MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Polynomial Time Perfect Sampler for Discretized Dirichlet Distribution

Tomomi MATSUI and Shuji KIJIMA

METR 2003–17 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 Sampler for Discretized Dirichlet Distribution

Tomomi Matsui* and Shuji Kijima

Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan.
http://www.simplex.t.u-tokyo.ac.jp/~tomomi/
kijima@simplex.t.u-tokyo.ac.jp

version: April 24th

Abstract. In this paper, we propose a perfect (exact) sampling algorithm according to a discretized Dirichlet distribution. Our algorithm is a monotone coupling from the past algorithm, which is a Las Vegas type randomized algorithm. We propose a new Markov chain whose limit distribution is a discretized Dirichlet distribution. Our algorithm simulates transitions of the chain $\mathrm{O}(n^3 \ln \Delta)$ times where n is the dimension (the number of parameters) and $1/\Delta$ is the grid size for discretization. Thus the obtained bound does not depend on the magnitudes of parameters. In each transition, we need to sample a random variable according to a discretized beta distribution (2-dimensional Dirichlet distribution). To show the polynomiality, we employ the path coupling method carefully and show that our chain is rapidly mixing.

1 Introduction

In this paper, we propose a polynomial time perfect (exact) sampling algorithm according to a discretized Dirichlet distribution. Our algorithm is a monotone coupling from the past algorithm, which is a Las Vegas type randomized algorithm. We propose a new Markov chain for generating random samples according to a discretized Dirichlet distribution. In each transition of our Markov chain, we need a random sample according to a discretized beta distribution (2-dimensional Dirichlet distribution). Our algorithm simulates transitions of the Markov chain $\mathrm{O}(n^3 \ln \Delta)$ times in expectation where n is the dimension (the number of parameters) and $1/\Delta$ is the grid size for discretization. To show the polynomiality, we employ the path coupling method and show that the mixing rate of our chain is bounded by $n(n-1)^2(1 + \ln n(\Delta - n)/2)$.

Statistical methods are widely studied in bioinformatics since they are powerful tools to discover genes causing a (common) disease from a number of observed data. These methods often use EM algorithm, Markov chain Monte Carlo

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

method, Gibbs sampler, and so on. The Dirichlet distribution appears as prior and posterior distribution for the multinomial distribution in these methods since the Dirichlet distribution is the conjugate prior of parameters of the multinomial distribution [11]. For example, Niu, Qin, Xu, and Liu proposed a Bayesian haplotype inference method [7], which decides phased (paternal and maternal) individual genotypes probabilistically. Another example is a population structure inferring algorithm by Pritchard, Stephens, and Donnely [8]. Their algorithm is based on MCMC method. In these examples, the Dirichlet distribution appears with various dimensions and various parameters. Thus we need an efficient algorithm for sampling from the Dirichlet distribution with arbitrary dimensions and parameters.

One approach of sampling from a (continuous) Dirichlet distribution is by rejection (see [4] for example). In this way, the number of required samples from the gamma distribution is equal to the size of the dimension of the Dirichlet distribution. Though we can sample from the gamma distribution by using rejection sampling, the ratio of rejection becomes higher as the parameter is smaller. Thus, it does not seems effective way for small parameters. Another approach is a descretization of the domain and adoption of Metropolis-Hastings algorithm. Recently, Matsui, Motoki and Kamatani proposed a Markov chain for sampling from discretized Dirichlet distribution [6]. The mixing rate of their chain is bounded by $(1/2)n(n-1)(1+\ln(\Delta-n))$ and so their algorithm is a polynomial time approximate sampler.

Propp and Wilson devised a surprising algorithm, called (monotone) CFTP algoritm (or backward coupling), which produces exact samples from the limit distribution [9, 10]. Monotone CFTP algorithm simulates infinite time transitions of a monotone Markov chain in a (probabilistically) finite time. To employ their result, we propose a new monotone Markov chain whose stationary distribution is a discretized Dirichlet distribution. We also show that the mixing rate of our chain is bounded by $n(n-1)^2(1+\ln n(\Delta-n)/2)$.

This paper is deeply related to authors' recent paper [5] which proposes a polynomial time perfect sampling algorithm for two-rowed contingency tables. The idea of the algorithm and outline of the proof described in this paper is similar to those in our paper [5]. The main difference of these two papers is that when we deal with the uniform distribution on two-rowed contingency tables, the transition probabilities of the Markov chain are predetermined. However, in case of discretized Dirichlet distribution, the transition probabilities vary with respect to the magnitude of paprameters. Thus, to show the monotonicity and polynomiality of the algorithm proposed in this paper, we need a detailed discussions which appear in Appendix section. Showing two lemmas in Appendix section is easier in case of the uniform distribution on two-rowed contingency tables.

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.

CFTP Algorithm [9]

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 $\boldsymbol{\lambda}$. (Note that every chain uses the common sequence $\boldsymbol{\lambda}$.)

Step 4. [Coalescence check]
If $\exists y \in \Omega$, $\forall x \in \Omega$, $y = \Phi_T^0(x, \boldsymbol{\lambda})$, then return y and stop.
Else, update the starting time period $T := 2T$, and go to Step 2.

Theorem 1. (CFTP Theorem [9]) *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 terminates with probability 1, then the obtained value is a realization of a random variable exactly distributed according to the stationary distribution.*

Theorem 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. (monotone CFTP [9,3]) *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 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, we can modify CFTP algorithm 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 and stop.

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

3 Perfect Sampling Algorithm

In this paper, we denote the set of integers (non-negative or positive integers) by Z (Z_+, Z_{++}). Dirichlet random vector $P = (P_1, P_2, \ldots, P_n)$ with non-negative parameters $u_1, \ldots, u_n$ is a vector of random variables that admits the probability density function

$$\frac{\Gamma(\sum_{i=1}^n u_i)}{\prod_{i=1}^n \Gamma(u_i)} \prod_{i=1}^n p_i^{u_i - 1}$$

defined on the set $\{(p_1, p_2, \ldots, p_n) \in \mathrm{R}^n \mid p_1 + \cdots + p_n = 1, p_1, p_2, \ldots, p_n > 0\}$ where $\Gamma(u)$ is the gamma function. Throughout this paper, we assume that $n \geq 2$.

For any integer $\Delta \geq n$, we discretize the domain with grid size $1/\Delta$ and obtain a discrete set of integer vectors Ω defined by

$$\Omega \stackrel{\text{def.}}{=} \{(x_1, x_2, \ldots, x_n) \in \mathrm{Z}_{++}^n \mid x_i > 0 \ (\forall i),\ x_1 + \cdots + x_n = \Delta\}.$$

A discretized Dirichlet random vector with non-negative parameters $u_1, \ldots, u_n$ is a random vector $X = (X_1, \ldots, X_n) \in \Omega$ with the distribution

$$\Pr[X = (x_1, \ldots, x_n)] \stackrel{\text{def.}}{=} C_\Delta \prod_{i=1}^n (x_i/\Delta)^{u_i - 1}$$

where C_Δ is the partition function (normarizing constant) defined by $(C_\Delta)^{-1} \stackrel{\text{def.}}{=} \sum_{\boldsymbol{x} \in \Omega} \prod_{i=1}^n (x_i/\Delta)^{u_i - 1}$.

For any integer $b \geq 2$, we introduce a set of 2-dimensional integer vectors $\Omega(b) \stackrel{\text{def.}}{=} \{(Y_1, Y_2) \in \mathrm{Z}^2 \mid Y_1, Y_2 > 0,\ Y_1 + Y_2 = b\}$ and a distribution function $f_b(Y_1, Y_2 \mid u_i, u_j) : \Omega(b) \to [0, 1]$ with non-negative parameters u_i, u_j defined by

$$f_b(Y_1, Y_2 \mid u_i, u_j) \stackrel{\text{def.}}{=} C(u_i, u_j, b) Y_1^{u_i - 1} Y_2^{u_j - 1}$$

where $(C(u_i, u_j, b))^{-1} \stackrel{\text{def.}}{=} \sum_{(Y_1, Y_2) \in \Omega(b)} Y_1^{u_i - 1} Y_2^{u_j - 1}$ is the partition function. We also introduce a vector $(g_b(0|u_i, u_j), g_b(1|u_i, u_j), \ldots, g_b(b-1|u_i, u_j))$ defined by

$$g_b(k|u_i, u_j) \stackrel{\text{def.}}{=} \begin{cases} 0 & (k = 0) \\ \sum_{l=1}^{k} C(u_i, u_j, b) l^{u_i - 1} (b - l)^{u_j - 1} & (k \in \{1, 2, \ldots, b-1\}) \end{cases}$$

It is clear that $0 = g_b(0|u_i, u_j) < g_b(1|u_i, u_j) < \cdots < g_b(b-1|u_i, u_j) = 1$.

We describe our Markov chain $\mathcal{M}$ with state space Ω. Given a current state $X \in \Omega$, we generate a random number $\lambda \in [1, n)$. Then transition $X \mapsto X'$ with respect to λ takes place as follows.

Markov chain $\mathcal{M}$

Input: A random number $\lambda \in [1, n)$.

Step 1: Put $i := \lfloor \lambda \rfloor$ and $b := X_i + X_{i+1}$.

Step 2: Let $k \in \{1, 2, \ldots, b-1\}$ be a unique value satisfying

$$g_b(k-1|u_i, u_{i+1}) \leq (\lambda - \lfloor \lambda \rfloor) < g_b(k|u_i, u_{i+1}).$$

Step 3: Put $X'_j := \begin{cases} k & (j = i), \\ b - k & (j = i+1), \\ X_j & (\text{otherwise}). \end{cases}$

The update function $\phi : \Omega \times [1, n) \to \Omega$ of our chain is defined by $\phi(X, \lambda) \stackrel{\text{def.}}{=} X'$ where X' is determined by the above procedure. Clearly, this chain is irreducible and aperiodic. Since the detailed balance equations hold, the stationary distribution of the above Markov chain $\mathcal{M}$ is the discretized Dirichlet distribution.

We define two special states $X_{\mathrm{U}}, X_{\mathrm{L}} \in \Omega$ by

$$X_{\mathrm{U}} \stackrel{\text{def.}}{=} (\Delta - n + 1, 1, 1, \ldots, 1), \quad X_{\mathrm{L}} \stackrel{\text{def.}}{=} (1, 1, \ldots, 1, \Delta - n + 1).$$

Now we describe our algorithm.

Algorithm 1

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

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

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]
If $\exists Y \in \Omega,\ Y = \Phi_T^0(X_{\mathrm{U}}, \boldsymbol{\lambda}) = \Phi_T^0(X_{\mathrm{L}}, \boldsymbol{\lambda})$, then return Y and stop.
Else, update the starting time period $T := 2T$, and go to Step 2.

Theorem 3. *With probability 1, Algorithm 2 terminates and returns a state. The state obtained by Algorithm 2 is a realization of a sample exactly according to the discretized Dirichlet distribution.*

The above theorem guarantees that Algorithm 2 is a perfect sampling algorithm. We prove the above theorem by showing the monotonicity in the next section.

4 Monotonicity of the Chain

In Section 2, we described two theorems. Thus we only need to show the monotonicity of our chain. First, we introduce a partial order on the state space Ω.

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

$$\mathrm{c}_X(i) \stackrel{\text{def.}}{=} \begin{cases} 0 & (i = 0), \\ X_1 + X_2 + \cdots + X_i & (i \in \{1, 2, \ldots, n\}), \end{cases}$$

where $\mathrm{c}_X = (\mathrm{c}_X(0), \mathrm{c}_X(1), \ldots, \mathrm{c}_X(n))$. Clearly, there exists a bijection between Ω and the set $\{\mathrm{c}_X \mid X \in \Omega\}$. For any pair of states $X, Y \in \Omega$, we say $X \succeq Y$ if and only if $\mathrm{c}_X \geq \mathrm{c}_Y$. It is clear that the relation "$\succeq$" is a partial order on Ω. We can see easily that $\forall X \in \Omega,\ X_{\mathrm{U}} \succeq X \succeq X_{\mathrm{L}}$.

We say that a state $X \in \Omega$ *covers* $Y \in \Omega$ (at j), denoted by $X \cdot\!\succ Y$ (or $X \cdot\!\succ_j Y$), when

$$\mathrm{c}_X(i) - \mathrm{c}_Y(i) = \begin{cases} 1 & (i = j), \\ 0 & (\text{otherwise}). \end{cases}$$

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

$$X_i - Y_i = \begin{cases} +1 & (i = j), \\ -1 & (i = j+1), \\ 0 & (\text{otherwise}). \end{cases}$$

Next, we show a key lemma for proving monotonicity.

Lemma 1. $\forall X, \forall Y \in \Omega,\ \forall \lambda \in [1, n),\ X \cdot\!\succ_j Y \Rightarrow \phi(X, \lambda) \succeq \phi(Y, \lambda)$.

Proof: We denote $\phi(X, \lambda)$ by X' and $\phi(Y, \lambda)$ by Y' for simplicity. For any index $i \neq \lfloor \lambda \rfloor$, it is easy to see that $\mathrm{c}_X(i) = \mathrm{c}_{X'}(i)$ and $\mathrm{c}_Y(i) = \mathrm{c}_{Y'}(i)$, and so $\mathrm{c}_{X'}(i) - \mathrm{c}_{Y'}(i) = \mathrm{c}_X(i) - \mathrm{c}_Y(i) \geq 0$ since $X \succeq Y$. In the following, we show that $\mathrm{c}_{X'}(\lfloor \lambda \rfloor) \geq \mathrm{c}_{Y'}(\lfloor \lambda \rfloor)$.

Clearly from the definition of our Markov chain, $X'_{\lfloor \lambda \rfloor}$ is a unique value k' satisfying

$$g_{b'}(k' - 1 | u_{\lfloor \lambda \rfloor}, u_{\lfloor \lambda \rfloor + 1}) \leq (\lambda - \lfloor \lambda \rfloor) < g_{b'}(k' | u_{\lfloor \lambda \rfloor}, u_{\lfloor \lambda \rfloor + 1})$$

where $b' \stackrel{\text{def.}}{=} X_{\lfloor\lambda\rfloor} + X_{\lfloor\lambda\rfloor+1}$. Similarly, $Y'_{\lfloor\lambda\rfloor}$ is a unique value k'' satisfying

$$g_{b''}(k''-1|u_{\lfloor\lambda\rfloor}, u_{\lfloor\lambda\rfloor+1}) \leq (\lambda - \lfloor\lambda\rfloor) < g_{b''}(k''|u_{\lfloor\lambda\rfloor}, u_{\lfloor\lambda\rfloor+1})$$

where $b'' \stackrel{\text{def.}}{=} Y_{\lfloor\lambda\rfloor} + Y_{\lfloor\lambda\rfloor+1}$. We need to consider following three cases.
Case 1: If $\lfloor\lambda\rfloor \neq j-1$ and $\lfloor\lambda\rfloor \neq j+1$, then $b' = b''$ and so we have $X'_{\lfloor\lambda\rfloor} = k' = k'' = Y'_{\lfloor\lambda\rfloor}$.
Case 2: Consider the case that $\lfloor\lambda\rfloor = j-1$. Since $X \succ_j Y$, we have $b' = b''+1$. From the definition of cumulative sum vector,

$$\begin{aligned}&\mathrm{c}_{X'}(j-1) - \mathrm{c}_{Y'}(j-1) = \mathrm{c}_{X'}(j-2) + X'_{j-1} - \mathrm{c}_{Y'}(j-2) - Y'_{j-1}\\ &= \mathrm{c}_X(j-2) + X'_{j-1} - \mathrm{c}_Y(j-2) - Y'_{j-1} = X'_{j-1} - Y'_{j-1}.\end{aligned}$$

Thus, it is enough to show that $X'_{j-1} \geq Y'_{j-1}$.
Lemma 5 in Appendix section implies the following inequalities,

$$\begin{aligned}0 &= g_{b''+1}(0|u_{j-1}, u_j) = g_{b''}(0|u_{j-1}, u_j) \leq g_{b''+1}(1|u_{j-1}, u_j) \leq g_{b''}(1|u_{j-1}, u_j) \leq \cdots\\ &\leq g_{b''+1}(k-1|u_{j-1}, u_j) \leq g_{b''}(k-1|u_{j-1}, u_j) \leq g_{b''+1}(k|u_{j-1}, u_j) \leq \cdots\\ &\leq g_{b''+1}(b''-1|u_{j-1}, u_j) \leq g_{b''}(b''-1|u_{j-1}, u_j) = g_{b''+1}(b''|u_{j-1}, u_j) = 1,\end{aligned}$$

which we will call *alternating inequalities*. For example, if inequalities

$$g_{b''+1}(k-1|u_{j-1}, u_j) \leq (\lambda - \lfloor\lambda\rfloor) < g_{b''}(k-1|u_{j-1}, u_j) \leq g_{b''+1}(k|u_{j-1}, u_j)$$

hold, then $X'_{\lfloor\lambda\rfloor} = k > k-1 = Y'_{\lfloor\lambda\rfloor}$. And if

$$g_{b''+1}(k-1|u_{j-1}, u_j) \leq g_{b''}(k-1|u_{j-1}, u_j) \leq (\lambda - \lfloor\lambda\rfloor) < g_{b''+1}(k|u_{j-1}, u_j)$$

hold, then $X'_{\lfloor\lambda\rfloor} = k = Y'_{\lfloor\lambda\rfloor}$. Thus we have

$$\begin{pmatrix} X'_{j-1} \\ Y'_{j-1} \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \ldots, \begin{pmatrix} b''-1 \\ b''-1 \end{pmatrix}, \begin{pmatrix} b'' \\ b''-1 \end{pmatrix} \right\}$$

(see Figure 1). From the above we have that $X'_{j-1} \geq Y'_{j-1}$.

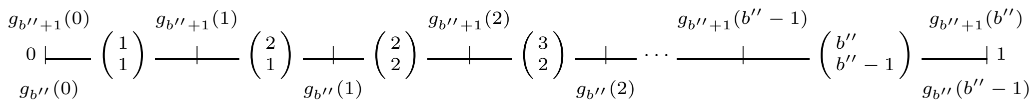


Fig. 1. A figure of alternating inequalities. In the above, we denote $g_{b''}(k|u_{j-1}, u_j)$ and $g_{b''+1}(k|u_{j-1}, u_j)$ by $g_{b''}(k), g_{b''+1}(k)$, respectively.

Case 3: Consider the case that $\lfloor\lambda\rfloor = j+1$. Since $X \succ_j Y$, we have $b'+1 = b''$. From the definition of cumulative sum vector,

$$\begin{aligned}&\mathrm{c}_{X'}(j+1) - \mathrm{c}_{Y'}(j+1) = \mathrm{c}_{X'}(j) + X'_{j+1} - \mathrm{c}_{Y'}(j) - Y'_{j+1}\\ &= \mathrm{c}_X(j) + X'_{j+1} - \mathrm{c}_Y(j) - Y'_{j+1} = 1 + X'_{j+1} - Y'_{j+1}.\end{aligned}$$

Thus, it is enough to show that $1 + X'_{j+1} \geq Y'_{j+1}$.

Then Lemma 5 also implies the following alternating inequalities

$$\begin{aligned}
0 &= g_{b'+1}(0|u_{j+1}, u_{j+2}) = g_{b'}(0|u_{j+1}, u_{j+2}) \\
&\leq g_{b'+1}(1|u_{j+1}, u_{j+2}) \leq g_{b'}(1|u_{j+1}, u_{j+2}) \leq \cdots \\
&\leq g_{b'+1}(k-1|u_{j+1}, u_{j+2}) \leq g_{b'}(k-1|u_{j+1}, u_{j+2}) \leq g_{b'+1}(k|u_{j+1}, u_{j+2}) \leq \cdots \\
&\leq g_{b'+1}(b'-1|u_{j+1}, u_{j+2}) \leq g_{b'}(b'-1|u_{j+1}, u_{j+2}) = g_{b'+1}(b'|u_{j+1}, u_{j+2}) = 1.
\end{aligned}$$

Then it is easy to see that

$$\begin{pmatrix} X'_{j+1} \\ Y'_{j+1} \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \ldots, \begin{pmatrix} b'-1 \\ b'-1 \end{pmatrix}, \begin{pmatrix} b'-1 \\ b' \end{pmatrix} \right\}$$

(see Figure 2). From the above, we obtain the inequality $1 + X'_{j+1} \geq Y'_{j+1}$. $\square$

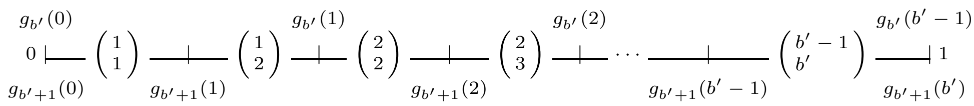


Fig. 2. A figure of alternating inequalities. In the above, we denote $g_{b'}(k|u_{j+1}, u_{j+2})$ and $g_{b'+1}(k|u_{j+1}, u_{j+2})$ by $g_{b'}(k), g_{b'+1}(k)$, respectively.

Lemma 2. *The Markov chain $\mathcal{M}$ defined by the update function ϕ is monotone with respect to "$\succeq$", i.e., $\forall \lambda \in [1, n),\ \forall X, \forall Y \in \Omega,\ X \succeq Y \Rightarrow \phi(X, \lambda) \succeq \phi(Y, \lambda)$.*

Proof: It is easy to see that there exists a sequence of states $Z_1, Z_2, \ldots, Z_r$ with appropreate length satisfying $X = Z_1 \cdot\!\succ Z_2 \cdot\!\succ \cdots \cdot\!\succ Z_r = Y$. Then applying Lemma 1 repeatedly, we can show that $\phi(X, \lambda) = \phi(Z_1, \lambda) \succeq \phi(Z_2, \lambda) \succeq \cdots \succeq \phi(Z_r, \lambda) = \phi(Y, \lambda)$. $\square$

Lastly, we show the correctness of our algorithm.

Proof of Theorem 3: From Lemma 2, the Markov chain is monotone, and it is clear that $(X_{\mathrm{U}}, X_{\mathrm{L}})$ is a unique pair of maximum and minimum states. Thus Algorithm 2 is a monotone CFTP algorithm and Theorems 1 and 2 implies Theorem 3. $\square$

5 Expected Running Time

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

Condition 1 *Parameters are arranged in non-increasing order, i.e., $u_1 \geq u_2 \geq \cdots \geq u_n$.*

The following is a main result of this paper.

Theorem 4. *Under Condition 1, the expected number of tansitions executed in Algorithm 1 is bounded by* $\mathrm{O}(n^3 \ln \Delta)$ *where* n *is the dimension (number of parameters) and* $1/\Delta$ *is the grid size for discretization.*

In the rest of this section, we prove Theorem 4 by estimating the expectation of *coalescence time* $T_* \in \mathrm{Z}_{++}$ defined by $T_* \stackrel{\text{def.}}{=} \min\{t > 0 \mid \exists y \in \Omega,\ \forall x \in \Omega,\ y = \Phi^0_{-t}(x, \boldsymbol{\Lambda})\}$. 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 $\mathrm{D_{TV}}(\nu_1, \nu_2) \stackrel{\text{def.}}{=} \frac{1}{2}\sum_{x\in\Omega} |\nu_1(x) - \nu_2(x)|$. The *mixing rate* of an ergodic Markov chain is defined by $\tau \stackrel{\text{def.}}{=} \max_{x\in\Omega}\{\min\{t \mid \forall s \geq t,\ \mathrm{D_{TV}}(\pi, P^s_x) \leq 1/\mathrm{e}\}\}$ where π is the stationary distribution and P^s_x is the probabilistic distribution of the chain at time s with initial state x. Path Coupling Theorem is a useful technique for bounding the mixing rate.

Theorem 5. (Path Coupling [1, 2]) *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* $l : \mathcal{E} \to \mathrm{R}$ *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* $d \stackrel{\text{def.}}{=} \min_{(x,y)\in\Omega^2} d(x, y)$ *and* $D \stackrel{\text{def.}}{=} \max_{(x,y)\in\Omega^2} d(x, y)$*.*

The above theorem differs from the original theorem in [1] 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 similarly.

Now, we show the polynomiality of our algorithm. 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 3. *Under Condition 1, the mixing rate* τ *of our Markov chain* $\mathcal{M}$ *satisfies* $\tau \leq n(n-1)^2(1 + \ln n(\Delta - n)/2)$*.*

Proof. Let $G = (\Omega, \mathcal{E})$ be an undirected simple graph with vertex set Ω and edge set $\mathcal{E}$ defined as follows. A pair of vertices $\{X, Y\}$ is an edge if and only if $(1/2)\sum_{i=1}^{n} |X_i - Y_i| = 1$. 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

$$|X_i - Y_i| = \begin{cases} 1 \ (i = j_1, j_2), \\ 0 \ (\text{otherwise}). \end{cases}$$

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 $X, Y \in \Omega$, 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., $\max_{(X,Y) \in \Omega^2} d(X, Y)$, is bounded by $n(\Delta - n)/2$, since $d(X, Y) \leq (n/2) \sum_{i=1}^{n} (1/2)|X_i - Y_i| \leq (n/2)(\Delta - n)$ for any $(X, Y) \in \Omega^2$. 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

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

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 $X_{j_2} + 1 = Y_{j_2}$.

Case 1: When $\lfloor \Lambda \rfloor = j_2 - 1$, we will show that

$$\mathrm{E}[d(X', Y') | \lfloor \Lambda \rfloor = j_2 - 1] \leq d(X, Y) - (1/2)(n - j_2 + 1)/(n-1).$$

In case $j_1 = j_2 - 1$, $X' = Y'$ with conditional probabilty 1. Hence $d(X', Y') = 0$. In the following, we consider the case $j_1 < j_2 - 1$. Put $b' = X_{j_2-1} + X_{j_2}$ and $b'' = Y_{j_2-1} + Y_{j_2}$. Since $X_{j_2} + 1 = Y_{j_2}$, $b' + 1 = b''$ holds. From the definition of the update function of our Marokov chain, we have followings,

$$\begin{aligned} X'_{j_2-1} = k &\Leftrightarrow [g_{b'}(k-1|u_{j_2-1}, u_{j_2}) \leq \Lambda - \lfloor \Lambda \rfloor < g_{b'}(k|u_{j_2-1}, u_{j_2})] \\ Y'_{j_2-1} = k &\Leftrightarrow [g_{b'+1}(k-1|u_{j_2-1}, u_{j_2}) \leq \Lambda - \lfloor \Lambda \rfloor < g_{b'+1}(k|u_{j_2-1}, u_{j_2})]. \end{aligned}$$

As described in the proof of Lemma 1, the alternating inequalities

$$\begin{aligned} 0 &= g_{b'+1}(0|u_{j_2-1}, u_{j_2}) = g_{b'}(0|u_{j_2-1}, u_{j_2}) \\ &\leq g_{b'+1}(1|u_{j_2-1}, u_{j_2}) \leq g_{b'}(1|u_{j_2-1}, u_{j_2}) \leq \cdots \\ &\leq g_{b'+1}(b'-1|u_{j_2-1}, u_{j_2}) \leq g_{b'}(b'-1|u_{j_2-1}, u_{j_2}) = g_{b'+1}(b'|u_{j_2-1}, u_{j_2}) = 1, \end{aligned}$$

hold. Thus we have

$$\begin{pmatrix} X'_{j_2-1} \\ Y'_{j_2-1} \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \ldots, \begin{pmatrix} b'-1 \\ b'-1 \end{pmatrix}, \begin{pmatrix} b'-1 \\ b' \end{pmatrix} \right\}.$$

If $X'_{j_2-1} = Y'_{j_2-1}$, the supporting pair of $\{X', Y'\}$ is $\{j_1, j_2\}$ and so $d(X', Y') = d(X, Y)$. If $X'_{j_2-1} \neq Y'_{j_2-1}$, the supporting pair of $\{X', Y'\}$ is $\{j_1, j_2 - 1\}$ and so $d(X', Y') = d(X, Y) - (n - j_2 + 1)/(n-1)$.

Lemma 6 in Appendix section implies that if $u_{j_2-1} \geq u_{j_2}$, then

$$\Pr[X'_{j_2-1} = Y'_{j_2-1} | \lfloor \Lambda \rfloor = j_2 - 1] \leq (1/2),$$
$$\Pr[X'_{j_2-1} \neq Y'_{j_2-1} | \lfloor \Lambda \rfloor = j_2 - 1] \geq (1/2),$$

by showing that the following inequality

$$\begin{aligned}\Pr[X'_{j_2-1} \neq Y'_{j_2-1} | \lfloor \Lambda \rfloor = j_2 - 1] - \Pr[X'_{j_2-1} = Y'_{j_2-1} | \lfloor \Lambda \rfloor = j_2 - 1]\\ = \textstyle\sum_{k=1}^{b'-1} [g_{b'}(k|u_{j_2-1}, u_{j_2}) - g_{b'+1}(k|u_{j_2-1}, u_{j_2})]\\ - \textstyle\sum_{k=1}^{b'-1} [g_{b'+1}(k|u_{j_2-1}, u_{j_2}) - g_{b'}(k-1|u_{j_2-1}, u_{j_2})] \geq 0\end{aligned}$$

hold when $u_{j_2-1} \geq u_{j_2}$ (see Figure 3).

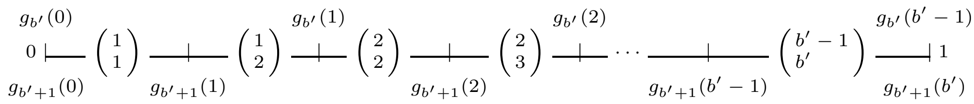


Fig. 3. A figure of alternating inequalities. In the above, we denote $g_{b'}(k|u_{j_2-1}, u_{j_2})$ and $g_{b'+1}(k|u_{j_2-1}, u_{j_2})$ by $g_{b'}(k), g_{b'+1}(k)$, respectively.

Condition 1 is necessary to show Lemma 6. Thus we obtain that

$$\begin{aligned}\mathrm{E}[d(X', Y') | \lfloor \Lambda \rfloor = j_2 - 1] &\leq (1/2)d(X, Y) + (1/2)(d(X, Y) - (n - j_2 + 1)/(n - 1))\\ &= d(X, Y) - (1/2)(n - j_2 + 1)/(n - 1).\end{aligned}$$

Case 2: When $\lfloor \Lambda \rfloor = j_2$, we can show that $\mathrm{E}[d(X', Y') | \lfloor \Lambda \rfloor = j_2] \leq d(X, Y) + (1/2)(n - j_2)/(n - 1)$ in a similar way with Case 1.

Case 3: When $\lfloor \Lambda \rfloor \neq j_2 - 1$ and $\lfloor \Lambda \rfloor \neq j_2$, it is easy to see that the supporting pair $\{j'_1, j'_2\}$ of $\{X', Y'\}$ satisfies $j_2 = \max\{j'_1, j'_2\}$. Thus $d(X, Y) = d(X', Y')$.

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 $n(\Delta - n)/2$, Theorem 5 implies that the mixing rate τ satisfies $\tau \leq n(n-1)^2(1 + \ln n(\Delta - n)/2)$. $\square$

Next, we estimate the coalescence time. When we apply Propp and Wilson's result in [9] straightforwardly, the obtained coalescence time is not tight. In the following, we use their technique carefully and derive a better bound.

Lemma 4. *Under Condition 1, the coalescence time T_* of $\mathcal{M}$ satisfies* $\mathrm{E}[T_*] = \mathrm{O}(n^3 \ln \Delta)$.

Proof. Let $G = (\Omega, \mathcal{E})$ be the undirected graph and $d(X,Y)$, $\forall X, \forall Y \in \Omega$, be the metric on G, both of which are defined in the proof of Lemma 3. We define $D \stackrel{\text{def.}}{=} d(X_{\mathrm{U}}, X_{\mathrm{L}})$ and $\tau_0 \stackrel{\text{def.}}{=} n(n-1)^2(1+\ln D)$. By using the inequality (1) obtained in the proof of Lemma 3, we have

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

By *submultiplicativity* of coalescence time ([9]), for any $k \in \mathrm{Z}_+$, $\Pr[T_* > k\tau_0] \leq (\Pr[T_* > \tau_0])^k \leq (1/\mathrm{e})^k$. Thus

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

Clearly, $D \leq n(\Delta - n)/2 \leq \Delta^2$ because $n \leq \Delta$. Then we obtain the result that $\mathrm{E}[T_*] = \mathrm{O}(n^3 \ln \Delta)$. □

Lastly, we bound the expected number of transitions executed in Algorithm 1.

Proof of Theorem 4: We denote T_* be the coalescence time of our chain. Clearly T_* is a random variable. Put $K = \lceil \log_2 T_* \rceil$. Algorithm 1 terminates when we set the starting time period $T = -2^K$ at $(K+1)$st iteration. Then the total number of simulated transitions is bounded by $2(2^0 + 2^1 + 2^2 + \cdots + 2^K) < 2 \cdot 2 \cdot 2^K \leq 8T_*$, since we need to execute two chains from both X_{U} and X_{L}. Thus the expectation of total number of transitions of $\mathcal{M}$ is bounded by $\mathrm{O}(\mathrm{E}[8T_*]) = \mathrm{O}(n^3 \ln N)$. □

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

6 Conclusion

In this paper, we proposed a monotone Markov chain whose stationary distribution is a discretized Dirichlet distribution. By employing Propp and Wilson's result, on monotone CFTP algorithm, we can construct a perfect sampling algorithm. We showed that our Markov chain is rapidly mixing by using path coupling method. Thus the rapidity implies that our perfect sampling algorithm is a polynomial time algorithm. The obtained time complexity does not depend on the magnitudes of parameters.

We can reduce the memory requirement by employing Wilson's read-once algorithm in [12].

Appendix

Lemma 5 is essentialy equivalent to the lemma appearing in Appendix section of the paper [6] by Matsui, Motoki and Kamatani which deals with an approximate sampler for discretized Dirichlet distribution.

Lemma 6 is a new result obtained in this paper.

Lemma 5.

$$\forall b \in \{2,3,\ldots\},\ \forall u_i, \forall u_j \geq 0,\ \forall k \in \{1,2,\ldots,b\},$$
$$g_{b+1}(k-1|u_i,u_j) \leq g_b(k-1|u_i,u_j) \leq g_{b+1}(k|u_i,u_j).$$

Proof: In the following, we show the second inequality. We can show the first inequality in a similar way.

We denote $C(u_i,u_j,b+1) = C_{b+1}$ and $C(u_i,u_j,b) = C_b$ for simplicity. From the definirion of $g_b(k|u_i,u_j)$, we obtain that

$$\begin{aligned}
H(k) &\stackrel{\text{def.}}{=} g_{b+1}(k|u_i,u_j) - g_b(k-1|u_i,u_j) \\
&= \textstyle\sum_{l=1}^{k} C_{b+1} l^{u_i-1}(b-l+1)^{u_j-1} - \sum_{l=1}^{k-1} C_b l^{u_i-1}(b-l)^{u_j-1} \\
&= (1 - C_{b+1}\textstyle\sum_{l=k+1}^{b} l^{u_i-1}(b-l+1)^{u_j-1}) - (1 - C_b \sum_{l=k}^{b-1} l^{u_i-1}(b-l)^{u_j-1}) \\
&= C_b \textstyle\sum_{l=k+1}^{b}(l-1)^{u_i-1}(b-l+1)^{u_j-1} - C_{b+1}\sum_{l=k+1}^{b} l^{u_i-1}(b-l+1)^{u_j-1} \\
&= \textstyle\sum_{l=k+1}^{b}(C_b(l-1)^{u_i-1}(b-l+1)^{u_j-1} - C_{b+1} l^{u_i-1}(b-l+1)^{u_j-1}) \\
&= \textstyle\sum_{l=k+1}^{b} C_b l^{u_i-1}(b-l+1)^{u_j-1}\left(\left(1-\frac{1}{l}\right)^{u_i-1} - \frac{C_{b+1}}{C_b}\right).
\end{aligned}$$

Similarly, we can also show that

$$\begin{aligned}
H(k) &= C_{b+1}\textstyle\sum_{l=1}^{k} l^{u_i-1}(b-l+1)^{u_j-1} - C_b \sum_{l=1}^{k-1} l^{u_i-1}(b-l)^{u_j-1} \\
&\geq C_{b+1}\textstyle\sum_{l=2}^{k} l^{u_i-1}(b-l+1)^{u_j-1} - C_b\sum_{l=2}^{k}(l-1)^{u_i-1}(b-l+1)^{u_j-1} \\
&= \textstyle\sum_{l=2}^{k}(C_{b+1} l^{u_i-1}(b-l+1)^{u_j-1} - C_b(l-1)^{u_i-1}(b-l+1)^{u_j-1}) \\
&= \textstyle\sum_{l=2}^{k} C_b l^{u_i-1}(b-l+1)^{u_j-1}\left(\frac{C_{b+1}}{C_b} - \left(1-\frac{1}{l}\right)^{u_i-1}\right).
\end{aligned}$$

By introducing the function $h : \{2,3,\ldots,b\} \to \mathrm{R}$ defined by $h(l) \stackrel{\text{def.}}{=} \left(1-\frac{1}{l}\right)^{u_i-1} - \frac{C_{b+1}}{C_b}$, we have the following equality and inequality

$$H(k) = \textstyle\sum_{l=k+1}^{b} C_b l^{u_i-1}(b-l+1)^{u_j-1}h(l) \tag{2}$$
$$\geq -\textstyle\sum_{l=2}^{k} C_b l^{u_i-1}(b-l+1)^{u_j-1}h(l). \tag{3}$$

(a) Consider the case that $u_i \geq 1$.

Since $u_i - 1 \geq 0$, the function $h(l)$ is monotone non-decreasing. When $h(k) \geq 0$ holds, we have $0 \leq h(k) \leq h(k+1) \leq \cdots \leq h(b)$, and so (2) implies the non-negativity $H(k) \geq 0$. If $h(k) < 0$, then inequalities $h(2) \leq h(3) \leq \cdots \leq h(k) < 0$ hold, and so (3) implies that $H(k) \geq -\sum_{l=2}^{k} C_b l^{u_i-1}(b-l+1)^{u_j-1}h(l) \geq 0$.

(b) Consider the case that $0 \leq u_i \leq 1$.

Since $u_i - 1 \leq 0$, the function $h(l)$ is monotone non-increasing. If the inequality $h(b) \geq 0$ hold, we have $h(2) \geq h(3) \geq \cdots \geq h(b) \geq 0$ and inequality (2) implies the non-negativity $H(k) \geq 0$. Thus, we only need to show that $h(b) = (\frac{b-1}{b})^{u_i - 1} - \frac{C_{b+1}}{C_b} \geq 0$.

In the rest of this proof, we substitute $u_{i'} - 1$ by $\alpha_{i'}$ for all i'. We define a function $H_0(b, \alpha_i, \alpha_j)$ by $H_0(b, \alpha_i, \alpha_j) \stackrel{\text{def.}}{=} (b-1)^{\alpha_i} C_{b+1}^{-1} - b^{\alpha_i} C_b^{-1}$. It is clear that if the condition $[-1 \leq \forall \alpha_i \leq 0,\ -1 \leq \forall \alpha_j,\ \forall b \in \{2, 3, 4, \ldots\},\quad H_0(b, \alpha_i, \alpha_j) \geq 0]$ holds, we obatin the required result that $h(b) \geq 0$ for each $b \in \{2, 3, 4, \ldots\}$. Now we transform the function $H_0(b, \alpha_i, \alpha_j)$ and obtain another expression as follows;

$$
\begin{aligned}
&H_0(b, \alpha_i, \alpha_j) = (b-1)^{\alpha_i} \sum_{k=1}^{b} k^{\alpha_i} (b-k+1)^{\alpha_j} - b^{\alpha_i} \sum_{k=1}^{b-1} k^{\alpha_i} (b-k)^{\alpha_j} \\
&= \textstyle\sum_{k=1}^{b} (b-1)^{\alpha_i} k^{\alpha_i} (b-k+1)^{\alpha_j} \frac{(b-k)+(k-1)}{b-1} - b^{\alpha_i} \sum_{k=1}^{b-1} k^{\alpha_i} (b-k)^{\alpha_j} \\
&= \textstyle\sum_{k=1}^{b-1} \Big[(b-1)^{\alpha_i} k^{\alpha_i} (b-k+1)^{\alpha_j} \left(\frac{b-k}{b-1}\right) + (b-1)^{\alpha_i} (k+1)^{\alpha_i} (b-k)^{\alpha_j} \left(\frac{k}{b-1}\right) \\
&\qquad\qquad - b^{\alpha_i} k^{\alpha_i} (b-k)^{\alpha_j} \Big] \\
&= \textstyle\sum_{k=1}^{b-1} \frac{(b-1)^{\alpha_i} k^{\alpha_i} (b-k)^{\alpha_j}}{b-1} \left[\left(1 + \frac{1}{b-k}\right)^{\alpha_j} (b-k) + \left(1 + \frac{1}{k}\right)^{\alpha_i} k - \left(\frac{b}{b-1}\right)^{\alpha_i} (b-1) \right].
\end{aligned}
$$

Then it is enough to show that the function

$$
H_1(b, \alpha_i, \alpha_j, k) \stackrel{\text{def.}}{=} \left(1 + \tfrac{1}{b-k}\right)^{\alpha_j} (b-k) + \left(1 + \tfrac{1}{k}\right)^{\alpha_i} k - \left(\tfrac{b}{b-1}\right)^{\alpha_i} (b-1)
$$

is nonnegative for any $k \in \{1, 2, \ldots, b-1\}$. Since $1 + 1/(b-k) > 1$ and $\alpha_j \geq -1$, we have

$$
H_1(b, \alpha_i, \alpha_j, k) \geq H_1(b, \alpha_i, -1, k) = \tfrac{(b-k)^2}{b-k+1} + \left(1 + \tfrac{1}{k}\right)^{\alpha_i} k - \left(\tfrac{b}{b-1}\right)^{\alpha_i} (b-1).
$$

We differentiate the function H_1 by α_i, and obtain the following

$$
\begin{aligned}
\frac{\partial}{\partial \alpha_i} H_1(b, \alpha_i, -1, k) &= \left(1 + \tfrac{1}{k}\right)^{\alpha_i} k \log\left(1 + \tfrac{1}{k}\right) - \left(\tfrac{b}{b-1}\right)^{\alpha_i} (b-1) \log\left(\tfrac{b}{b-1}\right) \\
&= \left(1 + \tfrac{1}{k}\right)^{\alpha_i} \log\left(1 + \tfrac{1}{k}\right)^{k} - \left(1 + \tfrac{1}{b-1}\right)^{\alpha_i} \log\left(1 + \tfrac{1}{b-1}\right)^{(b-1)}.
\end{aligned}
$$

Since k, b is a pair of positive integers satisfying $1 \leq k \leq b-1$, the non-positivity of α_i implies $0 \leq (1 + 1/k)^{\alpha_i} \leq (1 + 1/(b-1))^{\alpha_i}$ and $0 \leq \log(1 + 1/k)^k \leq \log(1 + 1/(b-1))^{b-1}$. Thus the function $H_1(b, \alpha_i, -1, k)$ is monotone non-decreasing with respect to $\alpha_i \leq 0$. Thus we have

$$
\begin{aligned}
H_1(b, \alpha_i, -1, k) \geq H_1(b, 0, -1, k) &= \tfrac{(b-k)^2}{b-k+1} + \left(1 + \tfrac{1}{k}\right)^{0} k - \left(\tfrac{b}{b-1}\right)^{0} (b-1) \\
&= \tfrac{(b-k)^2}{b-k+1} + k - b + 1 = \tfrac{(b-k)^2 + 1^2 - (b-k)^2}{b-k+1} = \tfrac{1}{b-k+1} \geq 0.
\end{aligned}
$$

□

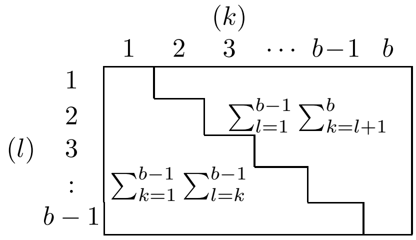
Lemma 6. $\forall b \in \{2,3,\ldots\}$, $\forall u_i \geq \forall u_j$,

$$\sum_{k=1}^{b-1}[g_b(k|u_i,u_j) - g_{b+1}(k|u_i,u_j)] - \sum_{k=1}^{b-1}[g_{b+1}(k|u_i,u_j) - g_b(k-1|u_i,u_j)] \geq 0.$$

Proof: We denote $C(u_i,u_j,b+1) = C_{b+1}$ and $C(u_i,u_j,b) = C_b$ for simplicity. It is not difficult to show the following equalities,

$$\begin{aligned}
G &\overset{\text{def.}}{=} \textstyle\sum_{k=1}^{b-1}[g_b(k|u_i,u_j) - g_{b+1}(k|u_i,u_j)] - \sum_{k=1}^{b-1}[g_{b+1}(k|u_i,u_j) - g_b(k-1|u_i,u_j)] \\
&= \textstyle\sum_{k=1}^{b-1} g_b(k|u_i,u_j) - \sum_{k=1}^{b-1} g_{b+1}(k|u_i,u_j) \\
&\quad \textstyle- \sum_{k=1}^{b-1} g_{b+1}(k|u_i,u_j) + \sum_{k=1}^{b-1} g_b(k-1|u_i,u_j) \\
&= \textstyle\sum_{k=1}^{b-1} g_b(k|u_i,u_j) - \sum_{k=1}^{b-1} g_{b+1}(k|u_i,u_j) \\
&\quad \textstyle- \sum_{k=1}^{b-1} g_{b+1}(k|u_i,u_j) + \sum_{k=2}^{b-1} g_b(k-1|u_i,u_j) \\
&= \textstyle\sum_{k=1}^{b-1} C_b \sum_{l=1}^{k} l^{u_i-1}(b-l)^{u_j-1} - \sum_{k=1}^{b-1} C_{b+1} \sum_{l=1}^{k} l^{u_i-1}(b-l+1)^{u_j-1} \\
&\quad \textstyle- \sum_{k=1}^{b-1} C_{b+1} \sum_{l=1}^{k} l^{u_i-1}(b-l+1)^{u_j-1} + \sum_{k=2}^{b-1} C_b \sum_{l=1}^{k-1} l^{u_i-1}(b-l)^{u_j-1} \\
&= \textstyle C_b \sum_{l=1}^{b-1}(b-l)l^{u_i-1}(b-l)^{u_j-1} - C_{b+1}\sum_{l=1}^{b-1}(b-l)l^{u_i-1}(b-l+1)^{u_j-1} \\
&\quad \textstyle -C_{b+1}\sum_{l=1}^{b-1}(b-l)l^{u_i-1}(b-l+1)^{u_j-1} + C_b \sum_{l=1}^{b-1}(b-l-1)l^{u_i-1}(b-l)^{u_j-1} \\
&= \textstyle C_b \sum_{l=1}^{b-1}(2b-2l-1)l^{u_i-1}(b-l)^{u_j-1} - C_{b+1}\sum_{l=1}^{b-1}(2b-2l)l^{u_i-1}(b-l+1)^{u_j-1} \\
&= \textstyle C_b C_{b+1}\Big(C_{b+1}^{-1}\sum_{l=1}^{b-1}(2b-2l-1)l^{u_i-1}(b-l)^{u_j-1} \\
&\qquad\qquad \textstyle -C_b^{-1}\sum_{l=1}^{b}(2b-2l)l^{u_i-1}(b-l+1)^{u_j-1}\Big) \\
&= \textstyle C_b C_{b+1}\Big(\sum_{k=1}^{b} k^{u_i-1}(b-k+1)^{u_j-1}\sum_{l=1}^{b-1}(2b-2l-1)l^{u_i-1}(b-l)^{u_j-1} \\
&\qquad\qquad \textstyle -\sum_{k=1}^{b-1} k^{u_i-1}(b-k)^{u_j-1}\sum_{l=1}^{b}(2b-2l)l^{u_i-1}(b-l+1)^{u_j-1}\Big) \\
&= \textstyle C_b C_{b+1}\Big(\sum_{k=1}^{b}\sum_{l=1}^{b-1}(2b-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
&\qquad\qquad \textstyle -\sum_{k=1}^{b-1}\sum_{l=1}^{b}(2b-2l)(kl)^{u_i-1}((b-k)(b-l+1))^{u_j-1}\Big) \\
&= \textstyle C_b C_{b+1}\Big(\sum_{k=1}^{b}\sum_{l=1}^{b-1}(2b-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
&\qquad\qquad \textstyle -\sum_{k=1}^{b}\sum_{l=1}^{b-1}(2b-2k)(kl)^{u_i-1}((b-l)(b-k+1))^{u_j-1}\Big) \\
&= \textstyle C_b C_{b+1}\sum_{k=1}^{b}\sum_{l=1}^{b-1}\left((2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1}\right) \\
&= \textstyle \frac{C_b C_{b+1}}{2}\Big(\sum_{k=1}^{b}\sum_{l=1}^{b-1}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
&\qquad\qquad \textstyle +\sum_{k=1}^{b}\sum_{l=1}^{b-1}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1}\Big) \\
&= \textstyle \frac{C_b C_{b+1}}{2}\Big(\sum_{k=1}^{b}\sum_{l=1}^{b-1}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
&\quad \textstyle +\sum_{k=1}^{b}\sum_{l=1}^{b-1}(2(b-k+1)-2(b-l)-1) \\
&\qquad ((b-k+1)(b-l))^{u_i-1}((b-(b-k+1)+1)(b-(b-l)))^{u_j-1}\Big)
\end{aligned}$$

$$
\begin{aligned}
= \ & \tfrac{C_b C_{b+1}}{2} \Big(\textstyle\sum_{k=1}^{b} \sum_{l=1}^{b-1} (2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& \qquad - \textstyle\sum_{k=1}^{b} \sum_{l=1}^{b-1} (2k-2l-1)((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \Big)
\end{aligned}
$$



$$
\begin{aligned}
= \ & \tfrac{C_b C_{b+1}}{2} \Big(\textstyle\sum_{l=1}^{b-1} \sum_{k=l+1}^{b} (2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& \qquad + \textstyle\sum_{k=1}^{b-1} \sum_{l=k}^{b-1} (2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& \qquad - \textstyle\sum_{l=1}^{b-1} \sum_{k=l+1}^{b} (2k-2l-1)((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \\
& \qquad - \textstyle\sum_{k=1}^{b-1} \sum_{l=k}^{b-1} (2k-2l-1)((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \Big) \\
= \ & \tfrac{C_b C_{b+1}}{2} \Big(\textstyle\sum_{l=1}^{b-1} \sum_{k=l+1}^{b} (2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& \qquad - \textstyle\sum_{l=1}^{b-1} \sum_{k=l+1}^{b} (2k-2l-1)(l(k-1))^{u_i-1}((b-l+1)(b-k+1))^{u_j-1} \\
& \qquad - \textstyle\sum_{l=1}^{b-1} \sum_{k=l+1}^{b} (2k-2l-1)((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \\
& \qquad + \textstyle\sum_{l=1}^{b-1} \sum_{k=l+1}^{b} (2k-2l-1)((b-l+1)(b-k+1))^{u_i-1}(l(k-1))^{u_j-1} \Big) \\
= \ & \frac{C_b C_{b+1}}{2} \sum_{l=1}^{b-1} \sum_{k=l+1}^{b} (2k-2l-1) \Big((kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& \qquad -(l(k-1))^{u_i-1}((b-l+1)(b-k+1))^{u_j-1} \\
& \qquad -((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \\
& \qquad +((b-l+1)(b-k+1))^{u_i-1}(l(k-1))^{u_j-1} \Big).
\end{aligned}
$$

We define a function $G_0(k,l,u_i,u_j)$ by

$$
G_0(k,l,u_i,u_j) \stackrel{\text{def.}}{=} \begin{pmatrix} (kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\ -(l(k-1))^{u_i-1}((b-l+1)(b-k+1))^{u_j-1} \\ -((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \\ +((b-l+1)(b-k+1))^{u_i-1}(l(k-1))^{u_j-1} \end{pmatrix}.
$$

Since $1 \le l < l+1 \le k \le b$, it is clear that $(2k-2l-1) > 0$. Thus, we only need to show that $\forall l \in \{1,2,\ldots,b-1\}$, $\forall k \in \{2,3,\ldots,b\}$, $\forall u_i \ge \forall u_j$, $G_0(l,k,u_i,u_j) \ge 0$. It is easy to see that

$$
\begin{aligned}
& G_0(k,l,u_i,u_j) \\
& = \left(\begin{array}{l} (l(k-1))^{u_i-1}(b-k+1)^{u_j-1} \left((1+\frac{1}{k-1})^{u_i-1}(b-l)^{u_j-1} - (b-l+1)^{u_j-1} \right) \\ +((b-k+1)(b-l))^{u_i-1} l^{u_j-1} \left(-k^{u_j-1} + (1+\frac{1}{b-l})^{u_i-1}(k-1)^{u_j-1} \right) \end{array} \right).
\end{aligned}
$$

Then it is clear that $G_0(k,l,u_i,u_j)$ is non-decreasing with respect to u_i and so $G_0(k,l,u_i,u_j) \ge G_0(k,l,u_j,u_j)$. By substituting u_i by u_j in the definition of $G_0(k, l, u_i, u_j)$, it is easy to see that $G_0(k, l, u_j, u_j) = 0$. Thus we have the desired result. □

References

1. Bubley, R., M. Dyer, M.: Path coupling: A technique for proving rapid mixing in Markov chains, 38th Annual Symposium on Foundations of Computer Science, IEEE, San Alimitos, 1997, 223–231.
2. Bubley, R.: Randomized Algorithms : Approximation, Generation, and Counting, Springer-Verlag, New York, 2001.
3. Dimakos, X. K.: A guide to exact simulation, International Statistical Review, 69 (2001), pp. 27–48.
4. Durbin, R., Eddy, R., Krogh, A., Mitchison, G.: Biological sequence analysis: probabilistic models of proteins and nucleic acids, Cambridge Univ. Press, 1998.
5. Kijima, S. and Matsui, T.,: Polynomial Time Perfect Sampling Algorithm for Two-rowed Contingency Tables, METR 2003-15, Mathematical Engineering Technical Reports, University of Tokyo, 2003. (available from http://www.keisu.t.u-tokyo.ac.jp/Research/techrep.0.html)
6. Matsui, T., Motoki, M., and Kamatani, N.,: Polynomial Time Approximate Sampler for Discretized Dirichlet Distribution, METR 2003-10, Mathematical Engineering Technical Reports, University of Tokyo, 2003. (available from http://www.keisu.t.u-tokyo.ac.jp/Research/techrep.0.html)
7. Niu, T., Qin, Z. S., Xu, X., Liu, J. S.: Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms, Am. J. Hum. Genet., **70** (2002) 157–169.
8. Pritchard, J. K., Stephens, M., Donnely, P.: Inference of population structure using multilocus genotype data, Genetics, **155** (2000) 945–959.
9. Propp, J. and Wilson, D.: Exact sampling with coupled Markov chains and applications to statistical mechanics, Random Structures and Algorithms, 9 (1996), pp. 232–252.
10. Propp, J. and Wilson, D.: 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.
11. Robert, C. P.: The Bayesian Choice, Springer-Verlag, New York, 2001.
12. Wilson, D.: How to couple from the past using a read-once source of randomness, Random Structures and Algorithms, 16 (2000), pp. 85–113.