MATHEMATICAL ENGINEERING TECHNICAL REPORTS

Polynomial Time Perfect Sampling Algorithm for Two-rowed Contingency Tables

Shuji KIJIMA and Tomomi MATSUI

METR 2003–15 April 2003

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm

Polynomial Time Perfect Sampling Algorithm for Two-rowed Contingency Tables *

Shuji Kijima[1] and Tomomi Matsui[2]
Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan

[1] `kijima@misojiro.t.u-tokyo.ac.jp`
[2] `http://www.simplex.t.u-tokyo.ac.jp/~tomomi/`

April 10, 2003

Abstract

This paper proposes a polynomial time perfect (exact) sampling algorithm for $2 \times n$ contingency tables. Our algorithm is a Las Vegas type randomized algorithm and the expected running time is bounded by $\mathrm{O}(n^3 \ln N)$ where n is the number of columns and N is the total sum of whole entries in a table. The algorithm is based on monotone coupling from the past (monotone CFTP) algorithm and new Markov chain for sampling two-rowed contingency tables uniformly. We employed the path coupling method and showed that the mixing rate of our chain is bounded by $n(n-1)^2(1+\ln(nN))$. Our result indicates that uniform generation of two-rowed contingency tables is easier than the corresponding counting problem, since the counting problem is known to be #P–complete.

1 Introduction

In this paper, we propose a polynomial time perfect (exact) sampling algorithm for $2 \times n$ contingency tables. Our algorithm is a Las Vegas type randomized algorithm and the expected running time is bounded by $\mathrm{O}(n^3 \ln N)$ where n is the number of columns and N is the total sum of whole entries in a table. Our result indicates that uniform generation of two-rowed contingency tables is easier than the corresponding counting problem, since the counting problem is known to be #P–complete [11]. The main idea of our algorithm is not the simple rejection sampling but the monotone coupling from the past (monotone CFTP) algorithm proposed by Propp and Wilson [18].

For sampling two-rowed contingency tables, Dyer and Greenhill proposed a fully polynomial time approximately uniform sampler in [10] based on the Metropolis-Hastings algorithm with a natural Markov chain. They showed that the mixing rate of their chain is bounded by $(1/2)n(n-1)(1+\ln N)$ by using the path coupling technique proposed by Bubley and Dyer [5]. We propose a new Markov chain which is obtained by forbidding some moves of Dyer and Greenhill's chain. The mixing rate of our chain is bounded by $n(n-1)^2(1+\ln(nN))$. Although we have shown the bound by using path coupling method also, we need to introduce a preprocedure which is not appeared in Dyer and Greenhill's method. We also introduced a specified partial order on the set of tables and showed the monotonicity of our chain. In the paper [18], Propp and Wilson showed that if we have a monotone

*Supported by Superrobust Computation Project of the 21st Century COE Program "Information Science and Technology Strategic Core."

chain with polynomial time mixing rate, there exists a polynomial time monotone CFTP algorithm. By applying their technique to our rapidly mixing monotone chain we can construct a polynomial time perfect sampling algorithm.

A contingency table is a matrix of nonnegative integers with prescribed positive row and column sums. Contingency tables are used in statistics to store data from sample survey. A test of independence between rows and columns is an statistic interest for contingency tables. Exact test proposed by Fischer [12] is one of the tests for this purpose. Diaconis and Effron also discussed a test for independence in contingency tables [7]. Exact test can be done by systematic enumeration of all tables, but it is hard to enumerate all tables. In practice, Markov chain Monte Carlo (MCMC) method is used for calculation of p value in exact test (see [1, 2] for example). A MCMC method is a Monte Carlo method which uses samples from the stationary distribution of a Markov chain. The problem of using MCMC method, however, is "how many times do we have to simulate the transitions for the purpose of sampling from stationary distribution?" A practical solution for this problem is to use approximate sampler obtained by interrupting transitions in finite time.

There are many works for almost uniform sampling contingency tables using a Markov chain. Diaconis and Saloff-Coste [8] discussed the rate of convergence of a simple Markov chain for 2-dimensional contingency tables. They have shown that the simple chain mixes polynomial time in the table sum when the numbers of rows and columns are fixed. Dyer, Kannan and Mount [11] proposed a different Markov chain for counting the number of 2-dimensional contingency tables. In case of sufficiently large marginal sums, their chain mixes polynomial time in the number of rows and columns. For two-rowed tables, Hernek [14] showed that the mixing time of the simple Markov chain is bounded by a polynomial of table sum and number of columns. Hernek bounded the mixing time of the chain by using coupling theorem shown by Aldous [3]. Dyer and Greenhill [10] proposed a rapidly mixing Markov chain for two-rowed contingency tables. Their chain mixes polynomial time in the number of columns and the logarithm of table sum. They analyzed the mixing rate of their chain by using path coupling technique proposed by Bubley and Dyer [4, 5]. In the paper [17], Matsui, Matsui and Ono extended Dyer and Greenhill's result to $2 \times \cdots \times 2 \times J$ contingency tables. Recently, Cryan, Dyer, Goldberg, Jerrum and Martin [6] showed that 2×2 chain, which is an extension of Dyer and Greenhill's, is rapidly mixing when the number of rows (or columns) is a constant.

Propp and Wilson devised a surprising simple algorithm, called CFTP algorithm (or backward coupling), which produces exact samples from the limit distribution [18, 19]. CFTP algorithm simulate infinite time transitions of a chain in a (probabilistically) finite time, for any finite Markov chain. In CFTP algorithm, however, we need to check the "coalescence condition", by executing the simulations from all the states. Thus CFTP algorithm is not available straightforwardly. A monotone CFTP algorithm is an algorithm for monotone Markov chain, which has a partially ordered state space and a transition rule which preserves the partial order. If the given chain has the monotonicity, it relaxes the difficulty of simulation from all states.

In the next section, we review the (monotone) CFTP algorithm and the theorem proposed by Propp and Wilson in [18]. We propose a new Markov chain for $2 \times n$ contingency tables and a sampling algorithm based on monotone CFTP algorithm in Section 3. In Section 4, we show the monotonicity of our chain. In Section 5, we analyze the expected running time.

2 Review of Coupling From The Past Algorithm

When we simulate an ergodic Markov chain for infinite time, we can gain a sample exactly according to the stationary distribution. Suppose that there exists a chain from infinite past, then a possible state at the present time of the chain for which we can have an evidence of the uniqueness without respect to an initial state of the chain, is a realization of a random sample exactly from the stationary distribution. This is the key idea of CFTP.

Suppose that we have an ergodic Markov chain MC with finite state space Ω and transition matrix P. The transition rule of the Markov chain $X \mapsto X'$ can be described by a deterministic function $\phi : \Omega \times [0,1) \to \Omega$, called *update function*, as follows. Given a random number Λ uniformly distributed over $[0,1)$, update function ϕ satisfies that $\Pr(\phi(x,\Lambda) = y) = P(x,y)$ for any $x, y \in \Omega$. We can realize the Markov chain by setting $X' = \phi(X, \Lambda)$. Clearly, update function corresponding to the given transition matrix P is not unique. The result of transitions of the chain from the time t_1 to t_2 ($t_1 < t_2$) with a sequence of random numbers $\boldsymbol{\lambda} = (\lambda[t_1], \lambda[t_1{+}1], \ldots, \lambda[t_2{-}1]) \in [0,1)^{t_2-t_1}$ is denoted by $\Phi_{t_1}^{t_2}(x, \boldsymbol{\lambda}) : \Omega \times [0,1)^{t_2-t_1} \to \Omega$ where $\Phi_{t_1}^{t_2}(x,\boldsymbol{\lambda}) \stackrel{\text{def.}}{=} \phi(\phi(\cdots(\phi(x,\lambda[t_1]), \ldots, \lambda[t_2-2]), \lambda[t_2-1])$. We say that a sequence $\lambda \in [0,1)^{|T|}$ satisfies the *coalescence condition*, when $\exists y \in \Omega$, $\forall x \in \Omega$, $y = \Phi_T^0(x, \boldsymbol{\lambda})$.

With these preparation, standard Coupling From The Past algorithm is expressed as follows.

Algorithm 1 (CFTP Algorithm [18])

Step 1. Set the starting time period $T := -1$ to go back, and set $\boldsymbol{\lambda}$ be the empty sequence.

Step 2. Generate random real numbers $\lambda[T], \lambda[T+1], \ldots, \lambda[\lceil T/2 \rceil - 1] \in [0,1)$, and insert them to the head of $\boldsymbol{\lambda}$ in order, i.e., put $\boldsymbol{\lambda} := (\lambda[T], \lambda[T+1], \ldots, \lambda[-1])$.

Step 3. Start a chain from each element $x \in \Omega$ at time period T, and run each chain to time period 0 according to the update function ϕ with the sequence of numbers in $\boldsymbol{\lambda}$. (Here we note that every chain uses the common sequence λ.)

Step 4. [Coalescence check] The state obtained at time period 0 can be denoted by $\Phi_T^0(x, \boldsymbol{\lambda})$.

(a) If $\exists y \in \Omega$, $\forall x \in \Omega$, $y = \Phi_T^0(x, \boldsymbol{\lambda})$, then return y and stop.

(b) Else, update the starting time period $T := 2T$, and go to Step 2.

Theorem 2.1 (CFTP Theorem [18]) *Let MC be an ergodic finite Markov chain with state space Ω, defined by an update function $\phi : \Omega \times [0,1) \to \Omega$. If the CFTP algorithm (Algorithm 1) terminates with probability 1, then the obtained value is a realization of a random variable exactly distributed according to the stationary distribution.*

Theorem 2.1 gives a (probabilistically) finite time algorithm for infinite time simulation. However, simulations from all states executed in Step 3 is a hard requirement.

Suppose that there exists a partial order "$\succeq$" on the set of states Ω. A transition rule expressed by a deterministic update function ϕ is called *monotone* (with respect to "$\succeq$") if $\forall \lambda \in [0,1)$, $\forall x, \forall y \in \Omega$, $x \succeq y \Rightarrow \phi(x,\lambda) \succeq \phi(y,\lambda)$. For ease, we also say that a chain is *monotone* if the chain has a *monotone* transition rule.

Theorem 2.2 (monotone CFTP [18, 9]) *Suppose that a Markov chain defined by an update function ϕ is monotone with respect to a partially ordered set of states $(\Omega, \succeq)$, and $\exists x_{\max}, \exists x_{\min} \in \Omega$, $\forall x \in \Omega$, $x_{\max} \succeq x \succeq x_{\min}$. Then the CFTP algorithm (Algorithm 1) terminates with probability 1, and a sequence $\lambda \in [0,1)^{|T|}$ satisfies the coalescence condition, i.e., $\exists y \in \Omega$, $\forall x \in \Omega$, $y = \Phi_T^0(x, \boldsymbol{\lambda})$, if and only if $\Phi_T^0(x_{\max}, \boldsymbol{\lambda}) = \Phi_T^0(x_{\min}, \boldsymbol{\lambda})$.*

When the given Markov chain satisfies the conditions of Theorem 2.2, we can modify Algorithm 1 by substituting Step 4 (a) by

Step 4. (a)′ If $\exists y \in \Omega$, $y = \Phi_T^0(x_{\max}, \boldsymbol{\lambda}) = \Phi_T^0(x_{\min}, \boldsymbol{\lambda})$, then return y.

The algorithm obtained by the above modification is called a monotone CFTP algorithm.

3 Perfect Sampler for $2 \times n$ Contingency Tables

In this section, we introduce our algorithm. We denote the set of real numbers by $\mathbb{R}$ and the set of integers (non-negative, positive integers) by $\mathbb{Z}$ ($\mathbb{Z}_+$, $\mathbb{Z}_{++}$), respectively. Let $\boldsymbol{r} = (r_1, r_2) \in \mathbb{Z}_{++}^2$ and $\boldsymbol{s} = (s_1, \ldots, s_n) \in \mathbb{Z}_{++}^n$ be a pair of vectors satisfying $\sum_{i=1}^2 r_i = \sum_{j=1}^n s_j = N \in \mathbb{Z}_{++}$. The set Ξ of $2 \times n$ contingency tables with row and column sums ($\boldsymbol{r}$,$\boldsymbol{s}$) is defined by

$$\Xi \stackrel{\text{def.}}{=} \left\{ X \in \mathbb{Z}_+^{2\times n} \;\middle|\; \begin{array}{l} \sum_{j=1}^n X[i,j] = r_i \ (1 \le \forall i \le 2), \\ \sum_{i=1}^2 X[i,j] = s_j \ (1 \le \forall j \le n) \end{array} \right\}$$

where $X[i,j]$ is the value in the cell indexed by ith row and jth column.

We propose a new Markov chain $\mathcal{M}$ with state space Ξ for given $\boldsymbol{r}$ and $\boldsymbol{s}$. For any column index $j \in \{1, \ldots, n-1\}$, we define

$$a_X(j) \stackrel{\text{def.}}{=} X[1,j] + X[1,j+1], \tag{1}$$

$$b_X(j) \stackrel{\text{def.}}{=} X[2,j] + X[2,j+1], \tag{2}$$

$$\theta_X(j) \stackrel{\text{def.}}{=} \min\{a_X(j), b_X(j), s_j, s_{j+1}\} + 1. \tag{3}$$

The transition rule of $\mathcal{M}$ is defined by the following *update function* $\phi : \Xi \times [1,n) \to \Xi$. For a current state $X \in \Xi$, the next state $X' = \phi(X, \lambda) \in \Xi$ with respect to a random number $\lambda \in [1,n)$ is defined by

$$\begin{aligned} X'[1,j] &= \begin{cases} \min\{a_X(j), s_j\} - \lfloor (\lambda - \lfloor \lambda \rfloor)\ \theta_X(j) \rfloor & (j = \lfloor \lambda \rfloor), \\ a_X(j) - X'[1, \lfloor \lambda \rfloor] = a_X(j) - \min\{a_X(j), s_j\} + \lfloor (\lambda - \lfloor \lambda \rfloor)\ \theta_X(j) \rfloor & (j = \lfloor \lambda \rfloor + 1), \\ X[1,j] & (\text{otherwise}), \end{cases} \\ X'[2,j] &= s_j - X'[1,j]. \end{aligned}$$

Our chain $\mathcal{M}$ is a modification of Dyer and Greenhill's chain ([10]) obtained by restricting to chose only a consecutive pair of columns. Clearly $\mathcal{M}$ is finite, aperiodic and irreducible and so ergodic. The chain has a unique stationary distribution, which is the uniform distribution.

We define two special tables X_{U} and $X_{\mathrm{L}} \in \Xi$ by

$$\begin{aligned} X_{\mathrm{U}} &\stackrel{\text{def.}}{=} \left(X[i,j] \in \mathbb{Z}_+ \;\middle|\; \begin{array}{l} \exists k \in \{1, \ldots, n\},\ r_1 = \sum_{j=1}^k X[1,j] \le \sum_{j=1}^k s_j, \\ X[2,j] = 0 \ (j = 1, \ldots, k-1) \end{array} \right), \\ X_{\mathrm{L}} &\stackrel{\text{def.}}{=} \left(X[i,j] \in \mathbb{Z}_+ \;\middle|\; \begin{array}{l} \exists l \in \{1, \ldots, n\},\ r_1 = \sum_{j=l}^n X[1,j] \le \sum_{j=l}^n s_j, \\ X[2,j] = 0 \ (j = l+1, \ldots, n) \end{array} \right). \end{aligned}$$

Here we note that $X_{\mathrm{U}}, X_{\mathrm{L}}$ are obtained by the North-West corner rule and the North-East corner rule, respectively. Now we describe our sampling algorithm.

Algorithm 2

Step 1. Set the starting time period $T := -1$ to go back, and set $\boldsymbol{\lambda}$ be the empty sequence.

Step 2. Generate random real numbers $\lambda[T], \lambda[T+1], \dots, \lambda[\lceil T/2 \rceil - 1] \in [1, n)$, and put $\boldsymbol{\lambda} := (\lambda[T], \lambda[T+1], \dots, \lambda[-1])$.

Step 3. Start two chains from X_{U} and X_{L}, respectively at time period T, and run them to time period 0 according to the update function ϕ with the sequence of numbers in $\boldsymbol{\lambda}$.

Step 4. [Coalescence check]

(a) If $\exists Y \in \Xi$, $Y = \Phi_T^0(X_{\mathrm{U}}, \boldsymbol{\lambda}) = \Phi_T^0(X_{\mathrm{L}}, \boldsymbol{\lambda})$, then return Y and stop.

(b) Else, update the starting time period $T := 2T$, and go to Step 2.

Theorem 3.1 *With probability 1, Algorithm 2 terminates and returns a table. The table obtained by Algorithm 2 is a realization of a random sample according to the exactly uniform distribution on* Ξ.

Theorem 3.1 guarantees that Algorithm 2 is a perfect sampling algorithm. We prove Theorem 3.1 by showing the monotonicity in the next section.

4 Monotonicity of the Chain

In Section 2, we described two theorems. Thus to prove Theorem 3.1, we only need to show that Algorithm 2 is a monotone CFTP algorithm. For this purpose, in this section, we introduce a partial order on Ξ, and show that X_{U} and X_{L} is a unique pair of maximum and minimum elements of Ξ, and Markov chain $\mathcal{M}$ is monotone.

For any $X \in \Xi$, we define the *cumulative sum vector* $f_X \in \mathbb{Z}_+^{n+1}$ by

$$f_X(i) \stackrel{\text{def.}}{=} \begin{cases} 0 & (i = 0), \\ X[1,1] + \cdots + X[1,i] & (i \in \{1, \dots, n\}), \end{cases}$$

where $f_X \stackrel{\text{def.}}{=} (f_X(0), f_X(1), \dots, f_X(n))$. Obviously from the definition, there exists a bijection between Ξ and $\{f_X \mid X \in \Xi\}$. For any pair $X, Y \in \Xi$, we say $X \succeq Y$ if and only if $f_X - f_Y \geq \mathbf{0}$. It is clear that the relation "$\succeq$" is a partial order on Ξ.

First, we consider the maximum and minimum states.

Lemma 4.1 *Given the partially ordered set* $(\Xi, \succeq)$*,* $X_{\mathrm{U}} \succeq X \succeq X_{\mathrm{L}}$ *for any* $X \in \Xi$.

Proof: It is clear that $\forall X \in \Xi$, $\forall i \in \{0, 1, \dots, n\}$, $0 \leq f_X(i) \leq r_1$. First, we show that $f_{X_{\mathrm{U}}}(i) \geq f_X(i)$ for any $X \in \Xi$ by induction on i. For $i = 0$, $f_{X_{\mathrm{U}}}(0) = f_X(0) = 0$. Suppose that $f_{X_{\mathrm{U}}}(i-1) \geq f_X(i-1)$. Then $f_{X_{\mathrm{U}}}(i) = f_{X_{\mathrm{U}}}(i-1) + X_{\mathrm{U}}[1,i] = \min\{f_{X_{\mathrm{U}}}(i-1) + s_i, r_1\} \geq \min\{f_X(i-1) + s_i, r_1\} \geq f_X(i)$. Thus we obtain $X_{\mathrm{U}} \succeq X$ for any $X \in \Xi$.

Next, we show that $f_X(i) \geq f_{X_{\mathrm{L}}}(i)$ for any $X \in \Xi$ by induction on i. For $i = n$, $f_X(n) = f_{X_{\mathrm{L}}}(n) = r_1$. Suppose that $f_X(i) \geq f_{X_{\mathrm{L}}}(i)$. Then $f_{X_{\mathrm{L}}}(i-1) = f_{X_{\mathrm{L}}}(i) - X_{\mathrm{L}}[1,i] = \max\{f_{X_{\mathrm{L}}}(i) - s_i, 0\} \leq \max\{f_X(i) - s_i, 0\} \leq f_X(i-1)$. We obtain $X \succeq X_{\mathrm{L}}$ for any $X \in \Xi$. $\square$

We say that a state $X \in \Xi$ *covers* $Y \in \Xi$ (at k), denoted by $X \cdot\!\!\succ Y$ (or $X \cdot\!\!\succ_k Y$), when

$$f_X(i) - f_Y(i) = \begin{cases} 1 & (i = k), \\ 0 & (\text{otherwise}). \end{cases}$$

Note that $X \cdot\!\!\succ_k Y$ if and only if

$$X[1,i] - Y[1,i] = \begin{cases} +1 & (i = k), \\ -1 & (i = k+1), \\ 0 & (\text{otherwise}). \end{cases}$$

Lemma 4.2 *If a pair of distinct states $X, Y \in \Xi$ satisfies $X \succeq Y$, then $\exists Z \in \Xi$, $X \cdot\!\succ Z \succeq Y$.*

Proof: We need to consider the following two cases.

1. Consider the case that $\exists k' \in \{0, 1, \ldots, n-1\}$, $f_X(k') > f_Y(k')$ and $X[1, k'+1] < s_{k'+1}$.
Let k be the minimum index satisfying the above condition. Clearly $k \geq 1$, since $f_X(0) = f_Y(0) = 0$. Now we show that $X[1, k] > 0$. If $f_X(k-1) > f_Y(k-1)$, the minimality of k implies that $X[1, k] = s_k > 0$. When $f_X(k-1) \leq f_Y(k-1)$, $X \succeq Y$ implies $f_X(k-1) = f_Y(k-1)$ and $X[1, k] = f_X(k) - f_X(k-1) > f_Y(k) - f_Y(k-1) = Y[1, k] \geq 0$. Then the table Z defined by
$$\begin{aligned} Z[1, l] &= \begin{cases} X[1, k] - 1 & (l = k), \\ X[1, k+1] + 1 & (l = k+1), \\ X[1, l] & (\text{otherwise}), \end{cases} \\ Z[2, l] &= s_l - Z[1, l] \quad (l = 1, \ldots, n), \end{aligned}$$
satisfies $Z \in \Xi$ and $X \cdot\!\succ Z \succeq Y$.

2. Consider the case that $\forall k' \in \{0, 1, \ldots, n-1\}$, $f_X(k') > f_Y(k') \Rightarrow X[1, k'+1] = s_{k'+1}$.
In the following, we show that $\forall k \in \{0, \ldots, n-1\}$, $X[1, k+1] \geq Y[1, k+1]$. If $f_X(k) > f_Y(k)$, the assumption of Case 2 implies that $X[1, k+1] = s_{k+1} \geq Y[1, k+1]$. When $f_X(k) \leq f_Y(k)$, $X \succeq Y$ implies that $f_X(k) = f_Y(k)$ and $f_X(k+1) \geq f_Y(k+1)$. Thus we have $X[1, k+1] = f_X(k+1) - f_X(k) \geq f_Y(k+1) - f_Y(k) = Y[1, k+1]$. Since $X[1, 1] + \cdots + X[1, n] = Y[1, 1] + \cdots + Y[1, n]$, the property that $\forall k \in \{0, 1, \ldots, n-1\}$, $X[1, k+1] \geq Y[1, k+1]$ implies $X = Y$. Contradiction. $\square$

The following is a key lemma for proving the monotonicity of our chain.

Lemma 4.3 *If a pair of states $X, Y \in \Xi$ satisfies $X \cdot\!\succ_k Y$, then $\forall \lambda \in [1, n)$, $\phi(X, \lambda) \succeq \phi(Y, \lambda)$.*

Proof: We denote $\phi(X, \lambda) = X'$ and $\phi(Y, \lambda) = Y'$ for simplicity. For any index $i \neq \lfloor \lambda \rfloor$, it is clear that $f_{X'}(i) = f_X(i)$ and $f_{Y'}(i) = f_Y(i)$, and so $f_{X'}(i) - f_{Y'}(i) = f_X(i) - f_Y(i) \geq 0$ since $X \succeq Y$. In case that $i = \lfloor \lambda \rfloor$,
$$\begin{aligned} f_{X'}(\lfloor \lambda \rfloor) - f_{Y'}(\lfloor \lambda \rfloor) &= (f_{X'}(\lfloor \lambda \rfloor - 1) + X'[1, \lfloor \lambda \rfloor]) - (f_{Y'}(\lfloor \lambda \rfloor - 1) + Y'[1, \lfloor \lambda \rfloor]) \\ &= \{f_X(\lfloor \lambda \rfloor - 1) - f_Y(\lfloor \lambda \rfloor - 1)\} + (X'[1, \lfloor \lambda \rfloor] - Y'[1, \lfloor \lambda \rfloor]) \\ &= \{f_X(\lfloor \lambda \rfloor - 1) - f_Y(\lfloor \lambda \rfloor - 1)\} \\ &\quad + \min\{a_X, s_{\lfloor \lambda \rfloor}\} - \lfloor (\lambda - \lfloor \lambda \rfloor)\, \theta_X \rfloor - \min\{a_Y, s_{\lfloor \lambda \rfloor}\} + \lfloor (\lambda - \lfloor \lambda \rfloor)\, \theta_Y \rfloor. \\ &= \begin{cases} \Delta\eta + \Delta\theta & (\lfloor \lambda \rfloor \neq k+1), \\ 1 + \Delta\eta + \Delta\theta & (\lfloor \lambda \rfloor = k+1), \end{cases} \end{aligned}$$
where $a_X \stackrel{\text{def.}}{=} a_X(\lfloor \lambda \rfloor)$, $a_Y \stackrel{\text{def.}}{=} a_Y(\lfloor \lambda \rfloor)$, $\theta_X \stackrel{\text{def.}}{=} \theta_X(\lfloor \lambda \rfloor)$, $\theta_Y \stackrel{\text{def.}}{=} \theta_Y(\lfloor \lambda \rfloor)$ (see (1) and (3) for detail), $\Delta\eta \stackrel{\text{def.}}{=} \min\{a_X, s_{\lfloor \lambda \rfloor}\} - \min\{a_Y, s_{\lfloor \lambda \rfloor}\}$ and $\Delta\theta \stackrel{\text{def.}}{=} -\lfloor (\lambda - \lfloor \lambda \rfloor)\, \theta_X \rfloor + \lfloor (\lambda - \lfloor \lambda \rfloor)\, \theta_Y \rfloor$.

1. Consider the case that $\lfloor \lambda \rfloor = k - 1$. Then $a_X = a_Y + 1$ and $b_X = b_Y - 1$, where $b_X \stackrel{\text{def.}}{=} b_X(\lfloor \lambda \rfloor)$ and $b_Y \stackrel{\text{def.}}{=} b_Y(\lfloor \lambda \rfloor)$ (see (2) for detail).
   (a) If $a_Y \geq s_{\lfloor \lambda \rfloor}$, then $\Delta\eta = 0$ and $\theta_X \leq \theta_Y$. Thus $\Delta\theta \geq 0$, hence $f_{X'}(\lfloor \lambda \rfloor) - f_{Y'}(\lfloor \lambda \rfloor) \geq 0$.
   (b) If $a_Y < s_{\lfloor \lambda \rfloor}$, then $\Delta\eta = 1$ and $\theta_X \leq \theta_Y + 1$. Thus $\Delta\theta \geq -1$, hence $f_{X'}(\lfloor \lambda \rfloor) - f_{Y'}(\lfloor \lambda \rfloor) \geq 0$.
2. Consider the case that $\lfloor \lambda \rfloor = k + 1$. Then $a_X = a_Y - 1$ and $b_X = b_Y + 1$.

(a) If $a_X \geq s_{\lfloor\lambda\rfloor}$, then $1+\Delta\eta \geq 1$ and $\theta_X \leq \theta_Y+1$. Thus $\Delta\theta \geq -1$ and $f_{X'}(\lfloor\lambda\rfloor)-f_{Y'}(\lfloor\lambda\rfloor) \geq 0$.

(b) If $a_X < s_{\lfloor\lambda\rfloor}$, then $1+\Delta\eta \geq 0$. Note that $a_X + b_X = a_Y + b_Y = s_{\lfloor\lambda\rfloor} + s_{\lfloor\lambda\rfloor+1}$, $\theta_X \leq \theta_Y$. Thus $\Delta\theta \geq 0$, hence $f_{X'}(\lfloor\lambda\rfloor) - f_{Y'}(\lfloor\lambda\rfloor) \geq 0$.

3. Consider the remained case that $\lfloor\lambda\rfloor \neq k+1$ and $\lfloor\lambda\rfloor \neq k-1$. Then $a_X = a_Y$, $\Delta\eta = 0$, $\Delta\theta = 0$, and $f_{X'}(\lfloor\lambda\rfloor) - f_{Y'}(\lfloor\lambda\rfloor) = 0$.

From the above, we have $f_{X'} \geq f_{Y'}$ and so $\phi(X,\lambda) \succeq \phi(Y,\lambda)$. $\square$

Lemma 4.4 *The Markov chain $\mathcal{M}$ is monotone, i.e., $\forall\lambda \in [1,n)$, $\forall X, \forall Y \in \Xi$, $X \succeq Y \Rightarrow \phi(X,\lambda) \succeq \phi(Y,\lambda)$.*

Proof: By applying Lemma 4.2 repeatedly, we can show that for any pair of states $X, Y \in \Xi$ satisfying $X \succeq Y$, there exists a sequence $X = Z_0, Z_1, \ldots, Z_R = Y$ with appropriate length such that $Z_i \in \Xi$ $(0 \leq i \leq R)$ and $Z_0 \cdot\!\succ Z_1 \cdot\!\succ \cdots \cdot\!\succ Z_R$. Then Lemma 4.3 implies that $\phi(Z_0,\lambda) \succeq \phi(Z_1,\lambda) \succeq \cdots \succeq \phi(Z_R,\lambda)$ for any $\lambda \in [1,n)$. Thus $\forall\lambda \in [1,n)$, $\phi(X,\lambda) \succeq \phi(Y,\lambda)$. $\square$

Lastly, we show the correctness of our algorithm.

Proof of Theorem 3.1: From Lemma 4.4, the Markov chain $\mathcal{M}$ is monotone and Lemma 4.1 says that $X_{\rm U}$ and $X_{\rm L}$ is a unique pair of the maximum and minimum elements. Then Algorithm 2 is a monotone CFTP algorithm, and so we can show Theorem 3.1 by using Theorem 2.1 and Theorem 2.2. $\square$

5 Expected Running Time

Here, we discuss the running time of our algorithm. In this section, we assume the following.

Condition 1 *Column sum vector $\boldsymbol{s}$ satisfies $s_1 \geq s_2 \geq \cdots \geq s_n$.*

The following is a main result of this paper.

Theorem 5.1 *Under Condition 1, the expected running time of Algorithm 2 is bounded by* $\mathrm{O}(n^3 \ln N)$, *where n is the number of columns and N is the total sum of whole entries in a table of Ξ.*

In the rest of this section, we prove Theorem 5.1 by estimating the expectation of *coalescence time* $T_* \in \mathbb{Z}_{++}$ defined by $T_* \stackrel{\rm def.}{=} \min\{t > 0 \mid \exists y \in \Omega,\ \forall x \in \Omega,\ y = \Phi^0_{-t}(x, \boldsymbol{\Lambda})\}$. Note that T_* is a random variable.

Given a pair of probabilistic distributions ν_1 and ν_2 on the finite state space Ω, the *total variation distance* between ν_1 and ν_2 is defined by $\mathrm{d_{TV}}(\nu_1,\nu_2) \stackrel{\rm def.}{=} \frac{1}{2}\sum_{x\in\Omega} |\nu_1(x) - \nu_2(x)|$. The *mixing rate* of an ergodic Markov chain is defined by $\tau \stackrel{\rm def.}{=} \max_{x\in\Omega}\{\min\{t \mid \forall s \geq t,\ \mathrm{d_{TV}}(\pi, P^s_x) \leq 1/\mathrm{e}\}\}$ where π is the stationary distribution and P^s_x is the probabilistic distribution of the chain at time period $s \geq 0$ with initial state x at time period 0. Path Coupling Theorem is a useful technique for bounding the mixing rate.

Theorem 5.2 (Path Coupling [5]) *Let MC be a finite ergodic Markov chain with state space Ω. Let $G = (\Omega, \mathcal{E})$ be a connected undirected graph with vertex set Ω and edge set $\mathcal{E} \subseteq \binom{\Omega}{2}$. Let $l : \mathcal{E} \to \mathbb{R}$ be a positive length defined on the edge set. For any pair of vertices $\{x, y\}$ of G, the distance between x and y, denoted by $d(x,y)$ and/or $d(y,x)$, is the length of a shortest path between x and y, where the length of a path is the sum of the lengths of edges in the path. Suppose that there exists a joint* *process* $(X,Y) \mapsto (X',Y')$ *with respect to* MC *satisfying that whose marginals are a faithful copy of* MC *and*

$$0 < \exists\beta < 1,\ \forall\{X,Y\} \in \mathcal{E},\ \mathrm{E}[d(X',Y')] \leq \beta d(X,Y).$$

Then the mixing rate τ *of Markov chain* MC *satisfies* $\tau \leq (1-\beta)^{-1}(1+\ln(D/d))$*, where* $d \stackrel{\text{def.}}{=} \min\{d(x,y) \mid \forall x, \forall y \in \Omega\}$ *and* $D \stackrel{\text{def.}}{=} \max\{d(x,y) \mid \forall x, \forall y \in \Omega\}$*.*

The above theorem differs from the original theorem in [5] since the integrality of the edge length is not assumed. We drop the integrality and introduced the minimum distance d. This modification is not essential and we can show Theorem 5.2 similarly.

Now, we show the polynomiality of Algorithm 2. First, we estimate the mixing rate of our chain $\mathcal{M}$ by employing Path Coupling Theorem. In the proof of the following lemma, Condition 1 plays an important role.

Lemma 5.3 *Under Condition 1, the mixing rate* τ *of our Markov chain* $\mathcal{M}$ *satisfies* $\tau \leq n(n-1)^2(1+\ln(nN))$*.*

Proof: Let $G = (\Xi, \mathcal{E})$ be an undirected simple graph with vertex set Ξ and edge set $\mathcal{E}$ defined as follows. A pair of vertices $\{X,Y\}$ is an edge if and only if $(1/2)\sum_{j=1}^{n} |X[1,j] - Y[1,j]| = 1$. Clearly, the graph G is connected. For each edge $e = \{X,Y\} \in \mathcal{E}$, there exists a unique pair of indices $j_1, j_2 \in \{1,\ldots,n\}$, called the *supporting pair* of e, satisfying

$$|X[1,j] - Y[1,j]| = \begin{cases} 1 & (j = j_1, j_2), \\ 0 & (\text{otherwise}). \end{cases}$$

We define the length $l(e)$ of an edge e by $l(e) \stackrel{\text{def.}}{=} (1/(n-1))\sum_{i=1}^{j^*-1}(n-i)$ where $j^* = \max\{j_1, j_2\} \geq 2$ and $\{j_1, j_2\}$ is the supporting pair of e. Note that $1 \leq \min_{e\in\mathcal{E}} l(e) \leq \max_{e\in\mathcal{E}} l(e) \leq n/2$. For each pair $X, Y \in \Xi$, we define the distance $d(X,Y)$ be the length of the shortest path between X and Y on G. Clearly, the diameter of G, i.e., $\max\{d(X,Y)\}$, is bounded by nN. The definition of edge length implies that for any edge $\{X,Y\} \in \mathcal{E}$, $d(X,Y) = l(\{X,Y\})$.

We define a joint process $(X,Y) \mapsto (X',Y')$ by $(X,Y) \mapsto (\phi(X,\Lambda), \phi(Y,\Lambda))$ with uniform real random number $\Lambda \in [1,n)$, where ϕ is the update function defined in Section 3. Now we show that

$$\mathrm{E}[d(X',Y')] \leq \beta d(X,Y),\ \beta = 1 - 1/(n(n-1)^2), \tag{4}$$

for any pair $\{X,Y\} \in \mathcal{E}$. In the following, we denote the supporting pair of $\{X,Y\}$ by $\{j_1, j_2\}$. Without loss of generality, we can assume that $j_1 < j_2$, and $X[1,j_2] = Y[1,j_2] - 1$. In the following proof, we define $a_X, a_Y, b_X, b_Y, \theta_X, \theta_Y$ in a similar way as in the proof of Lemma 4.3.

1. When $\lfloor\Lambda\rfloor = j_2 - 1$, we show that $\mathrm{E}[d(X',Y') \mid \lfloor\Lambda\rfloor = j_2 - 1] \leq d(X,Y) - (1/2)(n-j_2+1)/(n-1)$.
   (a) In case that $j_1 = j_2 - 1$, $X' = Y'$ with conditional probability 1. Hence $d(X',Y') = 0$.
   (b) In case that $j_1 \neq j_2 - 1$ and $\theta_X = \theta_Y$. Condition 1 implies that $s_{j_2-1} \geq s_{j_2}$. Since $\theta_X = \theta_Y$, $a_Y > a_X$ and $b_Y < b_X$, we have $\theta_X = \theta_Y = \min\{s_{j_2-1}, s_{j_2}\} = s_{j_2}$. Thus we have $s_{j_2-1} \geq a_Y > a_X \geq s_{j_2}$, and so $X'[1,j_2-1] = a_X - (\Lambda - \lfloor\Lambda\rfloor)\theta_X$, $Y'[1,j_2-1] = a_Y - (\Lambda - \lfloor\Lambda\rfloor)\theta_Y$ by the definition of the function ϕ. Then $X'[1,j_2-1] = Y'[1,j_2-1] - 1$ since $a_X = a_Y - 1$. Additionally, since $X'[1,j_2] = a_X - X'[1,j_2-1]$ and $Y'[1,j_2] = a_Y - Y'[1,j_2-1]$, we have $X'[1,j_2] = Y'[1,j_2]$. Hence $d(X',Y') = d(X,Y) - (n-j_2+1)/(n-1)$ with conditional probability 1.

(c) In case that $j_1 \neq j_2-1$ and $\theta_X \neq \theta_Y$. Clearly $|\theta_X-\theta_Y|=1$. First, we discuss the case that $\theta_X=\theta_Y-1$. We only need to consider two cases, one is the case that $\lfloor(\Lambda-\lfloor\Lambda\rfloor)\theta_X\rfloor = \lfloor(\Lambda-\lfloor\Lambda\rfloor)\theta_Y\rfloor$ and the other is that $\lfloor(\Lambda-\lfloor\Lambda\rfloor)\theta_X\rfloor = \lfloor(\Lambda-\lfloor\Lambda\rfloor)\theta_Y\rfloor-1$. In the former case, we have $X'[1,j_2-1]=Y'[1,j_2-1]$ and $X'[1,j_2]=Y'[1,j_2]-1$, and so $d(X',Y')=d(X,Y)$. In the latter case, we have $X'[1,j_2-1]=Y'[1,j_2-1]-1$ and $X'[1,j_2]=Y'[1,j_2]$, and so $d(X',Y')=d(X,Y)-(n-j_2+1)/(n-1)$. These two cases appear with the same probability $1/2$, hence $\mathrm{E}[d(X',Y') \mid \lfloor\Lambda\rfloor = j_2-1, j_1\neq j_2-1, \theta_X=\theta_Y-1] = d(X,Y)-(1/2)(n-j_2+1)/(n-1)$. We can show the remained case that $\theta_X=\theta_Y+1$ in a similar way.

2. When $\lfloor\Lambda\rfloor = j_2$, we show that $\mathrm{E}[d(X',Y') \mid \lfloor\Lambda\rfloor = j_2] \leq d(X,Y)+(1/2)(n-j_2)/(n-1)$.
   (a) In case that $\theta_X=\theta_Y$, we obtain the result that $X'[1,j_2]=Y'[1,j_2]-1$ and $X'[1,j_2+1]=Y'[1,j_2+1]$ in the same way as Case 1-(b). Hence $d(X',Y')=d(X,Y)$ with conditional probability 1.
   (b) Consider the case that $\theta_X\neq\theta_Y$. In a similar way with Case 1-(c), we can show that $d(X',Y')=d(X,Y)$ with conditional probability $1/2$ and $d(X',Y')=d(X,Y)+(n-j_2)/(n-1)$ with conditional probability $1/2$. Hence $\mathrm{E}[d(X',Y') \mid \lfloor\Lambda\rfloor=j_2, \theta_X\neq\theta_Y] = d(X,Y)+(1/2)(n-j_2)/(n-1)$.

3. When $\lfloor\Lambda\rfloor \neq j_2-1$ and $\lfloor\Lambda\rfloor\neq j_2$, $\{X',Y'\}$ is also an edge of G. It is easy to see that $j_2=\max\{j_1',j_2'\}$ where $\{j_1',j_2'\}$ is the supporting pair of $\{X',\ Y'\}$. Thus we have $d(X',Y')=d(X,Y)$.

The probability of appearance of Case 1 is equal to $1/(n-1)$, and that of Case 2 is less than or equal to $1/(n-1)$. From the above,

$$\begin{aligned}
\mathrm{E}[d(X',Y')] &\leq d(X,Y)-\frac{1}{n-1}\frac{1}{2}\frac{n-j_2+1}{n-1}+\frac{1}{n-1}\frac{1}{2}\frac{n-j_2}{n-1} = d(X,Y)-\frac{1}{2(n-1)^2}\\
&\leq \left(1-\frac{1}{2(n-1)^2}\frac{1}{\max_{\{X,Y\}\in\mathcal{E}}\{d(X,Y)\}}\right)d(X,Y) = \left(1-\frac{1}{n(n-1)^2}\right)d(X,Y).
\end{aligned}$$

Since the diameter of G is bounded by nN, Theorem 5.2 implies that the mixing rate τ satisfies

$$\tau \leq n(n-1)^2(1+\ln(nN)). \qquad \square$$

Next, we estimate the coalescence time.

Lemma 5.4 *Under Condition 1, the coalescence time T_* of $\mathcal{M}$ satisfies* $\mathrm{E}[T_*]=\mathrm{O}(n^3\ln N)$.

Proof: Let $G=(\Xi,\mathcal{E})$ be the undirected graph and $d(X,Y)$, $\forall X,\forall Y\in\Xi$, be the metric on G, both of which are defined in the proof of Lemma 5.3. We define $D_G \stackrel{\text{def.}}{=} d(X_{\mathrm{U}},X_{\mathrm{L}})$ and $\tau_0 \stackrel{\text{def.}}{=} n(n-1)^2(1+\ln D_G)$. By using the inequality (4) obtained in the proof of Lemma 5.3, we have

$$\begin{aligned}
\Pr(T_*>\tau_0) &= \Pr(\Phi^0_{-\tau_0}(X_{\mathrm{U}},\boldsymbol{\Lambda})\neq\Phi^0_{-\tau_0}(X_{\mathrm{L}},\boldsymbol{\Lambda})) = \Pr(\Phi^{\tau_0}_0(X_{\mathrm{U}},\boldsymbol{\Lambda})\neq\Phi^{\tau_0}_0(X_{\mathrm{L}},\boldsymbol{\Lambda}))\\
&\leq \sum_{(X,Y)\in\Xi^2} d(X,Y)\Pr(X=\Phi^{\tau_0}_0(X_{\mathrm{U}},\boldsymbol{\Lambda}), Y=\Phi^{\tau_0}_0(X_{\mathrm{L}},\boldsymbol{\Lambda}))\\
&= \mathrm{E}[d(\Phi^{\tau_0}_0(X_{\mathrm{U}},\boldsymbol{\Lambda}),\Phi^{\tau_0}_0(X_{\mathrm{L}},\boldsymbol{\Lambda}))] \leq \left(1-\frac{1}{n(n-1)^2}\right)^{\tau_0} d(X_{\mathrm{U}},X_{\mathrm{L}})\\
&= \left(1-\frac{1}{n(n-1)^2}\right)^{n(n-1)^2(1+\ln D_G)} D_G \leq \mathrm{e}^{-1}\mathrm{e}^{-\ln D_G}D_G \leq \frac{1}{\mathrm{e}}.
\end{aligned}$$

By *submultiplicativity* of coalescence time ([18]), for any $k \in \mathbb{Z}_+$, $\Pr(T_* > k\tau_0) \le (\Pr(T_* > \tau_0))^k \le (1/\mathrm{e})^k$. Thus

$$\begin{aligned}
\mathrm{E}[T_*] &= \sum_{t=0}^{\infty} t\Pr(T_* = t) \le \tau_0 + \tau_0 \Pr(T_* > \tau_0) + \tau_0 \Pr(T_* > 2\tau_0) + \cdots \\
&\le \tau_0 + \tau_0/\mathrm{e} + \tau_0/\mathrm{e}^2 + \cdots = \frac{\tau_0}{1 - 1/\mathrm{e}} \le 2\tau_0.
\end{aligned}$$

Clearly, $D_G \le nN \le N^2$ because $n \le N$. Then we obtain the result that $\mathrm{E}[T_*] = \mathrm{O}(n^3 \ln N)$. □

Lastly, we determine the expected running time of Algorithm 2.

Proof of Theorem 5.1: We denote T_* be the coalescence time of our chain. Note that T_* is a random variable. Put $K = \lceil \log_2 T_* \rceil$. Algorithm 2 terminates when we set the starting time period T to -2^K at $(K+1)$st iteration. Then the total number of generated random numbers in Algorithm 2 is bounded by $2^K \le 2T_*$ and the total number of simulated transitions is bounded by $2(2^0 + 2^1 + 2^2 + \cdots + 2^K) < 2 \cdot 2 \cdot 2^K \le 8T_*$. Under the assumption that we can generate a random number in constant time, each transition of a chain is simulated in constant time. Step 4 of Algorithm 2, "Coalescence check," requires $\mathrm{O}(n)$ time. Thus the expectation of total time complexity is bounded by $\mathrm{O}(\mathrm{E}[2T_*] + \mathrm{E}[8T_*] + \mathrm{E}[K+1]n) = \mathrm{O}(\mathrm{E}[T_*]) = \mathrm{O}(n^3 \ln N)$. □

We can assume Condition 1 by sorting column sums in $\mathrm{O}(n \ln n)$ time.

6 Discussions

We proposed a perfect sampling algorithm for $2 \times n$ contingency tables. Our algorithm based on a new Markov chain which is monotone and rapidly mixing. The rapidity implies the polynomiality of our algorithm. More precisely, Algorithm 2 produces exact samples from the uniform distribution, and the expected running time is bounded by $\mathrm{O}(n^3 \ln N)$ under Condition 1.

Our preliminary computational experience indicates that Condition 1 is an important requirement for rapidity. For example, when we set $\boldsymbol{r} = (500, 500)$, $\boldsymbol{s} = (500, 498, 1, 1)$ and executed monotone CFTP algorithm one thousand times, the average coalescence time was about 20. However, when we substituted $\boldsymbol{s}$ by (500,1,1,498) and executed the algorithm one thousand times, the average coalescence time was about 2 millions.

Even though Algorithm 2 is enough fast but needs to stock the random numbers. We can save the memory storage by using read once algorithm proposed by Wilson in [20]. For detail, please see [20, 13]. Note that the modified algorithm terminates in $\mathrm{O}(n^3 \ln N)$ time and the required memory is bounded by $\mathrm{O}(n \ln N)$.

Our perfect sampling algorithm is applicable to the problem for counting $2 \times n$ contingency tables. In our previous work [16], we modified Dyer and Greenhill's approximate counting scheme [10], and estimate the size of bias of the expectation of approximate solution. When we employ our perfect sampling algorithm in the algorithm proposed in [16], we can show that the total number of required samples halved.

It is easy to extend our monotone CFTP algorithm to $2 \times \cdots \times 2 \times J$ tables discussed in [17]. We can show that the extended algorithm is also a polynomial time algorithm for uniform sampler. In case of conditional multinomial distributions, the existence of polynomial time perfect sampler remains open.

Another remained major open problem is the existence of a monotone Markov chain for $m \times n$ contingency tables.

References

[1] A. Agresti, "A survey of exact inference for contingency tables," *Statistical Science*, 7 (1992), pp. 131–153.

[2] A. Agresti, *Categorical Data Analysis*, John Wiley & Sons, 2002.

[3] D. Aldous, "Random walks on finite groups and rapidly mixing Markov chains," in *Séminarie de Probabilitiés XVII 1981/1982*, vol. 986 of Springer-Verlag Lecture Notes in Mathematics, D. Dold and B. Eckmann, ed., Springer-Verlag, New York, 1983, pp. 243–297.

[4] R. Bubley, *Randomized Algorithms: Approximation, Generation, and Counting*, Springer-Verlag, New York, 2001.

[5] R. Bubley and M. Dyer, "Path coupling : a technique for proving rapid mixing in Markov chains," *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS 1997)*, pp. 223–231.

[6] M. Cryan, M. Dyer, L. A. Goldberg, M. Jerrum, and R. Martin, "Rapidly mixing Markov chains for sampling contingency tables with constant number of rows," *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS 2002)*, pp. 711–720.

[7] P. Diaconis and B. Effron, "Testing for independence in a two-way table: new interpretations of the chi-square statistic (with discussion)," *The Annals of Statistics*, 13 (1985), pp. 845–913.

[8] P. Diaconis and L. Saloff-Coste, "Random walk on contingency tables with fixed row and column sums," *Tech. rep.*, Department of Mathematics, Harvard University, 1995.

[9] X. K. Dimakos, "A guide to exact simulation," *International Statistical Review,* 69 (2001), pp. 27–48.

[10] M. Dyer and C. Greenhill, "Polynomial-time counting and sampling of two-rowed contingency tables," *Theoretical Computer Sciences,* 246 (2000), pp. 265–278.

[11] M. Dyer, R. Kannan, and J. Mount, "Sampling contingency tables," *Random Structures and Algorithms*, 10 (1997), pp. 487–506.

[12] R. A. Fisher, "The logic of inductive inference (with discussion)," *Journal of Royal Statistical Society*, 98 (1935), pp. 39–54.

[13] O. Häggström, *Finite Markov Chains and Algorithmic Application*, London Mathematical Society, Student Texts 52, Cambridge University Press, 2002.

[14] D. Hernek, "Random generation of $2 \times n$ contingency tables," *Random Structures and Algorithms*, 13 (1998), pp. 71–79.

[15] M. Jerrum and A. Sinclair, "The Markov chain Monte Carlo method: an approach to approximate counting and integration," in *Approximation Algorithm for NP-hard Problems*, D. Hochbaum, ed., PWS, 1996, pp. 482–520.

[16] S. Kijima and T. Matsui, "Approximate counting scheme for $m \times n$ contingency tables ," *METR* 2003-01, Mathematical Engineering Technical Reports, University of Tokyo, 2003. (available from http://www.keisu.t.u-tokyo.ac.jp/Research/techrep.0.html)

[17] T. Matsui, Y. Matsui, and Y. Ono, "Random generate of $2 \times \cdots \times 2 \times J$ contingency tables," *METR* 2003-03, Mathematical Engineering Technical Reports, University of Tokyo, 2003. (available from http://www.keisu.t.u-tokyo.ac.jp/Research/techrep.0.html)

[18] J. Propp and D. Wilson, "Exact sampling with coupled Markov chains and applications to statistical mechanics," *Random Structures and Algorithms,* 9 (1996), pp. 232–252.

[19] J. Propp and D. Wilson, "How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph," *J. Algorithms,* 27 (1998), pp. 170–217.

[20] D. Wilson, "How to couple from the past using a read-once source of randomness," *Random Structures and Algorithms*, 16 (2000), pp. 85–113.