistics for the item difficulty parameters.

The Rasch model has numerous attractive features and is widely used for constructing and scoring educational and psychological tests (Fischer and Molenaar 1995). There is a considerable literature on testing the goodness of fit of the Rasch model (see Glas and Verhelst 1995 for an overview). The validity of most of the proposed tests relies on asymptotic theory, a reliance that Rasch did not feel very comfortable with (Andersen 1995). In his seminal book, Rasch (1960) proposed a parameter-free "exact" test based on the conditional distribution of the zero–one matrix of responses with the observed marginal sums fixed. It is easy to see that under model (14), all of the zero–one tables are uniformly distributed conditional on the row and column sums. Because of the difficulty of accurately approximating the distribution of test statistics under this uniform distribution, Rasch never implemented his approach. Besag and Clifford (1989) and Ponocny (2001) have studied using Monte Carlo methods to test the Rasch model. The conceptually simpler and more efficient SIS strategy developed in this article is also ideally suited for implementing Rasch's ideas. For example, Chen and Small (2004) showed that in testing for item bias (Kelderman 1989), the uniformly most powerful (UMP) unbiased test resulting from Rasch's idea (Ponocny 2001) is both "exact" and highly powerful. In a simulation study with 100 samples, it was shown that the SIS-based UMP unbiased test had a power of .90 at the .05 significance level, whereas the popular Mantel–Haenszel test proposed by Holland and Thayer (1988) had only power .41. Chen and Small (2004) also reported that the SIS approach is more efficient and accurate than the Monte Carlo methods developed by Ponocny (2001) and Snijders (1991).

Here we study an example of the test of item bias for which the exact p value is known. A similar version of the following example was considered by Ponocny (2001) and Chen and Small (2004). The zero–one matrix of responses is 100×6 with all person scores (i.e., row sums) equal to 3 and all item totals (i.e., column sums) equal to 50. There are two subgroups, the first one consisting of the first 50 students. Consider testing the alternative hypothesis that item 1 is biased with the test statistic

$$f(\mathbf{T}) = \#\{\text{students in first subgroup who answer item 1 correctly}\}.$$

We want to calculate the p value for a table $\mathbf{T}_0$ with $f(\mathbf{T}_0) = 30$, that is, to calculate $P(f(\mathbf{T}) \geq 30 | \mathbf{r}, \mathbf{c})$ under the Rasch model. Because of the symmetry of the marginals and the fact that all marginals are given, the number of correct answers on item 1 among the first 50 students is hypergeometrically distributed; in particular, $P(f(\mathbf{T}) \geq 30 | \mathbf{r}, \mathbf{c}) = .03567$ under the Rasch model.

Based on 1,000 sampled tables, which took about 2 seconds for the CP sampling method, we estimated a p value of $.0353 \pm .0048$. The MCMC algorithm took about 2 seconds to generate 1,000,000 samples and estimated a p value of $.0348 \pm .0051$ (using 300,000 samples as burn-in). The MCMC and SIS algorithms gave similar performance on this example. Notice that the test statistic is the number of 1's among the first 50 entries of the first column. The Markov chain may have less autocorrelation for this simple test statistic.

6.4 Contingency Tables

To illustrate the SIS method described in Section 5 for counting the number of contingency tables, we consider the two examples discussed by Diaconis and Gangolli (1995). The first example is a 5×3 table with row sums 10, 62, 13, 11, and 39 and column sums 65, 25, and 45. We observed that the smallest cv^2 (1.07) was achieved when the column sums are arranged from the largest to the smallest and row sums are arranged from the smallest to the largest. We obtained 100,000 Monte Carlo samples, which took less than 1 second and provided us with the estimate of $(2.393 \pm .007) \times 10^8$. The true value of 239,382,173 was given by Diaconis and Gangolli (1995).

Besides counting the number of tables, the SIS method is also useful for carrying out certain hypothesis tests for contingency tables. The conditional volume test proposed by Diaconis and Efron (1985) addresses the question of whether the Pearson chi-square statistic of a contingency table is "atypical" when the observed table is regarded as a draw from the uniform distribution over tables with the given marginal sums. The observed chi-square statistic for the 5×3 table described earlier is 72.1821. With 1,000,000 Monte Carlo samples produced by

our SIS method, which took about 2 seconds, we estimated a p value for the conditional volume test of $.7610 \pm .0005$. A random-walk-based MCMC algorithm was discussed by Diaconis and Gangolli (1995) and can be described as follows. Pick two rows and two columns uniformly at random, then add or subtract 1 in the four entries at the intersection of the two rows and two columns according to the following pattern:

$$\begin{matrix} + & - \\ - & + \end{matrix} \quad \text{or} \quad \begin{matrix} - & + \\ + & - \end{matrix}.$$

The two patterns are chosen with equal probability. The random walk stays at the original table if the operation generates a negative table entry. This MCMC algorithm generated 800,000 samples in 2 seconds and estimated a p value of $.77 \pm .02$ (with 100,000 samples as burn-in). Thus the CP sampling is 1,600 times faster than the MCMC algorithm for this example. Diaconis and Gangolli (1995) gave the true value as .76086 based on a 12-hour exhaustive enumeration.

The second example is a 4×4 table with row sums 220, 215, 93, and 64 and column sums 108, 286, 71, and 127. Ordering the row sums from largest to smallest and the column sums from smallest to largest works best, and yielded a cv^2 around 3.7. The estimate (for the total number of tables) based on 1,000,000 samples, which took 2 seconds, was $(1.225 \pm .002) \times 10^{15}$. The true value of 1,225,914,276,768,514 was given by Diaconis and Gangolli (1995). Diaconis and Efron (1985) gave a formula for approximately counting the number of tables with given row and column sums that estimates 1.235×10^{16} tables for this example. Holmes and Jones (1996) estimated 1.226×10^{16} tables by the rejection method. We also performed the conditional volume test for this example. Based on the 1,000,000 Monte Carlo samples that we generated for estimating the total number of tables, we estimated the p value to be $.1532 \pm .0008$. In contrast, the MCMC algorithm took the same amount of time (2 seconds) to generate 800,000 samples and estimated a p value of $.166 \pm .003$ (with 100,000 samples as burn-in). Therefore, the SIS approach is about 14 times faster than the MCMC method for this problem.

Holmes and Jones (1996) gave another example, with five row sums, 9, 49, 182, 478, and 551, and four column sums, 9, 309, 355, and 596, and showed that the approximation formula of Diaconis and Efron (1985) does not work well. A distinctive feature of their example is that both the row and column sums have very small values. We tried SIS on this example, using the original order of the rows and ordering the column sums in decreasing order. The cv^2 was around 7, so that the effective sample size was about $N/(1+7) = 12.5\% \times N$. Holmes and Jones' first algorithm has an acceptance rate of 9.7%, and the revised one has an acceptance rate of 12.5%. In terms of effective sample size, our algorithm is as efficient as their revised algorithm. However, the SIS approach is simpler to implement and easier to understand than the revised algorithm of Holmes and Jones, which requires calculating the coefficients of a product of some very large polynomials.

For the Holmes and Jones example, we estimated the total number of tables to be $(3.384 \pm .009) \times 10^{16}$ based on 1,000,000 SIS samples. This took about 1 second to produce. Several estimates based on 10^8 samples were all around 3.383×10^{16}. In contrast, the estimates given by Holmes and Jones (1996) are 3.346×10^{16} and 3.365×10^{16}, which we believe underestimate the true number of tables.

7. DISCUSSION

In this article, we have developed a set of sequential importance sampling strategies for computing with zero–one or contingency tables. Our results show that these approaches are both very efficient and simple to implement. Two distinctive features of our approach to sampling zero–one tables are (a) it guarantees that sequential procedure always produces a valid table, thus avoiding wasting computational resources, and (b) it uses the CP sampling as the trial distribution to greatly increase its efficiency.

For CP sampling, we used weights proportional to $r_i^{(l)}/[n - (l-1) - r_i^{(l)}]$. Because the CP distribution $q(\mathbf{t}_1)$ is not exactly the same as the target distribution $p(\mathbf{t}_1)$ (the marginal distribution of the first column), we may want to adjust the weights to make $q(\mathbf{t}_1)$ closer to $p(\mathbf{t}_1)$. One easy adjustment is to use the set of weights $\{r_i^{(l)}/[n - (l-1) - r_i^{(l)}]\}^u$, where $u > 0$ can be chosen by the user. One may use a small sample size to estimate the cv^2 and choose the u that gives the lowest cv^2. This should not take more than a few seconds. For all of the zero–one tables that we have tested, the choice of $u = 1$ has worked very well. Although some small variation of u (e.g., ranging from .8 to 1.2) improved the efficiency of SIS a little, we did not observe any dramatic effect for the examples that we considered.

We used several different orderings of row sums and column sums. Our experience is that for zero–one tables, it is best to order the column sums from largest to smallest. This makes intuitive sense, because when we start with columns with many 1's, we do not have many choices, and $q(\mathbf{t}_l|\mathbf{t}_{l-1}, \ldots, \mathbf{t}_1)$ must be close to $p(\mathbf{t}_l|\mathbf{t}_{l-1}, \ldots, \mathbf{t}_1)$. After such columns have been sampled, the updated row sums will be greatly reduced, which will cause $q(\mathbf{t}_l|\mathbf{t}_{l-1}, \ldots, \mathbf{t}_1)$ to be closer to $p(\mathbf{t}_l|\mathbf{t}_{l-1}, \ldots, \mathbf{t}_1)$. Because of the way in which we do the sampling, we need to order the row sums from largest to smallest. Another option is to sample rows instead of columns. Our experience is that if the number of rows is greater than the number of columns, then sampling rows gives better results.

For contingency tables, we found that listing the column sums in decreasing order and listing the row sums in increasing order works best. The intuition is similar to that for zero–one tables. A surprising fact for contingency tables is that, given a certain ordering of the row and column sums, sampling the columns is the same as sampling the rows. It is not difficult to check this fact by carrying out our sampling method. Thus we do not need to worry about whether exchanging the roles of rows and columns provides better performance.

Because the tables produced by the SIS approach described here have a distribution very close to the target one (as evidenced by the low cv^2 values), the SIS method is markedly better than the available MCMC approach, which typically has a very long autocorrelation time, especially for large tables. This advantage of the SIS is reflected not only by a more accurate Monte Carlo approximation, but also by a more reliable estimate of the standard error of this approximation. To achieve the same accuracy, SIS usually needs fewer tables compared with the MCMC method; therefore, SIS becomes even more attractive when evaluating the test statistic itself is time-consuming. Furthermore, estimating the normalizing constant of the target distribution is a rather straightforward step for

the SIS method, but is much more difficult for MCMC strategies. For the table-counting problem, to our knowledge there are no MCMC algorithms that can achieve an accuracy even close to that of the SIS approach.

APPENDIX A: PROOF OF THEOREM 1

We start by giving an algorithm for generating tables uniformly from all $m \times n$ zero–one tables with given row sums $r_1, \ldots, r_m$ and first column sum c_1.

Algorithm.

1. For $i = 1, \ldots, m$, randomly choose r_i positions from the ith row and put 1's in. The choices of positions are independent across different rows.
2. Accept those tables with given first column sum c_1.

It is easy to see that tables generated by this algorithm are uniformly distributed over all $m \times n$ zero–one tables with given row sums $r_1, \ldots, r_m$ and first column sum c_1. We can derive the marginal distribution of the first column based on this algorithm. At step 1, we choose the first cell at the ith row [i.e., the cell at position $(i, 1)$] to put 1 in with probability

$$\binom{n-1}{r_1-1} \Big/ \binom{n}{r_1} = r_1/n.$$

Because the choices of positions are independent across different rows, after step 1 the marginal distribution of the first column is the same as the distribution of $\mathbf{Z}$ [defined by (5)] with $p_i = r_i/n$. Step 2 rejects the tables whose first column sum is not c_1. This implies that after step 2, the marginal distribution of the first column is the same as the conditional distribution of $\mathbf{Z}$ [defined by (6)] given $S_{\mathbf{Z}} = c_1$ with $p_i = r_i/n$.

APPENDIX B: PROOF OF THEOREM 2

Suppose that there are l distinct values among $\{b_1, \ldots, b_n\}$ and assume that $i_1 < \cdots < i_l$ are the jump points, that is,

$$b_{i_{k-1}+1} = \cdots = b_{i_k} > b_{i_k+1}, \qquad k = 1, \ldots, l-1,$$
$$b_{i_{l-1}+1} = \cdots = b_{i_l},$$

where $i_0 = 0$, $i_l = n$. Because b'_i equals b_i or $b_i - 1$ and $\mathbf{b}$ is ordered from largest to smallest, it is clear that if we have $i_{k-1} < i \le i_k$ and $i_k < j \le i_{k+1}$ for any i, j, k, then $b'_i \ge b'_j$. But within each block from b'_{i_k} to $b'_{i_{k+1}}$, some of the b'_i's are equal to b_{i_k} and the others are equal to $b_{i_k} - 1$. In other words, the b'_i's may not be ordered from largest to smallest in each block. If there is a j such that $i_{k-1} < j < i_k$ and

$$b'_j = b_{i_k} - 1, \qquad b'_{j+1} = b_{i_k},$$

then we show in the following that we can switch b'_j and b'_{j+1} and still maintain property (13). There are two different cases to consider:

Case (a): $\sum_{i=1}^{j} b'_i < \sum_{i=1}^{j} a'_i$. In this case, of course property (13) still holds after we switch b'_j and b'_{j+1} and obtain

$$b'_j = b_{i_k}, \qquad b'_{j+1} = b_{i_k} - 1.$$

Case (b): $\sum_{i=1}^{j} b'_i = \sum_{i=1}^{j} a'_i$, which we will show can never happen. Because $\sum_{i=1}^{j+1} b'_i \le \sum_{i=1}^{j+1} a'_i$, we have $a'_{j+1} \ge b'_{j+1} = b_{i_k}$. But because the a'_i are monotone nonincreasing, we have

$$a'_{i_{k-1}+1} \ge \cdots \ge a'_j \ge a'_{j+1} \ge b_{i_k}.$$

Because $b'_i \le b_{i_k}$ for $i_{k-1} + 1 \le i < j$, and $b'_j = b_{i_k} - 1$, we must have

$$\sum_{i=i_{k-1}+1}^{j} b'_i < \sum_{i=i_{k-1}+1}^{j} a'_i. \tag{B.1}$$

Combining (B.1) with the fact that $\sum_{i=1}^{i_{k-1}} b'_i \le \sum_{i=1}^{i_{k-1}} a'_i$, we finally have

$$\sum_{i=1}^{j} b'_i < \sum_{i=1}^{j} a'_i,$$

which contradicts the assumption that $\sum_{i=1}^{j} b'_i = \sum_{i=1}^{j} a'_i$.

The preceding arguments imply that we can always switch b'_j and b'_{j+1} and maintain (13) if

$$b'_j = b_{i_k} - 1, \qquad b'_{j+1} = b_{i_k}.$$

After a finite number of such switches, all of the b_i's in this block must be ordered from largest to smallest, $b'_{i_{k-1}+1} \ge \cdots \ge b'_{i_k}$, which easily leads to the conclusion that $\mathbf{b}' \prec \mathbf{a}'$.

[Received September 2003. Revised June 2004.]

REFERENCES

Andersen, E. B. (1995), "What Georg Rasch Would Have Thought About This Book," in *Rasch Models: Their Foundations, Recent Developments and Applications*, eds. G. H. Fischer and I. W. Molenaar, New York: Springer-Verlag, pp. 383–390.

Balmer, D. (1988), "Recursive Enumeration of $r \times c$ Tables for Exact Likelihood Evaluation," *Applied Statistics*, 37, 290–301.

Békéssy, A., Békéssy, P., and Komlos, J. (1972), "Asymptotic Enumeration of Regular Matrices," *Studia Scientiarum Mathematicarum Hungarica*, 7, 343–353.

Besag, J., and Clifford, P. (1989), "Generalized Monte Carlo Significance Tests," *Biometrika*, 76, 633–642.

Boyett, J. M. (1979), "Algorithm AS 144: Random $R \times S$ Tables With Given Row and Column Totals," *Applied Statistics*, 28, 329–332.

Brewer, K. R. W., and Hanif, M. (1983), *Sampling With Unequal Probabilities*, New York: Springer-Verlag.

Chen, S. X., and Liu, J. S. (1997), "Statistical Applications of the Poisson-Binomial and Conditional Bernoulli Distributions," *Statistica Sinica*, 7, 875–892.

Chen, X. H., Dempster, A. P., and Liu, J. S. (1994), "Weighted Finite Population Sampling to Maximize Entropy," *Biometrika*, 81, 457–469.

Chen, Y. (2001), "Sequential Importance Sampling With Resampling: Theory and Applications," unpublished Ph.D. dissertation, Stanford University, Dept. of Statistics.

Chen, Y., and Small, D. (2004), "Exact Tests for the Rasch Model via Sequential Importance Sampling," *Psychometrika*, in press.

Cobb, G. W., and Chen, Y.-P. (2003), "An Application of Markov Chain Monte Carlo to Community Ecology," *American Mathematical Monthly*, 110, 265–288.

Cody, M. L. (1983), "The Land Birds," in *Island Biogeography in the Sea of Cortez*, eds. T. J. Case and M. L. Cody, Berkeley, CA: University of California Press, pp. 210–245.

Connor, E. F., and Simberloff, D. (1979), "The Assembly of Species Communities: Chance or Competition," *Ecology*, 60, 1132–1140.

Cook, R. R., and Quinn, J. F. (1995), "The Influence of Colonization in Nested Species Subsets," *Oecologia*, 102, 413–424.

Diaconis, P., and Efron, B. (1985), "Testing for Independence in a Two-Way Table: New Interpretations of the Chi-Square Statistic," *The Annals of Statistics*, 13, 845–874.

Diaconis, P., and Gangolli, A. (1995), "Rectangular Arrays With Fixed Margins," in *Discrete Probability and Algorithms*, eds. D. Aldous, P. Diaconis, J. Spencer, and J. M. Steele, New York: Springer-Verlag, pp. 15–41.

Fischer, G. H., and Molenaar, I. W. (1995), *Rasch Models: Foundations, Recent Developments, and Applications*, New York: Springer-Verlag.

Forster, J. J., McDonald, J. W., and Smith, P. W. F. (1996), "Monte Carlo Exact Conditional Tests for Log-Linear and Logistic Models," *Journal of the Royal Statistical Society*, Ser. B, 58, 445–453.

Gail, M., and Mantel, N. (1977), "Counting the Number of $r \times c$ Contingency Tables With Fixed Margins," *Journal of the American Statistical Association*, 72, 859–862.

Gale, D. (1957), "A Theorem on Flows in Networks," *Pacific Journal of Mathematics*, 7, 1073–1082.

Glas, C. A. W., and Verhelst, N. D. (1995), "Testing the Rasch Model," in *Rasch Models: Their Foundations, Recent Developments and Applications*, eds. G. H. Fischer and I. W. Molenaar, New York: Springer-Verlag, pp. 69–95.

Good, I. J., and Crook, J. (1977), "The Enumeration of Arrays and a Generalization Related to Contingency Tables," *Discrete Mathematics*, 19, 23–65.

Holland, P. W., and Thayer, D. T. (1988), "Differential Item Performance and the Mantel–Haenszel Procedure," in *Test Validity*, eds. H. Wainer and H. I. Braun, Hillsdale, NJ: LEA, pp. 129–145.

Holmes, R. B., and Jones, L. K. (1996), "On Uniform Generation of Two-Way Tables With Fixed Margins and the Conditional Volume Test of Diaconis and Efron," *The Annals of Statistics*, 24, 64–68.

Kelderman, H. (1989), "Item Bias Detection Using Loglinear IRT," *Psychometrika*, 54, 681–697.

Kong, A., Liu, J. S., and Wong, W. H. (1994), "Sequential Imputations and Bayesian Missing Data Problems," *Journal of the American Statistical Association*, 89, 278–288.

Lehmann, E. L. (1986), *Testing Statistical Hypotheses*, New York: Wiley.

Liu, J. S., and Chen, R. (1998), "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, 93, 1032–1044.

Manly, B. F. (1995), "A Note on the Analysis of Species Co-Occurrences," *Ecology*, 76, 1109–1115.

Marshall, A. W., and Olkin, I. (1979), *Inequalities: Theory of Majorization and Its Applications*, New York: Academic Press.

Patefield, W. M. (1981), "An Efficient Method of Generating Random $r \times c$ Tables With Given Row and Column Totals," *Applied Statistics*, 30, 91–97.

Ponocny, I. (2001), "Nonparameteric Goodness-of-Fit Tests for the Rasch Model," *Psychometrika*, 66, 437–460.

Rao, A. R., Jana, R., and Bandyopadhyay, S. (1996), "A Markov Chain Monte Carlo Method for Generating Random $(0, 1)$-Matrices With Given Marginals," *Sankhyā*, Ser. A, 58, 225–242.

Rasch, G. (1960), *Probabilistic Models for Some Intelligence and Attainment Tests*, Copenhagen: Danish Institute for Educational Research.

Roberts, A., and Stone, L. (1990), "Island-Sharing by Archipelago Species," *Oecologia*, 83, 560–567.

Ryser, H. J. (1957), "Combinatorial Properties of Matrices of Zeros and Ones," *The Canadian Journal of Mathematics*, 9, 371–377.

Sanderson, J. G. (2000), "Testing Ecological Patterns," *American Scientist*, 88, 332–339.

Sanderson, J. G., Moulton, M. P., and Selfridge, R. G. (1998), "Null Matrices and the Analysis of Species Co-Occurrences," *Oecologia*, 116, 275–283.

Smith, P. W. F., Forster, J. J., and McDonald, J. W. (1996), "Monte Carlo Exact Tests for Square Contingency Tables," *Journal of the Royal Statistical Society*, Ser. A, 159, 309–321.

Snijders, T. A. B. (1991), "Enumeration and Simulation Methods for 0–1 Matrices With Given Marginals, *Psychometrika*, 56, 397–417.

Sullaway, F. J. (1982), "Darwin and His Finches: The Evolution of a Legend," *Journal of the History of Biology*, 15, 1–53.

Wang, B. Y. (1988), "Precise Number of $(0, 1)$-Matrices in $\mathcal{U}(R, S)$," *Scientia Sinica*, Ser. A, 1–6.

Wang, B. Y., and Zhang, F. (1998), "On the Precise Number of $(0, 1)$-Matrices in $\mathcal{U}(R, S)$," *Discrete Mathematics*, 187, 211–220.

Wilson, J. B. (1987), "Methods for Detecting Non-Randomness in Species Co-Occurrences: A Contribution," *Oecologia*, 73, 579–582.