Rapidly Mixing Multiple-try Metropolis Algorithms for Model Selection Problems

Anonymous Author(s)
Affiliation
Address
email

Abstract

The Multiple-try Metropolis (MTM) algorithm is an extension of the Metropolis-Hastings (MH) algorithm by selecting the proposed state among multiple trials according to some weight function. Although MTM has gained great popularity owing to its faster empirical convergence and mixing than the standard MH algorithm, its theoretical mixing property is rarely studied in the literature due to its complex proposal scheme. We prove that MTM can achieve a mixing time bound smaller than that of MH by a factor of the number of trials under a general setting applicable to high-dimensional model selection problems. Our theoretical results motivate a new class of weight functions and guide the choice of the number of trials, which leads to improved performance than standard MTM algorithms. We support our theoretical results by extensive simulation studies with several Bayesian model selection problems: variable selection, stochastic block models, and spatial clustering models.

1 Introduction

Markov chain Monte Carlo (MCMC) method has become a powerful and standard tool for Bayesian posterior computation. In particular, the Metropolis–Hastings (MH) algorithm is widely employed in many statistical and machine learning models to sample from posterior distributions with intractable normalizing constants. A prominent usage example of MH is the vast class of Bayesian model selection problems with discrete-valued high-dimensional parameters, whose posterior probabilities are infeasible to evaluate because the size of model space potentially grows (super-)exponentially with the number of parameters. Examples of Bayesian model selection problems and their applications are ubiquitous, including Bayesian variable selection (BVS) [33, and references therein], stochastic block model (SBM) [17, 26], Bayesian structure learning [23, 13], change-point detection model [4, 9], spatial clustering models [20, 22], and many others.

Despite the popularity of MCMC algorithms, their convergence rate analysis is often challenging. Mixing time is a key concept of interest when analyzing the convergence of MCMC, which defines the number of iterations needed to converge to a stationary distribution within a small total variation distance [19]. We say a chain is *rapidly mixing* if the mixing time grows at most polynomially in some complexity parameters. Mixing time bound has been studied extensively for problems such as card shuffling [3], random walks on groups [2], graph colorings [18], and the Ising models [12], across the fields of probability, physics, and computer science [30]. Although there is a rich literature on discrete-state-space mixing time analysis in many areas, the focus has been primarily on approximate counting and random generation, but only recently has research on the MCMC convergence in the context of statistical model selection begun to emerge. [35] proved rapid mixing of an MH algorithm in a high-dimensional BVS problem, and [29] improved the bound with some modification. Rapid mixing results of MH algorithms are established for Bayesian community detection problem with SBM [41], and for learning equivalence class in a high-dimensional Bayesian structure learning [39]. Recently, locally informed MH algorithms have been developed to incorporate local information

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

about the target distribution. [36] proposed a locally balanced proposal and proved their optimality in terms of Peskun ordering [28] for discrete space models without providing mixing time rate analysis. [40] also proposed an informed MCMC and proved a dimension-free mixing time bound but only for a BVS problem [40].

The multiple-try Metropolis (MTM) algorithm is an extension of the MH algorithm, which uses a weight function to choose the proposed state among randomly sampled trials from a proposal kernel and therefore easily enables parallelization. The variety of weight functions and the number of trials enrich the algorithm class but result in a much more complex transition probability matrix than standard MH algorithms, hampering the theoretical development on the mixing time of MTM. As a result, there is a lack of theoretical guidance for practitioners on the choices of weight functions and number of trials in MTM and also the choice between MTM and other MCMC algorithms. Only very recently, [34] considered the MTM independence sampler (MTM-IS), a simple special case of MTM where the proposal kernel does not depend on the current state. They provided the convergence rate analysis of MTM-IS that shows it is less efficient than the simpler approach of repeated Metropolised independent sampling at the same computational cost, but did not discuss the choices of weight functions and number of trials. As acknowledged in [34], MTM is significantly different from MTM-IS, and its convergence rate is more challenging to analyze. In this paper, we establish the mixing time bound of MTM for general model selection problems. Our main contributions are the following:

1) We show that MTM with existing popular choices of weight functions can have mixing issues and then propose a new class of weight functions.
2) With the new weight functions, we prove that the mixing time bound of the MTM algorithm is smaller than that of the MH algorithm by a factor of the number of trials in a general model selection setting.
3) We provide theoretical justification for the counterintuitive phenomenon that an increase in the number of trials N may not always lead to a proportionate improvement in mixing, and suggest a theoretically guided algorithm to choose N for practical use.
4) We validate our theoretical findings via extensive simulation studies with various model selection problems: Bayesian variable selection, stochastic block models, and spatial clustering models.

2 Preliminary

2.1 Notation

For a positive integer $m \in \mathbb{N}$, let $[m] = \{1, \ldots, m\}$ and $\lfloor m \rfloor$ be the largest integer no greater than m. Let $|\cdot|$ denote the cardinality of a set. Let $\mathcal{X} = \mathcal{X}_p$ denote a finite state space where $|\mathcal{X}|$ grows in a complexity parameter $p \in \mathbb{N}$. Define $\mathbf{P}$ as a transition probability of an irreducible, aperiodic and reversible Markov chain. We call a probability distribution π on $\mathcal{X}$ a stationary or target distribution of a chain $\mathbf{P}$ if it satisfies $\sum_{z \in \mathcal{X}} \pi(z)\mathbf{P}(z, x) = \pi(x)$ for all $x \in \mathcal{X}$. In Bayesian inference, π is the posterior distribution. Let $\mathcal{N} \colon \mathcal{X} \to 2^{\mathcal{X}}$ be a set-valued map called a neighborhood relation, which maps a state to a set of states. We say the neighborhood relation $\mathcal{N}$ is symmetric if $x' \in \mathcal{N}(x)$ iff $x \in \mathcal{N}(x')$ for all $x \neq x'$. We define the random walk proposal kernel as $\mathbf{K}_{\mathrm{RW}}(x, y) = \mathbb{1}_{\mathcal{N}(x)}(y)/|\mathcal{N}(x)|$ where $\mathbb{1}_{\mathcal{N}(x)}$ denotes the indicator function of $\mathcal{N}(x)$. We denote a graph by (V, E) where V is the vertex set and $E \subset V \times V$ is the set of edges. We say $\delta(x, y) = (v_0, \ldots, v_m)$ is a E-path (with length m) from x to y if $e_i = (v_{i-1}, v_i) \in E$ for $i \in [m]$, $v_0 = x, v_m = y$, and all vertices are distinct.

2.2 Multiple-try Metropolis algorithm

We present the MTM algorithm [21] in Algorithm 1. In MTM with random walk proposal kernel $\mathbf{K}_{\mathrm{RW}}$, N trial states are sampled from the proposal kernel $\mathbf{K}_{\mathrm{RW}}(x, \cdot)$, and then the proposed state y is selected among the trials using the weight function

$$w(y \,|\, x) = \pi(y)\mathbf{K}_{\mathrm{RW}}(y, x)\lambda(y, x), \tag{1}$$

where $\lambda(y, x)$ can be any symmetric function that is positive whenever $\mathbf{K}_{\mathrm{RW}}(y, x) > 0$. Note that Algorithm 1 becomes the MH algorithm when the number of trials N is one.

Algorithm 1: Multiple-try Metropolis (MTM) algorithm with proposal kernel $\mathbf{K}_{\mathrm{RW}}$

Input: An initial state x_0, a neighborhood relation $\mathcal{N}$, the number of trials N, a weight function w defined in (1), the number of Markov chain iterations T

for $t = 0, \ldots, T-1$ **do**

Step 1. Draw $y_1, \ldots, y_N$ uniformly at random from $\mathcal{N}(x_t)$, and compute $w(y_j \mid x_t)$ for $j = 1 \ldots, N$.

Step 2. Select $j \in [N]$ with probability proportional to $w(y_j \mid x_t)$ and define $y = y_j$.

Step 3. Sample $x_1^\star, \ldots, x_{N-1}^\star$ uniformly at random from $\mathcal{N}(y)$ and define

$$\alpha = \min\left\{1, \frac{w(y \mid x_t) + \sum_{l=1, l\neq j}^{N} w(y_l \mid x_t)}{w(x_t \mid y) + \sum_{l=1}^{N-1} w(x_l^\star \mid y)}\right\}. \tag{2}$$

Step 4. With probability α, accept y and let $x_{t+1} = y$, otherwise let $x_{t+1} = x_t$.

Output: A set of Markov chain samples $\{x_t\}_{t=1}^T$

There have been efforts to generalize Algorithm 1 [10, 8]. The work of [27] extends the weight function class by proposing a general form of acceptance probability to achieve the reversibility of a chain when the weight function does not belong to (1). While one can easily find a weight function satisfying (1), there is little research on the most appropriate form of weight functions. [21] investigated some particular class of weight functions and claimed the performance is insensitive to the choice of weight functions within this class; $w(y \mid x) = \pi(y)$ (when $\mathbf{K}_{\mathrm{RW}}$ is symmetric) and $w(y \mid x) = \pi(y)/\mathbf{K}_{\mathrm{RW}}(x, y)$ are popular choices in this class [5, 25].

Another important component of the MTM method is the number of trials N. It has been reported with empirical evidence that an increase in the number of trials N does not necessarily result in a corresponding improvement in mixing [24]. They illustrate a counterintuitive phenomenon that the acceptance probability for a proposal move to higher probability regions can be extremely low. As a solution, they suggest randomly selecting the number of trials N at each iteration. Despite the anecdotal observation that certain choices of N make the algorithm less efficient, the principled way to choose N is rarely discussed in the literature.

2.3 Analysis on mixing time via geometric tools

A theoretical foundation for employing MCMC methods to sample from a posterior distribution on a finite state space is the well-known convergence theorem, which states that an irreducible and aperiodic Markov chain with a stationary distribution converges to the stationary distribution regardless of the initial distribution [19, Theorem 4.9]. However, since it is impossible to simulate a chain infinitely long, our main interest is to analyze how well the chain approximates the target distribution after a certain number of steps. To this end, we introduce ϵ-mixing time $t_{\mathrm{mix}}(\epsilon) = \max_{x\in\mathcal{X}} \min\left\{t \in \mathbb{N} \colon \|\mathbf{P}_{\mathrm{lazy}}^t(x, \cdot) - \pi\|_{\mathrm{TV}} \leq \epsilon\right\}$, where $\|\cdot\|_{\mathrm{TV}}$ is the total variation distance, ϵ is a small constant, and $\mathbf{P}_{\mathrm{lazy}} = (\mathbf{P} + \boldsymbol{I})/2$ denotes the lazy version of $\mathbf{P}$. In words, t_{mix} quantifies the necessary number of steps for the chain to have a small distance from the stationary distribution. Among several techniques for characterizing the mixing property [7, 31, 19, 15], a path method is employed to derive our main result [11, 32]. Intuitively, the mixing time depends on how well the states "communicate", especially among those which have high values of π. In contrast to samplers defined on a continuous state space with a Euclidean topology, posterior landscape and modality on a finite state space are more difficult to envision since it highly depends on how one defines the neighborhood relation. Given a transition kernel $\mathbf{P}$, we define the neighborhood of a state x by $\mathcal{N}(x) = \{x' \in \mathcal{X} : \mathbf{P}(x, x') > 0\}$, the set of reachable states by one-step transition from x, an we say x is a mode if $\pi(x) > \max_{x'\in\mathcal{N}(x)} \pi(x')$. This specification of the neighborhood enables us to define a graph $(\mathcal{X}, E)$ where $E = \{(x, x') \in \mathcal{X} \times \mathcal{X} : x' \in \mathcal{N}(x)\}$ is an (undirected since we assume reversibility of $\mathbf{P}$) edge set. For each $(x, y) \in \mathcal{X} \times \mathcal{X}$, we choose only one E-path $\delta(x, y)$, and define a path ensemble by a collection of the paths $\Delta = \{\delta(x, y) : (x, y) \in \mathcal{X} \times \mathcal{X}\}$. A path ensemble is a configuration of path network that helps quantify the quality of the communication within the state space $\mathcal{X}$. The following proposition, which serves as a base of our main result, gives a mixing time bound by means of a path ensemble.

Proposition 1 ([32]). *For any path ensemble Δ,*

$$t_{\text{mix}}(\epsilon) \leq 2\rho(\Delta)\ell(\Delta)\left[\log(1/\epsilon) + \log\left\{\min_{x\in\mathcal{X}} \pi(x)\right\}^{-1}\right],$$

where $\ell(\Delta) = \max_{x,y} |\delta(x,y)|$ and

$$\rho(\Delta) = \max_{(u,v)\in E} \frac{1}{\pi(u)\mathbf{P}(u,v)} \sum_{\substack{x,y\in\mathcal{X} \\ (u,v)\in\delta(x,y)}} \pi(x)\pi(y) \tag{3}$$

For completeness, we provide a proof of Proposition 1 with a detailed discussion in Appendix A.1. The quantity $\rho(\Delta)$ is called the *congestion parameter*, and $\ell(\Delta)$ is the length of the longest path in Δ. To interpret $\rho(\Delta)$ and $\ell(\Delta)$, some terminologies from the transportation network in graph theory are often useful. Consider $\pi(u)\mathbf{P}(u,v)$ as the capacity of an edge (u,v), and $\pi(x)\pi(y)$ as the unit flow of a path $\delta(x,y)$. Then $\rho(\Delta)$ quantifies the maximum loading of an edge under the configuration Δ. If one edge with a small capacity is traversed by a huge number of paths with a high unit flow, it results in a constriction. Consider $\ell(\Delta)$ as the diameter of the configuration Δ. If the diameter is extremely large, it causes inefficient communication, especially among the states lying on the margin of the configuration. If it is possible to design the path system in a way that no edge is overloaded by paths and no two states are too far apart, then the chain can have a good mixing behavior. One of the key steps in the proof of the main result is identifying the path ensemble Δ^* (see Appendix A.3) which makes the mixing bound in (6) tight.

3 Main result

In this section, we prove the complexity of mixing time of the MTM algorithm for general model selection problems on a finite model space $\mathcal{X}$, where we assume that there is only one state with the highest posterior mass, denoted by x^*. Throughout the section, we may assume that $|\mathcal{X}|$ grows with the complexity parameter p (e.g., the number of variables in BVS), and the posterior distribution π and other related objects are implicitly parameterized by p.

3.1 A mixing time bound with a new class of weight functions

While extending the class of weight functions has been a main focus in the literature [27, 25], the choice of weight function is rarely discussed. We introduce a class of weight functions by using a function $h\colon \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ that satisfies $h(u) = uh(1/u)$ for all $u > 0$; such a function h is called a *balancing function* [36]. The class of balancing functions is very broad since we can define a balancing function from an arbitrary positive function $h_0(u)$ for $u \geq 1$ by applying the balancing rule to the value of $h_0(u)$ for $0 < u < 1$. Typical choices of balancing functions include $\sqrt{u}$, $\min\{1,u\}$, $\max\{1,u\}$, $u/(u+1)$, and $u+1$. The following proposition states that Algorithm 1 with the proposed balancing weight function induces a reversible Markov chain with π as its stationary distribution.

Proposition 2. *Suppose that a weight function is defined as*

$$w(y\,|\,x) = h\left(\frac{\pi(y)\mathbf{K}_{\text{RW}}(y,x)}{\pi(x)\mathbf{K}_{\text{RW}}(x,y)}\right), \tag{4}$$

where h is a balancing function. Then, a Markov chain $\mathbf{P}_{\text{MTM}}$ induced by Algorithm 1 has a stationary distribution π.

The proof is deferred to Appendix A.2. Compared with the popular choice of weight function in the existing MTM literature such as $w(y\,|\,x) = \pi(y)$, the weight function we propose in (2) can be viewed as a transformed and scaled version with respect to the posterior probability at current state, e.g., $w(y\,|\,x) = \sqrt{\pi(y)/\pi(x)}$ if $h(u) = \sqrt{u}$ and a symmetric proposal kernel is used. Before providing a rigorous theorem on the mixing time bound of MTM under our new choice of $w(y\,|\,x)$, we give an intuitive explanation for why the existing unscaled weight function may have poor mixing properties. First, it is reasonable to assume that the true generating model, x^*, will have the largest posterior mass when the posterior contraction or the model selection consistency holds asymptotically.

Indeed, such results have been established for a variety of model selection problems [35, 39, 41]. On the other hand, model spaces are often endowed with a natural neighborhood relation such that the degree of "model fit" (measured by goodness of fit and penalization on model size) between neighboring models tends to have small difference. This implies that a small modification of a model may lead to a marginal change in the posterior probability. For example, in a Bayesian variable selection problem with p number of variables, the model space can be represented as a set of binary vectors $\gamma \in \{0,1\}^p$ where $\gamma_j = 1$ (resp. $\gamma_j = 0$) indicates j-th variable is included (resp. excluded) in the model. Let us assume that γ^* is the true data generating model and has the highest posterior, and define a Hamming distance $d_{\mathrm{H}}(\gamma, \gamma') = \sum_{j=1}^{p} \mathbb{1}\{\gamma_j \neq \gamma'_j\}$ for $\gamma, \gamma' \in \{0,1\}^p$. One often defines the neighborhood of γ as the set of states with a small Hamming distance to γ [16, 35, 40], and models in the neighborhood tend to have similar posterior probabilities. A state with a large Hamming distance from γ^* usually has small posterior probability because its structure is too different from the true data generating model. Figure 1 depicts such relationship between posterior probabilities and structural similarities among the models. Given that γ is the current state, Step 2 of Algorithm 1 tends to propose a neighboring state γ' with a higher posterior probability, so γ' tends to be closer to γ^* in Hamming distance, i.e. $d_{\mathrm{H}}(\gamma', \gamma^*) < d_{\mathrm{H}}(\gamma, \gamma^*)$. On average, the Hamming distance between a neighboring state of γ' and γ^* will be smaller than that between a neighboring state of γ with γ^*. Therefore, the expected value of posterior probability of a neighboring state of γ' (in the denominator part in (2)) will be smaller than that of a neighboring state of γ (in the numerator part in (2)). If we resort to the weight function $w(\gamma' \,|\, \gamma) = \pi(\gamma')$, this can cause a significantly low acceptance probability, especially when the number of trials is large (due to the law of large numbers, the acceptance ratio converges to the ratio of average posterior probabilities of the two neighborhoods); see also Appendix A.4 for a concrete example. Our suggested weight function class in (4), however, is able to mitigate this counterintuitive phenomenon by "scaling" the weight function with respect to the current state. We shall demonstrate the effectiveness of the class of weight functions (4) in the simulation studies.

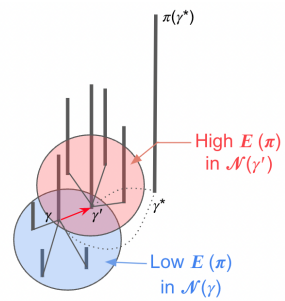

Figure 1: An illustration of the reason that the weight function $w(y \,|\, x) = \pi(y)$ may fail. Each vertical bar represents the posterior mass on the corresponding state, and thin lines connect neighboring states. The expected value of posterior probability of the neighborhood of γ' (red) is greater than that of γ (blue).

In order for the MTM procedure to efficiently sample states from a high posterior region, it is sensible to assign a larger weight to those states with a higher posterior probability in the proposal, i.e. Step 2 of Algorithm 1. To this end, we only require h to be non-decreasing. Now we introduce the main result, which states that the mixing time bound can be improved linearly in the number of trials N for our locally balanced weighting scheme.

Theorem 1. *Let $\mathbf{P}_{\mathrm{MTM}}$ denote a Markov chain induced by Algorithm 1 with a weight function (4) with some non-decreasing balancing function h. Define*

$$\mathcal{S}(x) = \{x' \in \mathcal{N}(x) \colon \pi(x')/\pi(x) > p^{t_1}\}, \tag{5}$$

for some constant t_1 and $s_0 = \max_{x \in \mathcal{X}} |\mathcal{S}(x)|$. Suppose that the following conditions hold with $p \geq 2$ and $t_2, t_3, t_4 \geq 0$ being some constants that satisfy $t_1 < t_2$, $t_3 \leq t_4 < t_2$.

(i) There exists $x' \in \mathcal{N}(x)$ such that $\pi(x')/\pi(x) \geq p^{t_2}$ for every $x \neq x^*$.

(ii) For any $x \in \mathcal{X}$, $p^{t_3} \leq |\mathcal{N}(x)| \leq p^{t_4}$.

(iii) $N = o\left(\min\left\{\frac{h(p^{t_2+t_3-t_4})}{h(p^{t_1+t_4-t_3})}, \frac{p^{t_3}}{s_0}\right\}\right)$

Then, we have the ϵ-mixing time of $\mathbf{P}_{\mathrm{MTM}}$

$$t_{\mathrm{mix}}(\epsilon) = O\left(\frac{1}{N}(p^{-t_4} - p^{-t_2})^{-1} \ell(\Delta^*) \log\left(\min_{x \in \mathcal{X}} \pi(x)\right)^{-1}\right), \tag{6}$$

where Δ^* is a path ensemble defined in Appendix A.3.

The proof is deferred to Appendix A.3. Condition (i) requires the posterior distribution to be unimodal with the peak at the model x^*. We emphasize that the unimodality is with respect to the neighborhood relation $\mathcal{N}$ and is usually satisfied with some appropriate choice of large enough neighborhood. For example, for Bayesian variable selection, it is shown in [35] that their proposed posterior may be multimodal if a neighborhood is defined as a set of models within 1-Hamming distance, whereas the 2-Hamming distance neighborhood satisfies Condition (i) with $t_2 = 2$. Similarly, the problem setting of community detection in [41] and that of structure learning in [39] satisfy (i) with some positive constant t_2. Next, Condition (ii) states that the neighborhood size $|\mathcal{N}(x)|$ should neither be too huge nor vary much from state to state. The purpose of this condition is to control the ratio $\mathbf{K}_{\mathrm{RW}}(y,x)/\mathbf{K}_{\mathrm{RW}}(x,y)$ in (4) so that the posterior ratio $\pi(y)/\pi(x)$ dominates the term. Last but not least, Condition (iii) implies that the number of trials N should not be arbitrarily large. This is consistent with the empirical observation of [24] that additional trials do not always result in better mixing. Condition (iii) also implies that $s_0 = o(p^{t_3})$ since $N \geq 1$. This seems to be a strong condition, since the number of neighboring states with a relatively large posterior probability with respect to the current state can be very large if we consider a state with the smallest posterior that is surrounded by states with greater posterior. Yet, it is a common practice to restrict the support of the prior to rule out unrealistic models by imposing some high-dimensional regularity assumptions, such as sparsity. For example, [35] introduced the parameter of the maximum number of important covariates in variable selection, and [39] considered the maximum in-degree and out-degree in structure learning to restrict the model space. In the same context, [41] suggested the use of a feasible set for the initial partition in the community detection problem. In this regard, we may consider $\mathcal{X}$ as the restricted space so that s_0 can be controlled. Most importantly, Condition (iii) provides a key idea on how to select the number of trial N, which will be discussed in Section 3.2.

Remark 1 (Problem-specific applications). Theorem 1 can be used to prove rapid mixing of the MTM algorithm for Bayesian variable selection [35] and structure learning [39] with an order N improvement of the mixing bound, given that the choice of N satisfies the condition (iii).

Remark 2 (Computational benefits). Since MTM with N trials requires calculating $2N-1$ weight functions at each iteration, improving the mixing time bound by the factor of N leads to the same overall computational complexity as the single-try MH algorithm until the convergence of Markov chain. However, compared to the single-try MH with N times longer sequential iterations, calculation of N weight functions can be efficiently done in parallel. Further, for some model selection problems, calculating multiple weight functions can often be converted into a serial of matrix multiplication problems, and optimized linear algebra libraries such as BLAS [6] can be used to exploit multiple execution units and pipe-lining. Table 1 in our simulation study results shows that MTM with moderate choice of N significantly reduces the computation time until the convergence of Markov chain.

3.2 Choice of the number of trials

Motivated by Condition (iii) in Theorem 1, we propose an algorithm to choose the number of trials N. For conciseness, we shall only focus on illustrating the case when we have balancing function $h(u) = \sqrt{u}$ and symmetric proposals (known $t_3 = t_4$) so that Condition (iii) becomes $N = o\left(\min\{p^{(t_2-t_1)/2}, p^{t_3}/s_0\}\right)$, although the algorithm can be easily generalized to other choices of h and non-symmetric proposals.

Algorithm 2: The choice of the number of trials, when $h(u) = \sqrt{u}$ and known $t_3 = t_4$

Input: An initial state x_0, neighborhood $\mathcal{N}$, constant $\psi \in (0,1)$

Step 1. Calculate $\log_p(\pi(y_j)/\pi(x_0))$ for all $y_j \in \mathcal{N}(x_0)$, $j = 1, \ldots, |\mathcal{N}(x_0)|$.

Step 2. Run k-means algorithm with k=2 on log-ratios $\{\log_p(\pi(y_j)/\pi(x_0))\}$ to obtain a partition, $\mathcal{C}_1$ (set of small log-ratios) and $\mathcal{C}_2$ (set of big log-ratios).

Step 3. Let $\hat{t}_2 = \min \mathcal{C}_2$ †, $\hat{t}_1 = \max \mathcal{C}_1$ and $\hat{s}_0 = |\mathcal{C}_2|$.

while $p^{(\hat{t}_2-\hat{t}_1)/2} < p^{t_3}/\hat{s}_0$ **do**

  Update $\mathcal{C}_1 \leftarrow \mathcal{C}_1 \backslash \{\hat{t}_1\}$, $\hat{t}_1 \leftarrow \max \mathcal{C}_1$ and $\hat{s}_0 \leftarrow \hat{s}_0 + 1$.

Output: The number of trials $N = \lfloor (p^{t_3}/\hat{s}_0)^{\psi} \rfloor$

† If $\hat{t}_2 < t_4$, redefine $\mathcal{C}_1$, $\mathcal{C}_2$ such that $\hat{t}_1 = \max \mathcal{C}_1 \leq t_4 < \min \mathcal{C}_2 = \hat{t}_2$. If no such $\mathcal{C}_2$ exists, set $\hat{t}_2 = t_4$, $\hat{s}_0 = 1$ and $\hat{t}_1 = \max \mathcal{C}_1$.

Algorithm 2 is depicted in fig. 2. In Step 1, log-probability ratio is calculated for all neighboring states of the initial state x_0. In Step 2, 2-means clustering algorithm identifies two subsets: the bad moves $\mathcal{C}_1$ and good moves $\mathcal{C}_2$. In Step 3, we set $\hat{t}_2 = \min \mathcal{C}_2$ which is motivated from Condition (i) of Theorem 1 where t_2 is a lower bound of log-probability ratios between the current state x and a good move x'. Although we considered any $x \in \mathcal{X} \backslash \{x^*\}$ in Condition (i), analyzing the initial state x_0 is not only computationally simple, but also gives a good guess of t_2 since as chain proceeds to the highest posterior state x^*, the number of good moves and the magnitude of posterior ratios generally decreases, for example in the Bayesian variable selection problem [40]. After t_2 is fixed, we first set $\hat{t}_1 = \max \mathcal{C}_1$, $\hat{s}_0 = |C_2|$ and gradually decrease t_1 (such that s_0 increases by one) until the "crossing" $p^{(\hat{t}_2 - \hat{t}_1)/2} > p^{t_3}/\hat{s}_0$ happens. The choice of $p^{t_3}/\hat{s}_0$ immediately after the crossing point approximates the worst-case scenario of $\min\{p^{(t_2-t_1)/2}, p^{t_3}/s_0\}$, hence providing a conservative estimate of N.

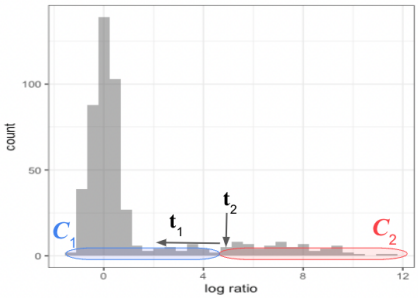

Figure 2: An illustration of Algorithm 2

4 Simulation studies and discussion

In this section, we present the results for two Bayesian model selection problems, BVS and SBM. We defer the result for spatial clustering models to Appendix B.1 due to the page limit. Throughout the simulation studies, we use symmetric proposal kernels. We run single-try MH and MTM with different choices of $N \in \{5, 10, 50, 100, 500, 1000, 2000, 5000\}$ and four different choices of weight functions, one with (unscaled) ordinary weight function $w_{\text{ord}}(y \,|\, x) = \pi(y)$, and three with the proposed balancing functions $w_{\text{sqrt}}(y \,|\, x) = \sqrt{\pi(y)/\pi(x)}$, $w_{\min}(y \,|\, x) = \min\{1, \pi(y)/\pi(x)\}$, and $w_{\max}(y \,|\, x) = \max\{1, \pi(y)/\pi(x)\}$. We consider the scenarios where the data generating model x^* receives the highest posterior probability to verify our main theoretical results. Similarly to [40], two performance measures are considered: 1) $H = \min\{t : x_t = x^*\}$: the number of iterations until the Markov chain $(x_0, x_1, \dots)$ hits x^*, 2) T_H: the wall time taken until the chain hits x^*.

4.1 Bayesian variable selection (BVS)

Under a high-dimensional linear model where the number of predictors p is much larger than the sample size n, BVS seeks to find the best subset of predictors using the sparsity-inducing prior. Consider the following sparsity-inducing prior of [35]:

$$\begin{aligned}
&\text{Linear model:} && \boldsymbol{y} = \boldsymbol{X}_\gamma \boldsymbol{\beta}_\gamma + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathsf{N}\left(\boldsymbol{0}, \phi^{-1} I_n\right) \\
&\text{Prior on } \boldsymbol{\beta}_\gamma \text{ and } \phi: && \boldsymbol{\beta}_\gamma \mid \phi, \gamma \sim \mathsf{N}\left(\boldsymbol{0}, \mathscr{G}\phi^{-1}\left(\boldsymbol{X}_\gamma^\top \boldsymbol{X}_\gamma\right)^{-1}\right), \quad \pi(\phi) \propto 1/\phi \\
&\text{Sparsity prior:} && \pi(\gamma) \propto p^{-\kappa|\gamma|} \mathbb{1}\left[|\gamma| \le s_{max}\right]
\end{aligned}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the response vector, $\boldsymbol{X}_\gamma$ is the submatrix of design matrix $\boldsymbol{X}$ formed of all columns with $\gamma_j = 1$, $\boldsymbol{\beta}_\gamma \in \mathbb{R}^{|\gamma|}$ is the subvector of $\boldsymbol{\beta} \in \mathbb{R}^p$ with nonzero coefficients, where $|\gamma| = \sum_{j=1}^p \gamma_j$. The hyperparameters $\mathscr{G}$, κ and $s_{\max}$ control the weight of prior effect [37], sparsity strength and the maximum model size, respectively. We consider the following random walk proposal kernel:

$$\mathbf{K}_{\text{RW}}(\gamma, \gamma') = \begin{cases} \frac{1}{p}\mathbb{1}_{\mathcal{N}_1(\gamma)}(\gamma') \text{ (single flip)} & \text{if } |\gamma| < s_{\max}, \\ \frac{1}{2p}\mathbb{1}_{\mathcal{N}_1(\gamma)}(\gamma') + \frac{1}{2|\gamma|(p-|\gamma|)}\mathbb{1}_{\mathcal{N}_2(\gamma)}(\gamma') \text{ (single or double flip)} & \text{if } |\gamma| = s_{\max}. \end{cases}$$

where $\mathcal{N}_1(\gamma) = \{\gamma' : d_{\text{H}}(\gamma, \gamma') = 1\}$ and $\mathcal{N}_2(\gamma) = \{\gamma' = (\gamma \cup \{j\}) \backslash \{\ell\} : j \notin \gamma, \ell \in \gamma\}$.

With $n = 1000$ and $p = 5000$, we consider different settings of design matrix where each row $\boldsymbol{x}_i$ follows $\boldsymbol{x}_i \overset{\text{iid}}{\sim} \mathsf{N}(0, I_p)$ (independent) or $\boldsymbol{x}_i \overset{\text{iid}}{\sim} \mathsf{N}(0, \Sigma)$, $\Sigma_{jj'} = \exp(-|j - j'|)$ (dependent). The response vector $\boldsymbol{y}$ is generated from $\boldsymbol{y} \sim \mathsf{N}(\boldsymbol{X}\boldsymbol{\beta}, I_n)$, with $\boldsymbol{\beta}$ having 10 nonzero entries at the

first 10 coordinates: $\boldsymbol{\beta}_{1:10} = \text{SNR}\sqrt{\log p/n}(2, -3, 2, 2, -3, 3, -2, 3, -2, 3)$ where SNR $\in \{2, 4\}$ is a signal-to-noise ratio. Under four different settings, we simulated 50 replicate datasets. Per each replicated data, algorithms are randomly initialized with state γ_0 such that $\gamma_0 \cap \gamma^* = \emptyset$ and $d_{\text{H}}(\gamma_0, \gamma^*) = 20$ which implies $H = 20$ is the minimum required hit iteration. All hyperparameter setups, the number of MCMC iterations, and other details are deferred to Appendix B.2.

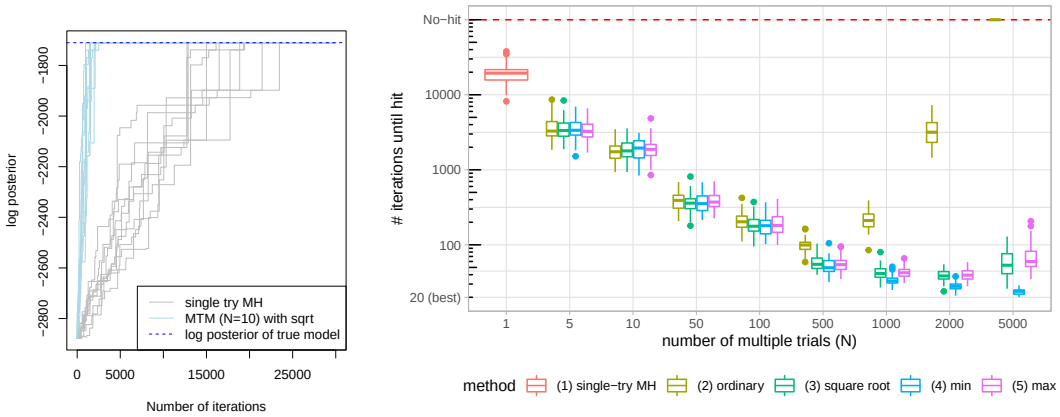

Figure 3: BVS simulation results for independent design and SNR = 4. (Left) Trace plots of unnormalized log-posterior probability using single-try MH and MTM ($N = 10$) with square root weighting function. (Right) Boxplot of H, the number of iterations until hit, against different choice of weight function and N based on 50 replicates.

Results under independent design with SNR=4 are summarized in fig. 3. The trace plot shows that single-try MH reaches the true state γ^* at around 20,000 iterations, whereas the MTM with $N = 10$ reaches the true state γ^* at around 2,000 iterations, smaller by a factor of 10. The boxplot confirms our findings that larger number of trials N is not always desirable, especially when using the ordinary weight function w_{ord}. The proposed weight functions using three different balancing functions perform much better than w_{ord} for large N. Especially, the performance of $w_{\min}$ seems to be robust to the choice of large N in the high SNR case. This can be explained by the fact that the denominator in (2) is often larger in high SNR data, and the choice of $\min\{1, \pi(y)/\pi(x)\}$ limits its growth; see also Section 3.1. The choice of N suggested from Algorithm 2 with $\psi = 0.9$ is overall reasonable, which has a median of 349 over 50 datasets. The detailed results for other settings are available at Appendix B.2.

4.2 Stochastic block model (SBM)

SBM [17] is a popular generative model of graph which assumes a block structure of edge connection probabilities to describe community structure. We consider Bayesian SBM [26] by assigning prior on edge probabilities and prior on partition $(z_i) = \mathbf{z} \in \{1, \dots, K\}^p$ with fixed number of clusters K, where z_i indicates membership label of i-th node. Here each model is represented as a partition $\mathbf{z}$, and the main interest is to find $\mathbf{z}$ that best describes the observed graph. We follow the prior of [41]:

$$\begin{aligned} \text{Edge appearance:} \quad & A_{ij} \mid Q, \mathbf{z} \overset{\text{ind}}{\sim} \mathsf{Bernoulli}(Q_{z_i z_j}), 1 \le i < j \le p \\ \text{Prior on blockwise probabilities:} \quad & Q_{uv} \overset{\text{iid}}{\sim} \mathsf{Beta}(\kappa_1, \kappa_2), 1 \le u \le v \le K \\ \text{Prior on partition:} \quad & \pi(\mathbf{z}) \propto \mathbb{1}(\mathbf{z} \in S_\alpha) \end{aligned}$$

The restricted space $S_\alpha = \{\mathbf{z} : \sum_{i=1}^p \mathbb{1}(z_i = u) \in \left[\frac{p}{\alpha K}, \frac{\alpha p}{K}\right] \text{ for all } u = 1, \dots, K\}$ for some $\alpha > 0$ excludes partitions whose block sizes differ too much. The graph is generated from the homogeneous SBM [14, 41], where $Q_{uv} = a$ if $u = v$ and $Q_{uv} = b$ otherwise. Both results assume the true partition $\mathbf{z}^*$ is balanced; i.e. the block sizes are all the same which ensures $\mathbf{z}^* \in S_\alpha$. The Chernoff-Hellinger divergence (CH) [1], defined as $\text{CH} := p(\sqrt{a} - \sqrt{b})^2/(K \log p)$ for $a, b \asymp \log p/p$, can be interpreted as a signal-to-noise ratio, and must be greater than 1 to ensure strong consistency [38, 1, 41].We consider the following proposal kernel where $\tilde{d}_{\text{H}}$ is a permutation-invariant Hamming distance [38]:

$$\mathbf{K}_{\text{RW}}(\mathbf{z}, \mathbf{z}') = 1/(p(K-1))\mathbb{1}_{\mathcal{N}(\mathbf{z})}(\mathbf{z}'), \quad \mathcal{N}(\mathbf{z}) = \{\mathbf{z}' : \tilde{d}_{\text{H}}(\mathbf{z}, \mathbf{z}') = 1\} \quad \text{(single flip)}$$

With the number of nodes $p = 1000$, we consider two different settings of the blocks (clusters) $K \in \{2, 5\}$ with true partition $\mathbf{z}^*$ being balanced. In each setting of K, we generated graph from the homogeneous SBM and where edge connection probabilities are set such that $\mathrm{CH} \approx 2$ (weak signal) and $\mathrm{CH} \approx 10$ (high signal). Under the four different settings of (K, CH) pair, we simulated 50 replicate datasets. For each replicated data, algorithms are randomly initialized with state $\mathbf{z}_0$ such that $\tilde{d}_{\mathrm{H}}(\mathbf{z}_0, \mathbf{z}^*) = 400$ which implies $H = 400$ is the minimum required hit iteration. All hyperparameter setups, the number of MCMC iterations, and other details are deferred to Appendix B.3.

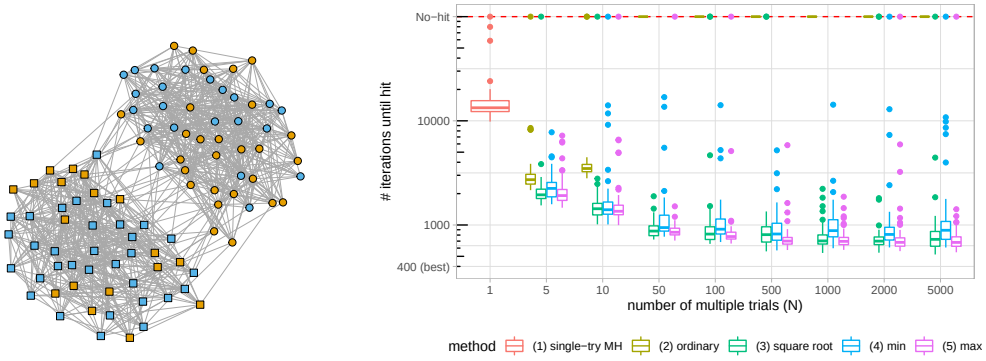

Figure 4: (Left) A graph generated from SBM with $p = 100, K = 2, a = 0.32$, and $b = 0.02$ so that $\mathrm{CH} \approx 2$. Node shape (square, circle) denotes the true partition. Node color (blue, yellow) represents the initial state $\mathbf{z}_0$ satisfying $\tilde{d}_{\mathrm{H}}(\mathbf{z}_0, \mathbf{z}^*) = 40$. (Right) Boxplot of H against different weight functions and N based on 50 replicates when $p = 1000, K = 2$, and $\mathrm{CH} \approx 2$.

Results for $K = 2$ and $\mathrm{CH} \approx 2$ are summarized in fig. 4. Left shows an example of SBM data and initialization when $p = 100$. The boxplot of H is similar to that of fig. 3: the ordinary weight function quickly deteriorates as N increases, but other weight functions of form (4) mostly converge with larger choice of N, although some Markov chains fail to converge due to the low CH . Especially, now the performance of $w_{\max}$ or w_{sqrt} is better than $w_{\min}$ when N is large in the low SNR case. This can be explained by the fact that in low SNR data, there are few neighboring states that has high probability ratio. It is important to catch such states to increase the acceptance ratio (2), and the min weight function cannot do this efficiently because of it is truncated at 1. The choice of N suggested from Algorithm 2 with $\psi = 0.9$ has median of 15 over 50 datasets. The detailed simulation results for other settings are available at Appendix B.3.

Finally, Table 1 provides the summary of T_H for the previous two examples, which shows clear computational benefit of MTM over the single-try MH due to the parallelism. As N increases, computation becomes more demanding compared to the decreasing rate of H, so the optimal choice of N with respect to T_H is often less than the minimizer of H.

Table 1: Summary of T_H, median wall time (in seconds) until the Markov chain hit the highest posterior state, for the previous two examples using the square root weight function over 50 replicates.

N		1	5	10	50	100	500	1000	2000	5000
T_H	BVS, indep, SNR$= 4$	1.30	0.81	0.46	0.11	0.07	**0.04**	0.05	0.09	0.30
	SBM, $K = 2$, $\mathrm{CH} \approx 2$	0.82	0.40	**0.35**	0.48	0.75	1.49	2.15	3.78	9.30

Discussion and future work In the paper, we prove that the mixing time bound of the MTM algorithm is smaller than that of the MH algorithm by a factor of the number of trials. Motivated by the observation that popular choices of weight function can cause mixing problems, we propose a new class of weight functions for which a mixing time bound is proven. We also suggest a theoretically guided choice of N for practical use. Future research may investigate the mixing time of other generalizations of MTM such as the case with correlated weight functions. It will also be of interest to extend MTM and analyze its mixing time to better handle the potential multimodality in the target distribution, by combining it with techniques such as annealing or tempering [8].

References

[1] E. Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

[2] D. Aldous. Random walks on finite groups and rapidly mixing markov chains. In *Séminaire de Probabilités XVII 1981/82*, pages 243–297. Springer, 1983.

[3] D. Aldous and P. Diaconis. Shuffling cards and stopping times. *The American Mathematical Monthly*, 93(5):333–348, 1986.

[4] D. Barry and J. A. Hartigan. A bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319, 1993.

[5] M. Bédard, R. Douc, and E. Moulines. Scaling analysis of multiple-try mcmc methods. *Stochastic Processes and their Applications*, 122(3):758–786, 2012.

[6] L. S. Blackford, A. Petitet, R. Pozo, K. Remington, R. C. Whaley, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, et al. An updated set of basic linear algebra subprograms (blas). *ACM Transactions on Mathematical Software*, 28(2):135–151, 2002.

[7] R. Bubley and M. Dyer. Path coupling: A technique for proving rapid mixing in markov chains. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 223–231. IEEE, 1997.

[8] R. Casarin, R. Craiu, and F. Leisen. Interacting multiple try algorithms with different proposal distributions. *Statistics and Computing*, 23(2):185–200, 2013.

[9] S. Chib. Estimation and comparison of multiple change-point models. *Journal of econometrics*, 86(2):221–241, 1998.

[10] R. V. Craiu and C. Lemieux. Acceleration of the multiple-try metropolis algorithm using antithetic and stratified sampling. *Statistics and computing*, 17(2):109–120, 2007.

[11] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of markov chains. *The annals of applied probability*, pages 36–61, 1991.

[12] R. L. Dobrushin, R. Koteckỳ, and S. Shlosman. *Wulff construction: a global shape from local interaction*, volume 104. American Mathematical Society Providence, 1992.

[13] N. Friedman and D. Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning*, 50(1):95–125, 2003.

[14] J. Geng, A. Bhattacharya, and D. Pati. Probabilistic community detection with unknown number of communities. *Journal of the American Statistical Association*, 114(526):893–905, 2019.

[15] V. Guruswami. Rapidly mixing markov chains: a comparison of techniques (a survey). *arXiv preprint arXiv:1603.01512*, 2016.

[16] C. Hans, A. Dobra, and M. West. Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association*, 102(478):507–516, 2007.

[17] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[18] M. Jerrum. A very simple algorithm for estimating the number of k-colorings of a low-degree graph. *Random Structures & Algorithms*, 7(2):157–165, 1995.

[19] D. A. Levin and Y. Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[20] F. Li and H. Sang. Spatial homogeneity pursuit of regression coefficients for large datasets. *Journal of the American Statistical Association*, 114(527):1050–1062, 2019.

[21] J. S. Liu, F. Liang, and W. H. Wong. The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.

[22] Z. T. Luo, H. Sang, and B. Mallick. A Bayesian contiguous partitioning method for learning clustered latent variables. *Journal of Machine Learning Research*, 22(37):1–52, 2021.

[23] D. Madigan, J. York, and D. Allard. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pages 215–232, 1995.

[24] L. Martino and F. Louzada. Issues in the multiple try metropolis mixing. *Computational Statistics*, 32(1):239–252, 2017.

[25] L. Martino and J. Read. On the flexibility of the design of multiple try metropolis schemes. *Computational Statistics*, 28(6):2797–2823, 2013.

[26] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association*, 96(455):1077–1087, 2001.

[27] S. Pandolfi, F. Bartolucci, and N. Friel. A generalization of the multiple-try metropolis algorithm for bayesian estimation and model selection. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 581–588. JMLR Workshop and Conference Proceedings, 2010.

[28] P. H. Peskun. Optimum monte-carlo sampling using markov chains. *Biometrika*, 60(3):607–612, 1973.

[29] D. Pollard and D. Yang. Rapid mixing of a markov chain for an exponentially weighted aggregation estimator. *arXiv preprint arXiv:1909.11773*, 2019.

[30] D. Randall. Rapidly mixing markov chains with applications in computer science and physics. *Computing in Science & Engineering*, 8(2):30–41, 2006.

[31] L. Saloff-Coste. Lectures on finite markov chains. *Lectures on probability theory and statistics*, pages 301–413, 1997.

[32] A. Sinclair. Improved bounds for mixing rates of markov chains and multicommodity flow. *Combinatorics, probability and Computing*, 1(4):351–370, 1992.

[33] M. G. Tadesse and M. Vannucci. *Handbook of Bayesian variable selection*. CRC Press, 2021.

[34] X. Yang and J. S. Liu. Convergence rate of multiple-try metropolis independent sampler. *arXiv preprint arXiv:2111.15084*, 2021.

[35] Y. Yang, M. J. Wainwright, and M. I. Jordan. On the computational complexity of high-dimensional bayesian variable selection. *The Annals of Statistics*, 44(6):2497–2532, 2016.

[36] G. Zanella. Informed proposals for local mcmc in discrete spaces. *Journal of the American Statistical Association*, 115(530):852–865, 2020.

[37] A. Zellner. On assessing prior distributions and bayesian regression analysis with g-prior distributions. *Bayesian inference and decision techniques*, 1986.

[38] A. Y. Zhang and H. H. Zhou. Minimax rates of community detection in stochastic block models. *The Annals of Statistics*, 44(5):2252–2280, 2016.

[39] Q. Zhou and H. Chang. Complexity analysis of bayesian learning of high-dimensional dag models and their equivalence classes. *arXiv preprint arXiv:2101.04084*, 2021.

[40] Q. Zhou, J. Yang, D. Vats, G. O. Roberts, and J. S. Rosenthal. Dimension-free mixing for high-dimensional bayesian variable selection. *arXiv preprint arXiv:2105.05719*, 2021.

[41] B. Zhuo and C. Gao. Mixing time of metropolis-hastings for bayesian community detection. *J. Mach. Learn. Res.*, 22:10–1, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See line 57-66 for the scope of our contributions.
   (b) Did you describe the limitations of your work? [Yes] See line 354-357 for the future work direction.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This work is mainly theoretical and does not present any foreseeable societal consequences.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] At the bottom of page 5, we clearly state three conditions and their interpretation of our main theorem.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A for all proofs.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementerial material for the code and Appendix B for the detailed settings of the simulation results.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We included the details in Section 4 and Appendix B
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 3, 4 in the main text. and Figure 5 in Appendix B.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix B for the detailed settings of the simulation results.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See attached code in the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]