Computational Statistics manuscript No.
(will be inserted by the editor)

Computing p-values in conditional independence models for a contingency table

Masahiro Kuroda · Hiroki Hashiguchi · Shigakazu Nakagawa

Received: date / Revised version: date

Abstract We present a Markov chain Monte Carlo (MCMC) method for generating Markov chains using Markov bases for conditional independence models for a four-way contingency table. We then describe a Markov basis characterized by Markov properties associated with a given conditional independence model and show how to use the Markov basis to generate random tables of a Markov chain. The estimates of exact p-values can be obtained from random tables generated by the MCMC method. Numerical experiments examine the performance of the proposed MCMC method in comparison with the χ^2 approximation using large sparse contingency tables.

Keywords Markov basis – Markov property – sparse contingency table – Markov chain Monte Carlo

1 Introduction

The use of asymptotic tests relied on large-sample approximations to the sample distributions of test statistics is unreliable when a contingency table is large and sparse. Exact tests evaluating statistical significance using p-values are preferred over asymptotic tests based on large sample approximations, such as χ^2 approximations. However, computing exact p-values is harder compared with computing asymptotic p-values, since the former requires enumeration of all contingency tables conditionally on the set

This research is supported by the Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (C), No 20500263.

Masahiro Kuroda
Department of Socio-Information, Okayama University of Science, 1-1 Ridaicho, Okayama 700-0005, JAPAN
E-mail: kuroda@soci.ous.ac.jp

Hiroki Hashiguchi
Graduate School of Science and Engineering, Saitama University, 255 Shimo-Okubo, Sakura, Saitama 338-8570, JAPAN

Shigakazu Nakagawa
Department of Computer Science and Mathematics, Kurashiki University of Science and the Arts, 2640 Nishinoura, Tsurajima-cho, Kurashiki-shi, Okayama 712-8505, JAPAN

of marginals. For a detailed review of exact inference for contingency tables, refer to Agresti [1][2].

When the exact p-value can not be calculated due to the infeasibility of completely enumerating all contingency tables, MCMC is a valuable tool for evaluating p-values. Diaconis and Sturmfels [10] proposed an algebraic algorithm for constructing MCMC for sampling from conditional distributions. The algorithm generates random tables by the use of Markov bases computed using Gröbner bases. Further, MCMC with a Markov basis generates an irreducible Markov chain. Dobra [11] presented explicit formulae that identify Markov bases for decomposable graphical models and showed an algorithm for generating Markov bases dynamically. Forster *et al.* [13] used Gibbs sampling to perform exact tests for goodness of fit of the all-two-way interaction model for a 2^d contingency table. Caffo and Booth [4] suggested a different MCMC approach that computes p-values by making use of importance sampling with rounded normal deviates. Although Gibbs sampling and the MCMC method using importance sampling enable the computation of exact p-values in various log-linear models for multi-way tables, the Markov chains generated may be reducible.

Our aim is to generate a Markov basis characterized by the Markov properties of Lauritzen [17] associated with a given conditional independence model for a four-way contingency table and develop a MCMC method for generating random tables by using the Markov basis. In Section 2, we describe conditional independence models for four-way contingency tables and show the Markov properties that form conditional independence structures for these models. In Section 3, we define the Markov bases for conditional independence models. In Section 4, we present an algorithm for generating a random table by employing the Markov basis. In Section 5, we give MCMC with the Markov basis for generating random tables of a Markov chain. The estimates of exact p-values can be calculated from random tables generated by MCMC. Numerical experiments in Section 6 examine the performance of the proposed MCMC method in comparison with χ^2 approximation. In Section 7, we present our concluding remarks.

2 Conditional independence models for a four-way contingency table

Consider an $I \times J \times K \times L$ contingency table $\mathbf{n} = \{n_{abcd}\}$ with non-negative integer cell entries formed from categorical variables A, B, C and D. To denote conditional independence models for $\mathbf{n}$, we use notations pertaining to their minimal sufficient statistics. For example, $[ABC][BCD]$ denotes the model with conditional independence between A and D given (B, C). Using the notation of Dawid [9], its relationship can be expressed as $A \perp D | (B, D)$. The minimal sufficient statistics for $[ABC][BCD]$ are marginals $\mathbf{n}_{ABC} = \{n_{abc+}\}$ and $\mathbf{n}_{BCD} = \{n_{+bcd}\}$, where

$$n_{abc+} = \sum_{d} n_{abcd}, \qquad n_{+bcd} = \sum_{a} n_{abcd}.$$

For a four-way contingency table, there exist four conditional independence models, as described in Darroch *et al.* [8]. Table 1 shows these conditional independence models and their interpretations. In Table 2, the conditional independence relationships for these models in the form of Markov properties are presented. Figure 1 illustrates the graphical representations for the conditional independence models in Table 1.

Lauritzen [17] and Sundberg [18] showed that the exact conditional distributions of $\mathbf{n}$ for M_1 to M_4 have hypergeometric distributions conditionally on the sets of marginal constraints:

$$\pi_{M_1}(\mathbf{n}) = \frac{\prod_{a,b,c} n_{abc+}! \prod_{b,c,d} n_{+bcd}!}{\prod_b n_{+bc+}! \prod_{a,b,c,d} n_{abcd}!}, \tag{1}$$

$$\pi_{M_2}(\mathbf{n}) = \frac{\prod_{a,b} n_{ab++}! \prod_{b,c,d} n_{+bcd}!}{\prod_b n_{+b++}! \prod_{a,b,c,d} n_{abcd}!}, \tag{2}$$

$$\pi_{M_3}(\mathbf{n}) = \frac{\prod_{a,b} n_{ab++}! \prod_{b,c} n_{+bc+}! \prod_{c,d} n_{++cd}!}{\prod_b n_{+b++}! \prod_c n_{++c+}! \prod_{a,b,c,d} n_{abcd}!}, \tag{3}$$

$$\pi_{M_4}(\mathbf{n}) = \frac{\prod_{a,b} n_{ab++}! \prod_{b,c} n_{+bc+}! \prod_{b,d} n_{+b+d}!}{\left\{\prod_b n_{+b++}!\right\}^2 \prod_{a,b,c,d} n_{abcd}!}. \tag{4}$$

Then, the MCMC method draws random tables from the above exact conditional distributions and is applied to estimate exact p-values from the generated random tables.

3 Markov bases for conditional independence models

Let $T(M)$ denote the set of all tables having the marginal totals of sufficient statistics of model M. To generate table $\mathbf{n}' \in T(M)$, we use the data swapping technique of Dalenius and Reiss [7] such that cell entries are moved from one cell to the other, while the fixed marginals of sufficient statistics of M are left unchanged. A data swap associated with $\mathbf{n}' \in T(M)$ is an array $\mathbf{f} = \{f_{abcd}\}$ with $f_{abcd} \in \{0, \pm 1, \pm 2, \ldots\}$ for all a, b, c, d.

Definition 1 A move $\mathbf{f} = \{f_{abcd}\}$ for model M is a data swap that preserves the marginals of sufficient statistics of M.

Then, we have $\mathbf{n} + \mathbf{f} \in T(M)$ if and only if $n_{abcd} + f_{abcd} \geq 0$ for all a, b, c, d. From the marginal constraints for M_1 to M_4, we have

$$M_1: \sum_d \sum_{a,b,c} f_{abcd} = \sum_{a,b,c} f_{abc+} = 0, \quad \sum_d n_{abcd} + f_{abcd} = n_{abc+}, \tag{5}$$

$$\sum_a \sum_{b,c,d} f_{abcd} = \sum_{b,c,d} f_{+bcd} = 0, \quad \sum_a n_{abcd} + f_{abcd} = n_{+bcd}, \tag{6}$$

$$M_2: \sum_a \sum_{b,c,d} f_{abcd} = \sum_{b,c,d} f_{+bcd} = 0, \quad \sum_a n_{abcd} + f_{abcd} = n_{+bcd}, \tag{7}$$

$$\sum_{c,d} \sum_{a,b} f_{abcd} = \sum_{a,b} f_{ab++} = 0, \quad \sum_{c,d} n_{abcd} + f_{abcd} = n_{ab++}, \tag{8}$$

$$M_3: \sum_{c,d} \sum_{a,b} f_{abcd} = \sum_{a,b} f_{ab++} = 0, \quad \sum_{c,d} n_{abcd} + f_{abcd} = n_{ab++}, \tag{9}$$

$$\sum_{a,d} \sum_{b,c} f_{abcd} = \sum_{b,c} f_{+bc+} = 0, \quad \sum_{a,d} n_{abcd} + f_{abcd} = n_{+bc+}, \tag{10}$$

$$\sum_{a,b}\sum_{c,d} f_{abcd} = \sum_{c,d} f_{++cd} = 0, \quad \sum_{a,b} n_{abcd} + f_{abcd} = n_{++cd}, \tag{11}$$

$$M_4: \sum_{c,d}\sum_{a,b} f_{abcd} = \sum_{a,b} f_{ab++} = 0, \quad \sum_{c,d} n_{abcd} + f_{abcd} = n_{ab++}, \tag{12}$$

$$\sum_{a,d}\sum_{b,c} f_{abcd} = \sum_{b,c} f_{+bc+} = 0, \quad \sum_{a,d} n_{abcd} + f_{abcd} = n_{+bc+}, \tag{13}$$

$$\sum_{a,c}\sum_{b,d} f_{abcd} = \sum_{b,d} f_{+b+d} = 0, \quad \sum_{a,c} n_{abcd} + f_{abcd} = n_{+b+d}. \tag{14}$$

Definition 2 A Markov basis $\mathcal{M}$ is a finite collection of moves that preserve the fixed marginals of the sufficient statistics of model M. For any two tables $\mathbf{n}, \mathbf{n}' \in T(M)$, there exists a sequence of moves $\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(R)} \in \mathcal{M}$ such that

$$\mathbf{n}' - \mathbf{n} = \sum_{r=1}^{R} \mathbf{f}^{(r)} \qquad \text{and} \qquad \mathbf{n} + \sum_{r=1}^{R'} \mathbf{f}^{(r)} \in T(M),$$

for $1 \leq R' \leq R$.

Diaconis and Sturmfels [10] showed that there exists a Markov basis for $T(M)$, which allows the construction of a Markov chain on $T(M)$.

A move $\mathbf{f} = \{f_{abcd}\}$ in which two entries are equal to 1, two entries are equal to -1 and the remaining entries are 0 is called a *primitive move*. Diaconis and Sturmfels [10] described a Markov basis that is the set of primitive moves for the independence model for a two-way contingency table. Dobra [11] also showed that the set of primitive moves is a Markov basis for the class of decomposable graphical models.

4 Algorithm for generating random tables conditionally on the fixed marginals

Consider an $I \times J \times K$ contingency table formed from variables X, Y and Z. Let $\mathbf{n} = \{n_{xyz}\}$ be the table of counts with non-negative integer cell entries. We assume that X and Z are conditionally independent given Y, i.e., $X \perp Z | Y$. The fixed marginals for the conditional independence model $[XY][YZ]$ are $\mathbf{n}_{XY} = \{n_{xy+}\}$ and $\mathbf{n}_{YZ} = \{n_{+yz}\}$.

By employing Theorem 3.1 of Diaconis and Sturmfels [10], the Markov basis for the model can be defined by the set of primitive moves with the following marginal constraints:

$$\sum_{z}\sum_{x,y} f_{xyz} = \sum_{x,y} f_{xy+} = 0, \quad \sum_{z} n_{xyz} + f_{xyz} = n_{xy+},$$

$$\sum_{x}\sum_{y,z} f_{xyz} = \sum_{x,y} f_{+yz} = 0, \quad \sum_{x} n_{xyz} + f_{xyz} = n_{+yz}.$$

The algorithm for generating a random table $\mathbf{n}' \in T([XY][YZ])$ is described below.

Algorithm: Random table generation for $[XY][YZ]$

Step 1: Select an index y_0 in $\{1, \ldots, J\}$.
Step 2: Choose indices x_1, x_2, z_1, z_2 such that $1 \leq x_1 < x_2 \leq I$ and $1 \leq z_1 < z_2 \leq K$.

Step 3: Define move $\mathbf{f} = \{f_{xyz}\}$ by

$$f_{xyz} = \begin{cases} 1, \text{ if } (x,z) \in \{(x_1,z_1),(x_2,z_2)\} \text{ with } y_0 \text{ fixed,} \\ -1, \text{ if } (x,z) \in \{(x_2,z_1),(x_1,z_2)\} \text{ with } y_0 \text{ fixed,} \\ 0, \text{ otherwise.} \end{cases}$$

Step 4: Generate $\mathbf{n}'$ from

$$\mathbf{n}' = \mathbf{n} + \varepsilon\mathbf{f},$$

where ε is 1 or -1 with probability 1/2 each.

All conditional independence structures for M_1 to M_4 can be transformed into the form $X \perp Z|Y$, as described below. Thus, Random table generation for $[XY][YZ]$ provides an essential algorithm for producing random tables in $T(M_1)$ to $T(M_4)$ and enables the generation of random tables for each model. The set $\{\mathbf{f}^{(r)}\}$ in a Markov basis for each of M_1 to M_4 is given by the set of primitive moves conditionally on the marginal constraints specified by Equations (5) to (14).

In the following subsections, we show how the Random table generation algorithm is applied to produce a move $\mathbf{f}$ in a Markov basis for each of M_1 to M_4. In Appendix, we describe the procedure for selecting indices in Steps 1 and 2 of the Random table generation algorithm.

4.1 Model $M_1 : [ABC][BCD]$

We identify pair (B,C) with new variable Y that has the categories defined by all possible combinations of values of B and C, and set $X = A$ and $Z = D$. By denoting M_1 as $[XY][YZ]$, we can generate a random table in $T(M_1)$ by applying the Random table generation algorithm.

4.2 Model M_2: $[AB][BCD]$

Similar to model M_1 above, we define new variable Y with KL combinations of categories of C and D. We denote M_2 as $[XY][YZ]$ by setting $X = A$ and $Y = B$ and apply the Random table generation algorithm to produce a random table in $T(M_2)$.

4.3 Model M_3: $[AB][BC][CD]$

As indicated in Table 2, there exist two Markov properties associated with M_3: (i) $M_{31} : A \perp (C,D)|B$ and (ii) $M_{32} : (A,B) \perp D|C$. We set

$$(X,Y,Z) = \begin{cases} (A,B,(C,D)) \text{ for } M_{31}, \\ ((A,B),C,D) \text{ for } M_{32}. \end{cases}$$

Then, we uniformly generate a move with probability proportional to the number of moves associated with M_{31} and M_{32}. Weights w_{31} and w_{32} represent the number of moves for M_{31} and M_{32}, respectively, and are calculated by

$$w_{31} = \left\{2 \times \binom{I}{2} \times \binom{KL}{2} \times J\right\}, \quad w_{32} = \left\{2 \times \binom{IJ}{2} \times \binom{L}{2} \times K\right\}.$$

Thus, we have $p_{31} = w_{31}/(w_{31} + w_{32})$ and $p_{32} = 1 - p_{31}$.

We select M_{31} or M_{32} with probability $\mathbf{p}_3 = (p_{31}, p_{32})$ and apply the Random table generation algorithm to generate a random table in $T(M_3)$.

4.4 Model M_4: $[AB][BC][BD]$

For M_4, we have three Markov properties: (i) $M_{41} : A \perp C|(B, D)$, (ii) $M_{42} : A \perp D|(B, C)$ and (iii) $M_{43} : C \perp D|(A, B)$. We set

$$(X, Y, Z) = \begin{cases} (A, (B, D), C) \text{ for } M_{41}, \\ (A, (B, C), D) \text{ for } M_{42}, \\ (C, (A, B), D) \text{ for } M_{43}. \end{cases}$$

To generate a move uniformly, we select one model from M_{41}, M_{42} and M_{43} with probability $\mathbf{p}_4 = (p_{41}, p_{42}, p_{43})$. Weights for these models are calculated by

$$w_{41} = \left\{2 \times \binom{I}{2} \times \binom{K}{2} \times JL\right\}, \quad w_{42} = \left\{2 \times \binom{I}{2} \times \binom{L}{2} \times JK\right\},$$
$$w_{43} = \left\{2 \times \binom{K}{2} \times \binom{L}{2} \times IJ\right\}.$$

Thus, we have $p_{41} = w_{41}/(w_{41} + w_{42} + w_{43})$, $p_{42} = w_{42}/(w_{41} + w_{42} + w_{43})$ and $p_{43} = 1 - (p_{41} + p_{42})$.

We generate a move in a Markov basis for the model chosen by probability $\mathbf{p}_4$ and obtain a random table in $T(M_4)$ by using the Random table generation algorithm.

5 MCMC for the computation of p-values

In the two sections below, we present the Metropolis-Hastings (M-H) algorithm of Hastings [16] for generating Markov chains, followed by our approach to computing the corresponding p-values.

5.1 Metropolis-Hastings algorithm for generating a Markov chain

Diaconis and Sturmfels [10] showed the M-H algorithm for generating a Markov chain using a Markov basis and then proved that the Markov chain is an irreducible, aperiodic Markov chain with stationary distribution π.

The M-H algorithm for generating random tables of a Markov chain on $T([XY][YZ])$ is presented below.

Algorithm: M-H algorithm for $[XY][YZ]$

Step 1: Initialize the iteration counter $r = 1$ and set $\mathbf{n}^{(0)}$ as the initial contingency table.

Step 2: Generate candidate table $\mathbf{n}' = \mathbf{n}^{(r-1)} + \varepsilon\mathbf{f}$ by using the Random table generation algorithm.

Step 3: If all cell entries of $\mathbf{n}'$ are non-negative integers, accept $\mathbf{n}'$ as the next table $\mathbf{n}^{(r)}$ with probability

$$\alpha(\mathbf{n}', \mathbf{n}^{(r-1)}) = \min\left\{ \frac{\pi(\mathbf{n}')}{\pi(\mathbf{n}^{(r-1)})}, 1 \right\}, \tag{15}$$

otherwise retain $\mathbf{n}^{(r-1)}$ and $\mathbf{n}^{(r)} = \mathbf{n}^{(r-1)}$.

Step 4: Increment counter r and return to Step 2.

We apply the M-H algorithm presented above to generate $\{\mathbf{n}^{(r)}\}$ of a Markov chain on each of $T(M_1)$ to $T(M_4)$.

When $\mathbf{n}'$ is drawn from hypergeometric distributions (1) to (4), the ratio $\pi(\mathbf{n}')/\pi(\mathbf{n}^{(r-1)})$ of Equation (15) involves only four cell counts associated with indices y_0, x_1, x_2, z_1, z_2 selected in Steps 1 and 2 of the Random table generation algorithm. For example, this algorithm generates table $\mathbf{n}' \in T(M_1)$ in which the four cell counts in $\mathbf{n}^{(r-1)}$ are modified as

$$\begin{aligned} n'_{x_1 y_0 z_1} &= n'_{a_1 b_0 c_0 d_1} = n^{(r-1)}_{a_1 b_0 c_0 d_1} + \varepsilon, \quad & n'_{x_2 y_0 z_1} &= n'_{a_2 b_0 c_0 d_1} = n^{(r-1)}_{a_2 b_0 c_0 d_1} - \varepsilon, \\ n'_{x_1 y_0 z_2} &= n'_{a_1 b_0 c_0 d_2} = n^{(r-1)}_{a_1 b_0 c_0 d_2} - \varepsilon, \quad & n'_{x_2 y_0 z_2} &= n'_{a_2 b_0 c_0 d_2} = n^{(r-1)}_{a_2 b_0 c_0 d_2} + \varepsilon. \end{aligned}$$

Then, the ratio $\pi(\mathbf{n}')/\pi(\mathbf{n}^{(r-1)})$ is given by

$$\begin{aligned} \frac{\pi(\mathbf{n}')}{\pi(\mathbf{n}^{(r-1)})} &= \frac{n^{(r-1)}_{x_1 y_0 z_1}!\, n^{(r-1)}_{x_2 y_0 z_1}!\, n^{(r-1)}_{x_1 y_0 z_2}!\, n^{(r-1)}_{x_2 y_0 z_2}!}{n'_{x_1 y_0 z_1}!\, n'_{x_2 y_0 z_1}!\, n'_{x_1 y_0 z_2}!\, n'_{x_2 y_0 z_2}!} \\ &= \frac{n^{(r-1)}_{a_1 b_0 c_0 d_1}!\, n^{(r-1)}_{a_2 b_0 c_0 d_1}!\, n^{(r-1)}_{a_1 b_0 c_0 d_2}!\, n^{(r-1)}_{a_2 b_0 c_0 d_2}!}{n'_{a_1 b_0 c_0 d_1}!\, n'_{a_2 b_0 c_0 d_1}!\, n'_{a_1 b_0 c_0 d_2}!\, n'_{a_2 b_0 c_0 d_2}!} \\ &= \begin{cases} \dfrac{n^{(r-1)}_{a_2 b_0 c_0 d_1} n^{(r-1)}_{a_1 b_0 c_0 d_2}}{(n^{(r-1)}_{a_1 b_0 c_0 d_1} + 1)(n^{(r-1)}_{a_2 b_0 c_0 d_2} + 1)}, \text{ when } \varepsilon = 1, \\ \dfrac{n^{(r-1)}_{a_1 b_0 c_0 d_1} n^{(r-1)}_{a_2 b_0 c_0 d_2}}{(n^{(r-1)}_{a_2 b_0 c_0 d_1} + 1)(n^{(r-1)}_{a_1 b_0 c_0 d_2} + 1)}, \text{ when } \varepsilon = -1. \end{cases} \end{aligned}$$

5.2 Computation of p-values

As the measure of the goodness of fit of model M, we use the Pearson chi-squared statistic

$$\chi^2(\mathbf{n}, \mathbf{m}) = \sum_{a,b,c,d} \frac{(n_{abcd} - m_{abcd})^2}{m_{abcd}},$$

where $\mathbf{m} = \{m_{abcd}\}$ is the set of fitted values under M. It follows that the p-value for M can be computed from

$$p = \sum_{\{\mathbf{n}' \in T(M)\}} I\left\{\chi^2(\mathbf{n}', \mathbf{m}) \geq \chi^2(\mathbf{n}, \mathbf{m})\right\} \pi_M(\mathbf{n}'), \tag{16}$$

where $I\{\cdot\}$ denotes an indicator function. However, the computation of the exact p-value in Equation (16) is not feasible when complete enumeration of contingency tables in $T(M)$ is very large or more complicated. The MCMC method is applicable to circumvent this enumeration problem.

Let $\{\mathbf{n}^{(r)}\}_{0\le r\le (S+R)}$ be random tables in $T(M)$ generated by the M-H algorithm described above. After discarding the first S tables as burn-in, the MCMC p-value can be obtained by

$$\hat{p} = \frac{1}{R}\sum_{r=S+1}^{S+R} I\left\{\chi^2(\mathbf{n}^{(r)},\mathbf{m}) \ge \chi^2(\mathbf{n},\mathbf{m})\right\}. \tag{17}$$

6 Numerical experiments

We provide two numerical examples: the first is that exact results obtained by complete enumeration are available for comparison, and the second is that MCMC is the only feasible method for exact inference. The computation is performed by using the `R` language.

Example 1 Table 3 is a study of nonmetastatic osteosarcoma by Goorin *et al.* [15]. The response shown in the table indicates whether the subject achieved a three-year disease-free interval.

For M_1 and M_2, the exact p-values are calculated from complete enumeration of $T(M_1)$ and $T(M_2)$ using `MIMWIN` of Edwards [12]. When exact p-values are unknown, we use the Monte Carlo estimation method from the importance sampling (IS) algorithm of Booth and Butler [3] instead of exact computations. The Monte Carlo p-values for M_3 and M_4 are obtained from 1,000,000 random tables by using the `R` package `exactLoglinTest` of Caffo [5], an implementation of the IS algorithm. We compare the p-values from the χ^2 approximation and the proposed MCMC approach with the exact or Monte Carlo p-values.

The MCMC method described in Section 4 generates 1,000,000 random tables with $10,000$ tables as burn-in. To evaluate the accuracy of a MCMC p-value, we compute its standard error using the batch means method of Geyer [14]. When dividing random tables $\{\mathbf{n}^{(r)}\}_{S+1\le r\le S+R_1R_2}$ into R_1 batches each of size R_2, the MCMC p-value for the k-th batch is calculated as

$$\hat{p}_k = \frac{1}{R_2}\sum_{r=S+(k-1)R_1+1}^{S+kR_1} I\left\{\chi^2(\mathbf{n}^{(r)},\mathbf{m}) \ge \chi^2(\mathbf{n},\mathbf{m})\right\}.$$

The variance estimate of $\hat{p}$ is obtained from

$$\mathrm{Var} = \frac{1}{R_1-1}\sum_{k=1}^{R_1}(\hat{p}_k - \hat{p})^2$$

and the batch means estimate of the MCMC standard error is calculated from $\sqrt{\mathrm{Var}/R_1}$. In our experiments, we set $R_1 = 100$ and $R_2 = 10,000$. The values in parenthesis are the standard errors of the MCMC p-values.

Results are shown in Table 4. As shown in the table, MCMC p-values are in good agreement with the exact and Monte Carlo p-values, while the asymptotic p-values using the chi-squared distribution differ greatly. Figure 2 shows histograms of values

$\{\chi^2(\mathbf{n}^{(r)}, \mathbf{m})\}$ of the Pearson chi-squared statistics for M_1 to M_4. For models other than M_3, we see substantial discrepancies between the asymptotic and MCMC estimated exact distributions. They indicate that the asymptotic χ^2 approximation is unreliable for sparse data.

Example 2 Table 5 shows abortion opinion data from Christensen [6]. Observations are classified by four factors: Race (A), Sex (B), Opinion (C) and Age (D). For C, three different opinions are possible: a "Yes" answer supports legalized abortion; a "No" answer opposes legalized abortion; an "Undecided" answer is undecided. Because the contingency table contains many small expected values and the sample size is large, the computation of exact p-values obtained by enumeration is not feasible.

Table 6 presents the asymptotic and MCMC p-values for M_1 to M_4. The MCMC p-values for each model are obtained from 1,000,000 simulated tables after a burn-in of 10,000 tables. Standard error values of p are shown in parentheses. The small MCMC p-value for M_4 may lack accuracy, because a MCMC estimate from 1,000,000 tables guarantees at most three-digit precision.

7 Concluding remarks

In this paper, we proposed the use of MCMC with a Markov basis for estimating exact p-values for the conditional independence models for a four-way contingency table. A Markov basis is required to generate random tables with fixed marginals specified by a conditional independence model. We defined the Markov basis characterized by the Markov properties associated with a given conditional independence model. Then, we developed an algorithm for generating primitive moves in the Markov basis and presented a MCMC method for producing a Markov chain by using the Markov basis.

Results from our first experiment demonstrated that the proposed MCMC method finds the estimates of exact p-values in close agreement with exact p-values. Our second experiment illustrated that the MCMC method is feasible for exact inference when the asymptotic χ^2 approximation is unreliable and the exact computation of p-values by complete enumeration is not feasible.

In the future, we intend to develop a MCMC method for decomposable graphical models in multi-way contingency tables by extending our MCMC method with a Markov basis characterized by Markov properties.

Acknowledgements The authors would like to thank the editor and a referee for their valuable comments and helpful suggestions.

Appendix

We describe the procedure for selecting indices in Steps 1 and 2 of the Random table generation algorithm for each model.

Algorithm: Random table generation for M_1

Step 1: Select $y_0 = (b_0, c_0)$ in $\{1, \ldots, J\} \times \{1, \ldots, K\}$.
Step 2: Choose x_1, x_2 in $\{1, \ldots, I\}$ and z_1,z_2 in $\{1, \ldots, L\}$.

Algorithm: Random table generation for M_2

Step 1: Select y_0 in $\{1,\ldots,J\}$.
Step 2: Choose x_1, x_2 in $\{1,\ldots,I\}$ and $z_1=(c_1,d_1)$, $z_2=(c_2,d_2)$ in $\{1,\ldots,K\}\times\{1,\ldots,L\}$.

Algorithm: Random table generation for M_3

Determine M_{31} or M_{32} with probability $\mathbf{p}_3$. If M_{31} is selected, set $(X,Y,Z)=(A,B,(C,D))$, else $(X,Y,Z)=((A,B),C,D)$.

Step 1: Select y_0:
- for M_{31}, y_0 in $\{1,\ldots,J\}$,
- for M_{32}, y_0 in $\{1,\ldots,K\}$.

Step 2: Choose x_1, x_2, z_1, z_2:
- for M_{31}, x_1, x_2 in $\{1,\ldots,I\}$ and $z_1=(c_1,d_1)$, $z_2=(c_2,d_2)$ in $\{1,\ldots,K\}\times\{1,\ldots,L\}$,
- for M_{32}, $x_1=(a_1,b_1)$, $x_2=(a_2,b_2)$ in $\{1,\ldots,I\}\times\{1,\ldots,J\}$ and z_1, z_2 in $\{1,\ldots,L\}$.

Algorithm: Random table generation for M_4

Select one model among M_{41}, M_{42} and M_{43} with probability $\mathbf{p}_4$. Set

$$(X,Y,Z)=\begin{cases}(A,(B,D),C), & \text{when choosing } M_{41},\\ (A,(B,C),D), & \text{when choosing } M_{42},\\ (C,(A,B),D), & \text{when choosing } M_{43}.\end{cases}$$

Step 1: Select y_0:
- for M_{41}, $y_0=(b_0,d_0)$ in $\{1,\ldots,J\}\times\{1,\ldots,L\}$,
- for M_{42}, $y_0=(b_0,c_0)$ in $\{1,\ldots,J\}\times\{1,\ldots,K\}$,
- for M_{43}, $y_0=(a_0,b_0)$ in $\{1,\ldots,I\}\times\{1,\ldots,J\}$.

Step 2: Choose x_1, x_2, z_1, z_2:
- for M_{41}, x_1, x_2 in $\{1,\ldots,I\}$ and z_1, z_2 in $\{1,\ldots,K\}$,
- for M_{42}, x_1, x_2 in $\{1,\ldots,I\}$ and z_1, z_2 in $\{1,\ldots,L\}$,
- for M_{43}, x_1, x_2 in $\{1,\ldots,K\}$ and z_1, z_2 in $\{1,\ldots,L\}$.

References

1. Agresti A (1992). A survey of exact inference for contingency tables. Statist. Sci. 7: 131-153.
2. Agresti A (1999). Exact inference for categorical data: recent advances and continuing controversies. Stat. Med. 20: 2709-2722.
3. Booth JG, Butler RW (1999). An importance sampling algorithm for exact conditional test in log-linear models. Biometrika 86: 321-332.
4. Caffo BS, Booth JG (2001). A Markov chain Monte Carlo algorithm for approximating exact conditional probabilities. J. Comput. Graph. Statist. 10: 730-745.
5. Caffo BS (2006). Exact hypothesis tests for log-linear models with exactLoglinTest. The Journal of Statistical Software 17: 1-17.
6. Christensen R (1997). Log-linear models and logistic regression 2nd edition. Springer-Verlag, New York.

7. Dalenius P, Reiss RS (1982). Data-swapping: A technique for disclosure control. J. Statist. Plann. Inference 6: 73-85.
8. Darroch JN, Lauritzen SL, Speed TP (1980). Markov-fields and log-linear models for contingency tables. Ann. Statist. 8: 522-539.
9. Dawid AP (1979). Conditional independence in statistical theory (with discussion). J. Roy. Statist. Soc. Ser. B 41: 1–31.
10. Diaconis P, Sturmfels B (1998). Algebraic algorithms for sampling from conditional distributions. Ann. Statist. 26: 363-397.
11. Dobra A (2003). Markov bases for decomposable graphical models. Bernoulli 9: 1093-1108.
12. Edwards D (1995). Introduction to graphical modelling. Springer-Verlag, New York.
13. Forster JJ, McDonald JW, Smith PW (1996). Monte Carlo exact conditional tests for log-linear and logistic models. J. Roy. Statist. Soc. Ser. B 58: 445–453.
14. Geyer CJ (1992). Practical Markov Chain Monte Carlo. Statist. Sci. 7: 473-483.
15. Goorin AM, Perez-Atayde A, Gebhardt M, Andersen JW, Wilkinson RH, Delorey MJ, Watts H, Link M, Jaffe N, Frei 3dE. (1987). Weekly high-dose methotrexate and doxorubicin for osteosarcoma: the Dana-Farber Cancer Institute/the Children's Hospital–study III. J. Clin. Oncol. 5: 1178-1184.
16. Hastings WK (1970). Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57: 97-109.
17. Lauritzen S L (1996). Graphical models. Oxford University Press, New York.
18. Sundberg R (1975). Some results about decomposable (or Markov-type) models for multidimensional contingency tables: distribution of marginals and partitioning of tests. Scand. J. Statist. 2: 71-79.

Table 1 Conditional independence models for a four-way contingency table

Model	Interpretation
M_1: $[ABC][BCD]$	A is conditional independent of D given (B, C)
M_2: $[AB][BCD]$	A is conditional independent of (C, D) given B
M_3: $[AB][BC][CD]$	A is conditional independent of (C, D) given B, and (A, B) is conditional independent of D given C
M_4: $[AB][BC][BD]$	A, C and D are all conditional independent given B

Table 2 Markov properties for the conditional independence models

Model	Markov property
M_1: $[ABC][BCD]$	$A \perp D \| (B, C)$
M_2: $[AB][BCD]$	$A \perp (C, D) \| B$
M_3: $[AB][BC][CD]$	$A \perp (C, D) \| B$ and $(A, B) \perp D \| C$
M_4: $[AB][BC][BD]$	$A \perp C \| (B, D)$, $A \perp D \| (B, C)$ and $C \perp D \| (A, B)$

Table 3 Study of nonmetastatic osteosarcoma from Goorin *et al.* [15]

	Lymphocytic	Osteoblastic	Disease-free(D)	
Sex(A)	Infiltration(B)	Pathology(C)	Yes	No
Female	High	No	3	0
		Yes	4	0
	Low	No	5	0
		Yes	5	4
Male	High	No	2	0
		Yes	1	0
	Low	No	3	2
		Yes	6	11

Table 4 Asymptotic p-values versus estimated exact p-values

			p-value		
Model	df	χ^2 value	asymptotic	MCMC	exact
$M_1 : [ABC][BCD]$	4	3.471	0.4821	0.2066 (0.0013)	0.2061
$M_2 : [AB][BCD]$	6	4.748	0.5765	0.3649 (0.0023)	0.3674
$M_3 : [AB][BC][CD]$	8	12.250	0.1404	0.1291 (0.0017)	0.1250*
$M_4 : [AB][BC][BD]$	8	9.267	0.3202	0.1036 (0.0018)	0.1017*

$*$ indicates a Monte Carlo p-value from the IS algorithm of Booth and Butler [3].

Table 5 Abortion opinion data from Christensen [6]

			Age(D)					
Race(A)	Sex(B)	Opinion(C)	18-25	26-35	36-45	46-55	56-65	66+
White	Male	Yes	96	138	117	75	72	83
		No	44	64	56	48	49	60
		Und	1	2	6	5	6	8
	Female	Yes	140	171	152	101	102	111
		No	43	65	58	51	58	67
		Und	1	4	9	9	10	16
Nonwhite	Male	Yes	24	18	16	12	6	4
		No	5	7	7	6	8	10
		Und	2	1	3	4	3	4
	Female	Yes	21	25	20	17	14	13
		No	4	6	5	5	5	5
		Und	1	2	1	1	1	1

Table 6 Goodness-of-fit statistics and MCMC p-values

			p-value	
Model	df	χ^2 value	asymptotic	MCMC
$M_1 : [ABC][BCD]$	30	23.10	0.8112	0.8370 (0.0043)
$M_2 : [AB][BCD]$	34	54.67	0.0138	0.0195 (0.0015)
$M_3 : [AB][BC][CD]$	49	59.62	0.1422	0.1505 (0.0068)
$M_4 : [AB][BC][BD]$	54	114.39	3.156×10^{-6}	9.6×10^{-5} (7.7×10^{-5})

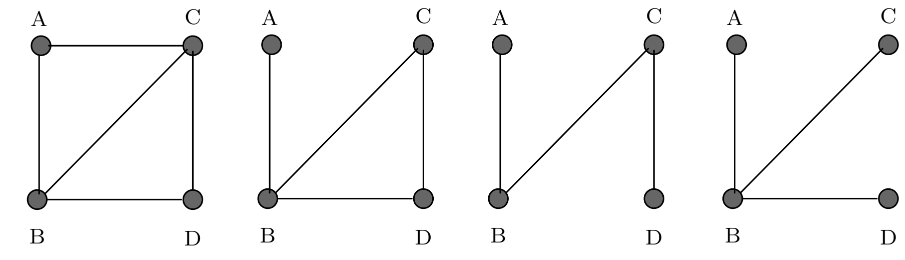

Fig. 1 Graphical representation for the conditional independence models in Table 1

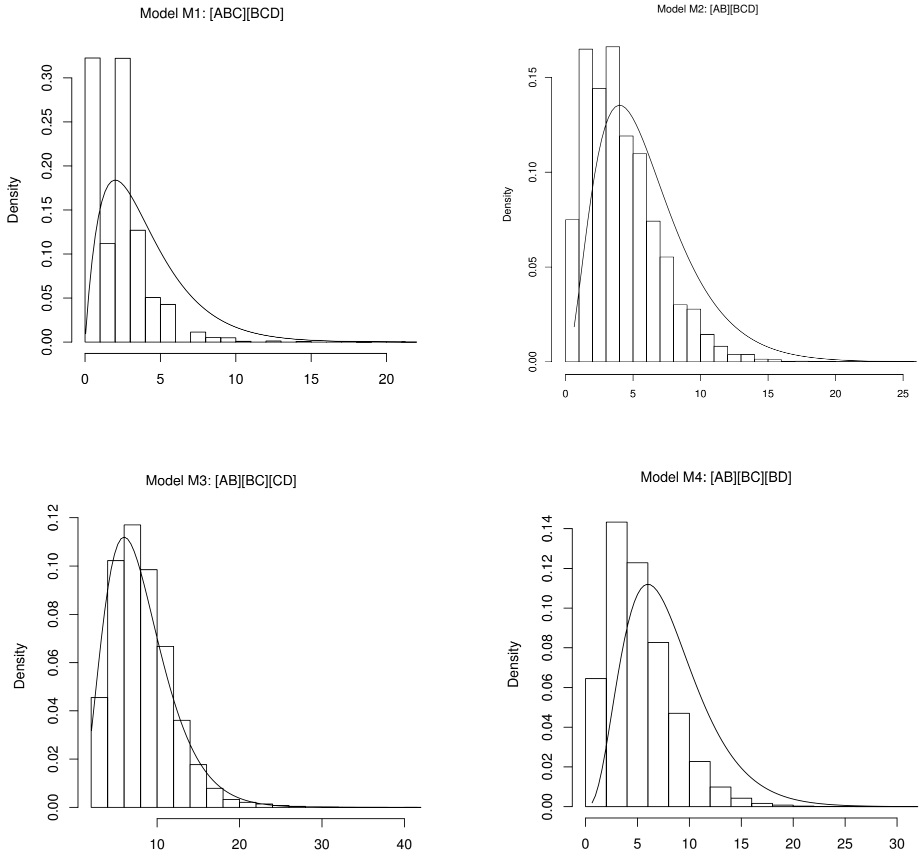

Fig. 2 Histograms drawn by MCMC samples of the chi-squared statistics for M_1 to M_4