A Simple Method for Generating Correlated Binary Variates

Chul Gyu PARK, Taesung PARK, and Dong Wan SHIN

Correlated binary data are frequently analyzed in studies of repeated measurements, reliability analysis, and others. In such studies correlations among binary variables are usually nonnegative. This article provides a simple algorithm for generating an arbitrary dimensional random vector of nonnegatively correlated binary variables. In some frequently encountered situations the algorithm reduces to explicit expressions. The correlated binary variables are generated from correlated Poisson variables. The key idea lies in the property that any Poisson random variable can be expressed as a convolution of other independent Poisson random variables. The binary variables have desired correlations by sharing common independent Poisson variables.

KEY WORDS: Generalized estimating equations; Poisson variables; Random number generation.

1. INTRODUCTION

Correlated binary data are encountered in many areas such as clinical trials with repeated measures, analyses of system reliability, genetic disease studies with observations from siblings of common parents, and others. Analysis of correlated data has been an important issue in biostatistics, particularly when measurements are taken repeatedly over time from the same subject. In many clinical trials binary responses, for example, the presence or absence of some disease symptoms, are usually observed. The theory and methods on this subject were comprehensively reviewed in Prentice (1988).

A typical area is found in the generalized estimating equation (GEE) approach suggested by Liang and Zeger (1986) and Zeger, Liang, and Albert (1988). This approach does not specify a complete form of the joint distribution of measurements within the same experimental unit, but it requires the specification of the first two moments including the correlation structure. Although the asymptotic results of the GEE estimators are found in Liang and Zeger (1986), their small-sample properties are not well studied. In order to evaluate the finite sample properties of estimators, we need to conduct a Monte Carlo simulation that requires generating binary random variates having specified mean structures and correlation structures. Another area is found in the analysis of system reliability, in which the system is composed of dependent components. When analytical evaluation of system performance is difficult, one needs to generate correlated binary variables for a Monte Carlo study.

There have been several approaches for generating correlated binary variables. Let p_j denote the marginal expectation for binary random variable $Z_j, j = 1, \ldots, k$ and let $\rho_{ij} = \operatorname{corr}(Z_i, Z_j)$. Bahadur (1961) suggested a parametric model that is expressed as a complicated joint mass function for $Z_1, Z_2, \ldots, Z_k$. Because Bahadur's distribution is computationally difficult to handle particularly for large k unless many higher order correlations are zero, this approach is not suitable for generating a high-dimensional correlated binary random vector.

A more appealing algorithm for simulating correlated binary random variables was proposed by Emrich and Piedmonte (1991). Let $\Phi(x_1, x_2; r)$ denote the cumulative distribution function for a standard bivariate normal distribution with correlation coefficient r. The algorithm first solves the equations

$$\Phi[z(p_i), z(p_j); r_{ij}] = \rho_{ij}(p_i q_i p_j q_j)^{1/2} + p_i p_j \qquad (1.1)$$

for r_{ij} $(i = 1, \ldots, k-1; j = i+1, \ldots, k)$, where $q_i = 1 - p_i$ and $z(p)$ denotes the pth quantile of the standard normal distribution. The next step is to generate a k-dimensional multivariate normal random vector $Y = (Y_1, \ldots, Y_k)'$ with zero means, unit variances, and correlation matrix $R = [r_{ij}]$. The desired binary variables are obtained by setting $Z_i = 1$ if $Y_i \le z(p_i)$ and $Z_i = 0$ otherwise. A drawback of Emrich and Piedmonte's algorithm is that we must solve the nonlinear equation (1.1), which requires numerical integrations in evaluating Φ. Recently, Lee (1993) presented a method for generating binary random sequences that works for any marginal distributions and any set of consistent odds ratios. He also provided two algorithms, based on linear programming and the concept of copula, specifically for generating binary variables having equal correlation. However, these approaches also require solving a large number of nonlinear equations. More recently Gange (1995) proposed an iterative procedure for generating dependent multivariate categorical variates using the iterative proportional fitting algorithm. In this procedure, first, dependence of the categorical variables is formulated by contingency tables. Next, a joint probability is constructed by fitting a log-linear model to the contingency tables. Finally, dependent categorical variables are generated by comparing a simulated uniform random numbers with the joint probability. This procedure is also discouraging in the sense that it requires another iterative fitting procedure.

Chul Gyu Park is Assistant Professor, Department of Applied Statistics, University of Suwon, 445-743, Suwon, South Korea. Taesung Park is Associate Professor, Department of Statistics, Hankuk University of Foreign Studies, Seoul, 449-791, South Korea, and the National Institute of Child Health and Human Development, Bethesda, MD 20892. Dong Wan Shin is Associate Professor, Department of Statistics, Ewha Womans University, Seoul, 120-750, South Korea. The authors thank an associate editor, two referees, and Professors Beong Soo So and You Sung Park for valuable suggestions and comments. This work was supported by Directed Research grant 94-0701-01-01-3 from the Korea Science and Engineering Foundation.

Therefore, it is worth developing a simple procedure for generating correlated binary variables. In most experiments with repeated measurements observations are naturally exposed to nonnegative correlations because they come from the same experimental unit. For simulating binary random variables in this particular situation we propose a simple algorithm that requires no equation-solving. The key idea of the proposed algorithm lies in the property that any Poisson random variable can be expressed as a convolution of several other independent Poisson random variables.

2. THE ALGORITHM

In this section we propose an algorithm for generating a random vector $(Z_1, \ldots, Z_k)'$ of binary variables such that $E[Z_i] = p_i$ and $\operatorname{corr}(Z_i, Z_j) = \rho_{ij} \geq 0, i \neq j$. In practical situations the probabilities and correlation parameters might be functions of covariates. We first consider $k = 2$ to present our idea. Let $X(\alpha)$ denote a Poisson random variable having mean $\alpha \geq 0$. By convention $X(0) = 0$. Hereafter we assume that $X(\cdot)$'s are mutually independent if they appear with different subscripts. Consider two random variables defined by

$$Y_1 = X_1(\alpha_{11} - \alpha_{12}) + X_3(\alpha_{12})$$

and

$$Y_2 = X_2(\alpha_{22} - \alpha_{12}) + X_3(\alpha_{12}), \tag{2.1}$$

where α_{11}, α_{22}, and α_{12} are nonnegative constants. Obviously, Y_1 and Y_2 follow Poisson distributions with means α_{11} and α_{22}, respectively. Because they share a common term $X_3(\alpha_{12})$, the two Poisson variables Y_1 and Y_2 are nonnegatively correlated. For $i = 1, 2$ set $Z_i = I_{\{0\}}(Y_i)$, where I_A is the indicator function of a set A such that $I_A(y) = 1$ if $y \in A$ and $I_A(y) = 0$ if $y \notin A$. Because Y_1 and Y_2 are nonnegatively correlated, so are Z_1 and Z_2. It is clear that the correlation coefficient ρ_{12} of Z_1 and Z_2 is increasing in α_{12}. The constants α_{11}, α_{22}, and α_{12} can be chosen so that $E(Z_1) = p_1, E(Z_2) = p_2$, and $\operatorname{corr}(Z_1, Z_2) = \rho_{12}$. The constants α_{ii} are given by $-\log p_i, i = 1, 2$ and the constant α_{12} is given by (2.3) as explained below. Because

$$E(Z_i) = E(Z_i^2) = P(X_i = X_3 = 0) = e^{-\alpha_{ii}} = p_i, \quad i = 1, 2$$

and

$$\begin{aligned} E(Z_1 Z_2) &= P[X_1 = X_2 = X_3 = 0] \\ &= \exp[-(\alpha_{11} + \alpha_{22} - \alpha_{12})] = p_1 p_2 e^{\alpha_{12}}, \end{aligned}$$

we have

$$\operatorname{var}(Z_i) = p_i q_i$$

and

$$\operatorname{cov}(Z_1, Z_2) = p_1 p_2 (e^{\alpha_{12}} - 1) = \rho_{12} (p_1 q_1 p_2 q_2)^{1/2}.$$

Therefore, Z_1 and Z_2 have correlation coefficient

$$\rho_{12} = p_1 p_2 (e^{\alpha_{12}} - 1)/(p_1 q_1 p_2 q_2)^{1/2}. \tag{2.2}$$

Solving (2.2) gives the desired α_{ij}'s:

$$\alpha_{ij} = \log[1 + \rho_{ij} \{q_i p_i^{-1} q_j p_j^{-1}\}^{1/2}], \tag{2.3}$$

where

$$q_i = 1 - p_i, \qquad i, j = 1, 2.$$

Of course, the case with $\alpha_{12} = 0$ corresponds to independence between Z_1 and Z_2.

Note that $E(Z_1 Z_2) = P(Z_1 = Z_2 = 1) \leq P(Z_i = 1) = p_i, i = 1, 2$, and we have $\operatorname{cov}(Z_1, Z_2) \leq p_1 q_2$ and $\operatorname{cov}(Z_1, Z_2) \leq p_2 q_1$. Hence, as described in Emrich and Piedmonte (1991), ρ_{12} is not free over $[-1, 1]$. That is,

$$\rho_{12} \leq [p_2 q_1/(p_1 q_2)]^{1/2} \quad \text{and} \quad \rho_{12} \leq [p_1 q_2/(p_2 q_1)]^{1/2}. \tag{2.4}$$

This inequality yields the relation $\alpha_{12} \leq \log(p_i^{-1}) = \alpha_{ii}, i = 1, 2$. Therefore, (2.1) generates all nonnegatively correlated bivariate binary random vector.

For general $k \geq 2$ consider a set of k Poisson random variables $Y_1, Y_2, \ldots, Y_k$ that are partial sums of independent Poisson variables, say, $X_1(\beta_1), \ldots, X_\tau(\beta_\tau)$, for some nonnegative integer τ and nonnegative real numbers $\beta_1, \ldots, \beta_\tau$. Some of $X_1(\beta_1), \ldots, X_\tau(\beta_\tau)$ may appear simultaneously in several Y_j's. The expected values and correlation structure of the binary variables $Z_1 = I_{\{0\}}(Y_1), \ldots, Z_k = I_{\{0\}}(Y_k)$ can be met by properly controlling the pattern of the simultaneous appearance of $X_1(\beta_1), \ldots, X_\tau(\beta_\tau)$ and the magnitudes of $\beta_1, \ldots, \beta_\tau$. The algorithm below describes how to determine $\tau, \beta_1, \ldots, \beta_\tau$ and the pattern of the partial sums for generating a k-dimensional binary random vector $(Z_1, \ldots, Z_k)'$ with mean vector $(p_1, \ldots, p_k)'$ and correlation matrix $R = [\rho_{ij}]$ with $\rho_{ij} \geq 0$. In order to clarify the idea we simply sketch the algorithm. A detailed implementation of the algorithm is given in the Appendix.

Algorithm: Step 0. Compute α_{ij} in (2.3) for $1 \leq i, j \leq k$. Let $l = 0$.

Step 1. Let $l = l + 1$. Let $T_l = \{\alpha_{ij}\colon \alpha_{ij} > 0, 1 \leq i, j \leq k\}$. Let $\beta_l = \alpha_{rs}$ be the smallest element in the set T_l. If $\alpha_{rr} = 0$ or $\alpha_{ss} = 0$, then stop. Otherwise, choose an index set S_l containing $\{r, s\}$ and satisfying $\alpha_{ij} > 0$ for all $\{i, j\} \in S_l$. The set S_l is chosen to have as many elements as possible.

Step 2. For all $\{i, j\} \in S_l$ replace α_{ij} with $\alpha_{ij} - \beta_l$. If all $\alpha_{ij} = 0$, then go to Step 3. Otherwise, go to Step 1.

Step 3. Let $\tau = l$. For $i = 1, 2, \ldots, k$ let $Y_i = \sum_{l=1}^{\tau} X_l(\beta_l) I_{S_l}(i)$ and set $Z_i = I_{\{0\}}(Y_i)$.

Note that in Step 1 the minimum β_l is chosen to make sure that all of the Poisson random variables have nonnegative means. It is possible that α_{rs} and S_l in Step 1 may not be uniquely determined. In that case we may choose α_{rs} and S_l arbitrarily. Validation of the algorithm will be discussed after the following numerical example.

Example. Consider a case of $k = 3, p_1 = .9, p_2 = .8, p_3 = .7, \rho_{12} = .1, \rho_{13} = .5$, and $\rho_{23} = .5$. Steps 1 and 2 are illustrated in the table below for $[\alpha_{ij}, j \geq i]$ matrices for $l = 1, 2, \ldots, 6$. In each matrix $[\alpha_{ij}]$ T_l consists of all positive numbers. Underlined bold numbers are β_l. Bold numbers correspond to pair of indices (i, j) in S_l.

	$l=1$			$l=2$			$l=3$			$l=4$			$l=5$			$l=6$		
	.105	**.017**	**.104**	**.088**	.000	**.087**	**.002**	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
		.223	**.152**		.206	.135		.207	.135		**.207**	**.135**		**.072**	.000		.000	.000
			.357			**.340**			.253			**.253**			.118			**.118**
(r, s)		(1, 2)			(1, 3)			(1, 1)			(2, 3)			(2, 2)			(3, 3)	
S_l		{1, 2, 3}			{1, 3}			{1}			{2, 3}			{2}			{3}	

The numbers under "$l = 1$" are computed from (2.3). Because all $\alpha_{ij} > 0, T_1 = \{\alpha_{ij}\colon 1 \le i \le j \le 3\} = \{.105, .017, .104, .223, .152, .357\}$. In T_1, $\alpha_{12} = .017$ is the minimum element. Therefore, $\beta_1 = .017$ and $(r, s) = (1, 2)$. Note that $S_1 = \{1, 2, 3\}$ because $\{r, s\} = \{1, 2\} \in S_1$ and $\alpha_{ij} > 0$ for $1 \le i \le j \le 3$. Now, update $\alpha_{ij} = \alpha_{ij} - \beta_1$ for $(i, j) \in \{1, 2, 3\}$ and $i \le j$.

The resulting numbers are given under "$l = 2$." We have $T_2 = \{.088, .087, .206, .135, .340\}$, the set of positive numbers under "$l = 2$." The minimum element in T_2 is $\beta_2 = \alpha_{13} = .087$. Now, $(r, s) = (1, 3)$. We have $S_2 = \{1, 3\}$. Note that $2 \notin S_2 = \{1, 3\}$ because $\alpha_{12} = 0$. Next, updating the numbers α_{ij} for $(i, j) \in S_2 = \{1, 3\}$ gives $\alpha_{11} = \alpha_{11} - \beta_2 = .088 - .087 = .002, \alpha_{13} = \alpha_{13} - \beta_2 = .087 - .087 = 0, \alpha_{33} = .340 - .087 = .253$. Note that α_{22} does not change because $2 \notin S_2$. The resulting numbers are given under "$l = 3$." The algorithm continues until all α_{ij}'s become zero. Note that $\alpha_{rr} > 0$ and $\alpha_{ss} > 0$ in each step. Using β_l and $S_l, l = 1, \ldots, 6$ we get

$$\begin{array}{llllll}
Y_1 = X_1(.017) & +X_2(.087) & +X_3(.002), & & & \\
Y_2 = X_1(.017) & & & +X_4(.135) & +X_5(.072), & \\
Y_3 = X_1(.017) & +X_2(.087) & & +X_4(.135) & & +X_6(.118).
\end{array}$$

The correlated binary variables are $Z_i = I_{\{0\}}(Y_i), i = 1, 2, 3$.

One can compute the joint distribution of (Z_1, Z_2, Z_3). For example, $P(Z_1 = 1, Z_2 = 1, Z_3 = 1) = P\{X_l = 0, l = 1, \ldots, 6\} = \prod_{l=1}^{6} \exp(-\beta_l) = .6505$. By enumerating all possibilities, the joint distribution is tabulated as follows:

(z_1, z_2, z_3)	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
probability	.0319	.0001	.0668	.0012	.1197	.0483	.0815	.6505

The binary random vector (Z_1, Z_2, Z_3) can also be simulated directly from this joint distribution. Let U be a uniform random number between 0 and 1. If $U \le .0319$, then $(Z_1, Z_2, Z_3) = (0, 0, 0)$. If $.0319 < U \le .0319 + .0001$, then $(Z_1, Z_2, Z_3) = (0, 0, 1)$ and so on.

We now give validation for the algorithm. Scrutinizing the algorithm we first observe that

$$E[Y_i] = \sum_{l=1}^{\tau} E[X_l(\beta_l) I_{S_l}(i)] = \sum_{l=1}^{\tau} \beta_l I_{S_l}(i) = \alpha_{ii}$$

and hence $E(Z_i) = p_i$ from (2.3). In Step 3 we observe that $X_l(\beta_l)$ is added to both Y_i and Y_j if i and j both belong to S_l. Thus the total amount commonly taken by Y_i and Y_j is

$$\sum_{l=1}^{\tau} X_l(\beta_l) I_{S_l}(i) I_{S_l}(j).$$

From Steps 1 and 2

$$E\left[\sum_{l=1}^{\tau} X_l(\beta_l) I_{S_l}(i) I_{S_l}(j)\right] = \sum_{l=1}^{\tau} \beta_l I_{S_l}(i) I_{S_l}(j) = \alpha_{ij}.$$

Therefore, $\operatorname{corr}(Z_i, Z_j) = \rho_{ij}$ from (2.3).

Note that the criteria for the choices of α_{rs} and S_l in Step 1 are to maintain the updated α_{ii}'s as large as possible throughout the algorithm. If the feasibility condition (2.4) is violated, eventually we shall encounter $\alpha_{rr} = 0$ or $\alpha_{ss} = 0$ at Step 1 and the algorithm stops, failing to find the desired correlated variables. We believe that the algorithm works for most practical high-dimensional nonnegative correlation structures.

3. SOME SIMPLE CASES

In this section we apply the algorithm to derive a simple explicit expression for the correlated binary variates Z_i in frequently encountered situations. In many practical cases α_{ij} in the algorithm has the following monotone property:

$$\begin{array}{lll}
\text{(M1)} & \alpha_{1j} \ge \alpha_{1,j+1}, & 1 \le j < k, \\
\text{(M2)} & \alpha_{ik} \ge \alpha_{i-1,k}, & 1 < i \le k, \\
\text{(M3)} & \alpha_{ij} - \alpha_{i,j+1} \ge \alpha_{i-1,j} - \alpha_{i-1,j+1}, & \\
 & & 1 \le j \le k - i, 1 < i \le k.
\end{array}$$

For AR(1) correlation $\rho_{ij} = \rho^{|j-i|}$, (M1)–(M2) reduces to

$$\rho \le \min\{[(q_j p_{j+1})/(p_j q_{j+1})]^{1/2}, \qquad j = 1, \ldots, k-1\}$$

and a sufficient condition for (M3) is

$$\rho \leq \min\{[(q_j p_{j-1})/(p_j q_{j-1})]^{1/2}, \qquad j = 2, \ldots, k\},$$

which, as discussed in (2.4), holds for all feasible binary random vectors with AR(1) correlation and possibly different means.

When p_i's are all the same (time-stationary case) and $\rho_{ij} = \rho_{|j-i|}$ (banded correlation) a sufficient condition for (M1)–(M3) is

$$\rho_j^2 \leq \rho_{j-1}\rho_{j+1}, \qquad j = 2, \ldots, k-1.$$

Under (M1)–(M3) the algorithm proceeds as in the Example to yield an explicit expression for Z_i's given in the following theorem.

Theorem. Let the conditions (M1)–(M3) hold. Then $Z_i = I_{\{0\}}(Y_i)$ has mean p_i and Z_i and Z_j have correlation ρ_{ij}, where

$$Y_i = \sum_{t=1}^{i} \sum_{s=1}^{k-i+1} X_{ts}(\beta_{ts}), \qquad i = 1, \ldots, k,$$

$$\beta_{11} = \alpha_{1k}, \qquad \beta_{1s} = \alpha_{1,k-s+1} - \alpha_{1,k-s+2}, \qquad s = 2, \ldots, k,$$

$$\begin{aligned}
&\beta_{t1} = \alpha_{tk} - \alpha_{t-1,k},\\
&\beta_{ts} = (\alpha_{t,k-s+1} - \alpha_{t,k-s+2}) - (\alpha_{t-1,k-s+1} - \alpha_{t-1,k-s+2}),\\
&\quad s = 2, \ldots, k-t+1, t = 2, \ldots, k-1,\\
&\beta_{k1} = \alpha_{kk} - \alpha_{k-1,k}.
\end{aligned}$$

Proof. Note that $\sum_{s=1}^{k-i+1} \beta_{1s} = \alpha_{1i}$ and $\sum_{s=1}^{k-i+1} \beta_{ts} = \alpha_{ti} - \alpha_{t-1,i}, t = 2, \ldots, k$. Hence $\sum_{t=1}^{i} \sum_{s=1}^{k-j+1} \beta_{ts} = \alpha_{ij}, 1 \leq i \leq j \leq k$. Therefore, for $1 \leq i \leq j \leq k$

$$\begin{aligned}
Y_i &= \sum_{t=1}^{i} \sum_{s=1}^{k-j+1} X_{ts}(\beta_{ts}) + \sum_{t=1}^{i} \sum_{s=k-j+2}^{k-i+1} X_{ts}(\beta_{ts})\\
&= X_0(\alpha_{ij}) + X_i(\alpha_{ii} - \alpha_{ij}),\\
Y_j &= \sum_{t=1}^{i} \sum_{s=1}^{k-j+1} X_{ts}(\beta_{ts}) + \sum_{t=i+1}^{j} \sum_{s=1}^{k-j+1} X_{ts}(\beta_{ts})\\
&= X_0(\alpha_{ij}) + X_j(\alpha_{jj} - \alpha_{ij}).
\end{aligned}$$

By the same arguments between (2.1) and (2.3) $E(Z_i) = p_i$ and $\operatorname{corr}(Z_i, Z_j) = \rho_{ij}$, and this completes the proof.

For the time-stationary compound symmetric correlation ($p_1 = p_2 = \cdots = p_k = p, \rho_{ij} = \rho, i \neq j$) a further simplification is possible. Let $\alpha = \log(1 + \rho q p^{-1})$ and let $\beta = \log p^{-1}$. We have $\alpha_{ii} = \beta$ and $\alpha_{ij} = \alpha$, for $i \neq j$. Hence $\beta_{11} = \alpha$ and $\beta_{t,k-t+1} = \beta$ and other β_{ts} are all zero. Therefore,

$$Y_i = X_{11}(\alpha) + X_{i,k-i+1}(\beta - \alpha), \qquad i = 1, \ldots, k.$$

In this case we need $\tau = k + 1$ Poisson random variables. However, for the AR(1) correlation $\beta_{ts} \neq 0$ for all (t, s), and we need $\tau = k(k+1)/2$ Poisson random variables.

4. DISCUSSION

Unlike the algorithms of Emrich and Piedmonte (1991), Lee (1993), and Gange (1995) the proposed algorithm requires no complicated numerical procedures such as equation-solving or numerical integration. Under many frequently encountered correlations the algorithm reduces to explicit simple expressions. However, the algorithm of Emrich and Piedmonte (1991) still deserves our interest because it can generate more generally correlated binary random variables such as negatively correlated binary random variables.

It is worth comparing the computational efficiency of our method and other procedures. Generating correlated binary variates consists of two steps: parameter determination and data generation. Our method has a simpler parameter determination step in that it requires no equation-solving. In the data generation step our method requires, in the worst case, $k(k+1)/2$ independent Poisson variables, while Emrich and Piedmonte's method requires a k-dimensional normal random vector. A brief comparison of the computational efficiency of the two methods is made in the setup: AR(1) correlation $\rho_{ij} = 0.5^{|i-j|}, p_1 = p_2 = \cdots = p_k = 0.5, k = 2, 4, 8, 16$. In implementing the algorithms we use IMSL Fortran subroutines: ANORIN, BNRDF, ZBREN, RNMVN, and RNPOI for normal percentile, bivariate normal probability, nonlinear equation-solving, multivariate normal random number generation, and Poisson random number generation, respectively. Average computing times for generating one sample based on 1,000 replications in a 90 MHz Pentium machine are listed below.

	Parameter generation					Data generation				
k	2	4	8	16		2	4	8	16	
Our method	16	49	157	445	(10^{-7})	5	16	71	183	(10^{-5})
Emrich and Piedmonte	10	87	264	910	(10^{-4})	10	16	27	65	(10^{-5})

The numbers in parentheses represent time units in seconds. The table reveals that our method has a much faster parameter determination step, and that, when k is large, Emrich and Piedmonte's method has a more efficient data generation step than our method. However, in the case of moderate k, as in the repeated measures studies, our method is better because it has a simpler parameter generation step and comparable data generation step. Furthermore, our method is more efficient than the methods of Lee (1993) and Gange (1995) because they first determine the joint distribution, which requires computations of order 2^k.

It is interesting to observe that the Poisson assumption in our method is not crucial. Any other discrete distributions that are closed under summation can also be used. The class of these distributions is characterized by the infinitely divisible property of random variables whose characteristic functions are of the type $\phi_\alpha(t) = [\phi_1(t)]^\alpha = \exp[\alpha \log \phi_1(t)], \alpha \geq 0$ for some characteristic function $\phi_1(t)$. See Chow and Teicher (1988, p. 433). Recall that the Poisson assumption is used only through the probability of zero together with closedness under summation. Hence the distribution with characteristic function $\phi_\alpha(t)$ can also be used subject to the appropriate calculation with $P(W_1(\alpha) = 0)$ instead of $P(X_1(\alpha) = 0) = e^{-\alpha}$.

Often, the dependencies between pairs of binary variables $\{Z_i, Z_j\}$ are formulated by the odd ratios $\gamma_{ij} = [(p_i - p_{ij})(p_j - p_{ij})]^{-1} p_{ij}(1 - p_i - p_j + p_{ij})$, where $p_{ij} = P(Z_i = Z_j = 1)$. Note that, given the marginal probabilities p_i and odd ratios γ_{ij}, the joint probability p_{ij} and hence the correlation coefficient ρ_{ij} can be determined by solving a quadratic equation. Hence our algorithm can also be applied to generate correlated binary variables having given odd ratios.

APPENDIX: DETAILED ALGORITHM

The index set S_l is Step 1 in the algorithm may not be uniquely determined. A simple way of choosing S_l is given in the detailed implementation of the algorithm below.

Step 0. Initialization:

Compute α_{ij} in (2.3) for $1 \leq i, j \leq k$. Let $l = 0$ and let $\alpha^1_{ij} = \alpha_{ij}, 1 \leq i, j \leq k$.

Step 1. Let $l = l + 1$.

Determine β_l and (r, s):

$$\text{Let } \beta_l = \alpha^l_{rs} = \min\{\alpha^l_{ij}\colon\ \alpha^l_{ij} > 0, \qquad 1 \leq i, j \leq k\}.$$

Check the feasibility:

If $\alpha_{rr} = 0$ or $\alpha_{ss} = 0$, then stop.

Determine S_l:

Let $S^0_l = \{r, s\}$.

For $i = 1, 2, \ldots, k$, let

$$\begin{aligned} S^i_l &= S^{i-1}_l \cup \{i\} \quad \text{if } \alpha^l_{ij} > 0 \quad \text{for all} \quad j \in S^{i-1}_l \\ &= S^{i-1}_l, \quad \text{otherwise.} \end{aligned}$$

Let $S_l = S^k_l$.

Step 2. Update α_{ij}'s:

$$\begin{aligned} \alpha^{l+1}_{ij} &= \alpha^l_{ij} - \beta_l, \quad \text{for all} \quad i \in S_l \text{ and } j \in S_l \\ &= \alpha^l_{ij}, \quad \text{for all} \quad i \notin S_l \text{ or } j \notin S_l. \end{aligned}$$

If all $\alpha^{l+1}_{ij} = 0$ for $1 \leq i, j \leq k$, go to Step 3. Otherwise go to Step 1.

Step 3. Let $\tau = l$. For $i = 1, 2, \ldots, k$, let

$$Y_i = \sum_{l=1}^{\tau} X_l(\beta_l) I_{S_l}(i) \text{ and } Z_i = I_{\{0\}}(Y_i).$$

[Received April 1995. Revised December 1995.]

REFERENCES

Bahadur, R. R. (1961), "A Representation of the Joint Distribution of Responses to n Dichotomous Items," in *Studies in Item Analysis and Prediction* (Stanford Mathematical Studies in the Social Sciences VI), ed. H. Solomon, Stanford, CA: Stanford University Press.

Chow, Y. S., and Teicher, H. (1988), *Probability Theory, Independence, Interchangeability, Martingales* (2nd ed.), New York: Springer-Verlag.

Emrich, L. J., and Piedmonte, M. R. (1991), "A Method for Generating High-Dimensional Multivariate Binary Variables," *The American Statistician*, 45, 302–304.

Gange, S. J. (1995), "Generating Multivariate Categorical Variates Using the Iterative Proportional Fitting Algorithm," *The American Statistician*, 49, 134–138.

Lee, A. J. (1993), "Generating Random Binary Deviates Having Fixed Marginal Distributions and Specified Degrees of Association," *The American Statistician*, 47, 209–215.

Liang, K. Y., and Zeger, S. L. (1986), "Longitudinal Data Analysis Using Generalized Linear Models," *Biometrika*, 73, 13–22.

Prentice, R. L. (1988), "Correlated Binary Regression with Covariates Specific to Each Binary Observation," *Biometrics*, 44, 1033–1048.

Zeger, S. L., Liang, K. Y., and Albert, P. S. (1988), "Models for Longitudinal Data: A Generalized Estimating Equation Approach," *Biometrics*, 44, 1049–1060.