Bayesian Analysis (2024) **19**, Number 2, pp. 595–621

Bayesian Analysis of Exponential Random Graph Models Using Stochastic Gradient Markov Chain Monte Carlo*†

Qian Zhang‡ and Faming Liang§

Abstract. The exponential random graph model (ERGM) is a popular model for social networks, which is known to have an intractable likelihood function. Sampling from the posterior for such a model is a long-standing problem in statistical research. We analyze the performance of the stochastic gradient Langevin dynamics (SGLD) algorithm (also known as noisy Longevin Monte Carlo) in tackling this problem, where the stochastic gradient is calculated via running a short Markov chain (the so-called inner Markov chain in this paper) at each iteration. We show that if the model size grows with the network size slowly enough, then SGLD converges to the true posterior in 2-Wasserstein distance as the network size and iteration number become large regardless of the length of the inner Markov chain performed at each iteration. Our study provides a scalable algorithm for analyzing large-scale social networks with possibly high-dimensional ERGMs.

MSC2020 subject classifications: Primary 62F15; secondary 62A09.

Keywords: inner Markov chain, intractable normalizing constant, log-concave density, social network, Wasserstein distance.

1 Introduction

The exponential random graph model (ERGM) (Robins et al., 2007a,b), as a popular model for social networks, has the capacity to describe a wide range of dependence structures among the actors within a social network and support statistical inference of social networks from various backgrounds. Consider an N-actor social network with adjacency matrix $\boldsymbol{y} \in \mathcal{Y}$, where

$$\mathcal{Y} := \{\boldsymbol{y} : \boldsymbol{y} \in \{0,1\}^{N\times N}, \boldsymbol{y} = \boldsymbol{y}^T,\ \boldsymbol{y}_{ii} = 0,\ 1 \le i \le N\}.$$

The ERGM defines the probability mass function (or likelihood) of $\boldsymbol{y}$ via an exponential family distribution as

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\theta}^T \boldsymbol{S}(\boldsymbol{y})}}{Z(\boldsymbol{\theta})}, \tag{1.1}$$

*Liang's research is supported in part by NSF grants DMS-2015498 and DMS-2210819 and the NIH grant R01-GM126089.

†Supplementary material of this article is available at the journal's website.

‡Department of Statistics, Purdue University, West Lafayette, IN 47907, USA, zhan3761@purdue.edu

§Department of Statistics, Purdue University, West Lafayette, IN 47907, USA. Corresponding author, fmliang@purdue.edu

 https://doi.org/10.1214/23-BA1364

where $\boldsymbol{\theta} \in \mathbb{R}^p$ is the vector of parameters, $\boldsymbol{S}(\boldsymbol{y}) \in \mathbb{R}^p$ is the vector of sufficient statistics, and $Z(\boldsymbol{\theta}) = \sum_{\tilde{\boldsymbol{y}} \in \mathcal{Y}} e^{\boldsymbol{\theta}^T \boldsymbol{S}(\tilde{\boldsymbol{y}})}$ is the normalizing constant which is intractable even for a moderate value of N. Let $\pi(\boldsymbol{\theta})$ denote the prior distribution of $\boldsymbol{\theta}$. Then the posterior distribution of $\boldsymbol{\theta}$ is given by $\pi(\boldsymbol{\theta}|\boldsymbol{y}) = c(\boldsymbol{y})\pi(\boldsymbol{\theta})e^{\boldsymbol{\theta}^T \boldsymbol{S}(\boldsymbol{y})}/Z(\boldsymbol{\theta})$, where $c(\boldsymbol{y})$ is the inverse of the normalizing constant of the posterior distribution, while $Z(\boldsymbol{\theta})$ is still intractable.

The existence of the intractable normalizing constant $Z(\boldsymbol{\theta})$ has brought a great challenge to statistical inference of the ERGM. For example, the maximum likelihood estimator (MLE) of $\boldsymbol{\theta}$ cannot be directly computed, and the posterior $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ cannot be directly sampled from, either. To tackle this difficulty, a variety of methods have been proposed in the literature. The existing methods can be roughly classified into two categories according to the strategies employed by them, namely, approximation-based methods and auxiliary sample-based methods. See also Park and Haran (2018) for a comparative review of the existing methods from Bayesian perspective.

The methods in the first category are to approximate the intractable normalizing constant $Z(\boldsymbol{\theta})$, the gradient $\nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta})$, or the ratio $Z(\boldsymbol{\theta})/Z(\boldsymbol{\theta}')$. For example, Geyer and Thompson (1992) proposed the Monte Carlo maximum likelihood estimation (MCMLE) method for estimating $\boldsymbol{\theta}$, where $Z(\boldsymbol{\theta})$ is approximated using importance sampling based on the samples drawn from a trial distribution $p(\boldsymbol{y}|\boldsymbol{\theta}_0)$ for a pre-specified point $\boldsymbol{\theta}_0$. It is known that this method can converge to a suboptimal solution, if $\boldsymbol{\theta}_0$ is not close to the true MLE.

Liang (2007) and Atchade et al. (2013) proposed to approximate $Z(\boldsymbol{\theta})$ using an adaptive kernel smoothing method, where $Z(\boldsymbol{\theta})$ is viewed as a marginal density function of the unnormalized distribution $e^{\boldsymbol{\theta}^T \boldsymbol{S}(\boldsymbol{y})}$. Toward sampling from the posterior $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, Liang and Jin (2013) proposed the Monte Carlo Metropolis-Hastings (MCMH) algorithm, where $Z(\boldsymbol{\theta})/Z(\boldsymbol{\theta}')$ is approximated at each iteration using auxiliary samples simulated from either $p(\boldsymbol{y}|\boldsymbol{\theta})$ or $p(\boldsymbol{y}|\boldsymbol{\theta}')$ through an inner Markov chain. Alquier et al. (2016) called the MCMH algorithm a noisy exchange algorithm and established its approximation rate with respect to the length of the inner Markov chain. The Bayesian stochastic approximation Monte Carlo (SAMC) algorithm (Jin and Liang, 2014) and the marginal MCMC algorithm (Everitt, 2012) also belong to the class of noisy exchange algorithms, which, in general, require to simulate a large number of auxiliary samples at each iteration for ensuring their convergence. Quite recently, Alquier et al. (2016) proposed the noisy Langevin algorithm, where auxiliary samples generated by an inner Markov chain are used to approximate the gradient $\nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta})$ at each iteration. While the noisy Hamiltonian Monte Carlo (HMC) algorithm (Stoehr et al., 2019) approximates both $\nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta})$ and $Z(\boldsymbol{\theta})/Z(\boldsymbol{\theta}')$ using the auxiliary samples at each iteration.

The methods in the second category aim to cancel the normalizing constant ratio $Z(\boldsymbol{\theta})/Z(\boldsymbol{\theta}')$ by augmenting appropriate auxiliary samples to the target distribution and/or the proposal distribution. The early works include Møller et al. (2006) and Murray et al. (2006), which are limited to spin systems only as they require a perfect sampler (Propp and Wilson, 1996) for drawing auxiliary samples. To tackle this limitation, Liang (2010) proposed the double Metropolis-Hastings (DMH) sampler, where auxiliary samples are drawn through an inner Markov chain induced by the Metropolis-Hastings algorithm. Similar algorithms have been proposed in Caimo and Friel (2011)

and Everitt (2012) for social network analysis. Since in these algorithms a short inner Markov chain is used for generating auxiliary samples at each iteration, the resulting estimates are only approximately correct. To overcome this issue, Liang et al. (2016) proposed an adaptive exchange (AEX) algorithm, where auxiliary samples are drawn via an importance sampling procedure running in parallel to the target Markov chain.

The literature review shows that use of a short inner Markov chain for generating auxiliary samples has been adopted by many Bayesian methods in dealing with the issue of intractable normalizing constants, see e.g., Liang and Jin (2013), Alquier et al. (2016), and Stoehr et al. (2019). Unfortunately, by the theory of Alquier et al. (2016), those methods often fail to converge to the true posterior $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ unless the inner Markov chain has been run long enough at each iteration. In particular, for the noisy Langevin Monte Carlo algorithm, their theory implies that to ensure the convergence to the true posterior, the length of the inner Markov chain at iteration t should grow with the dimension of the ERGM and the reciprocal of the learning rate of the Langevin Monte Carlo algorithm.[1] Therefore, if a decreasing learning rate is used for the Langevin Monte Carlo algorithm, the length of the inner Markov chain should grow with iterations, making the algorithm impractical.

In this paper, we re-analyzed the convergence of the noisy Langevin Monte Carlo algorithm based on the convergence theory of the stochastic gradient Langevin Monte Carlo (SGLD) algorithm (Welling and Teh, 2011), see e.g., Dalalyan and Karagulyan (2019), Bhatia et al. (2019), and Song et al. (2020). We show that for an ERGM, if $p = o(N^{\kappa})$ holds for some constant κ, then the noisy Langevin Monte Carlo algorithm converges to the true posterior distribution in 2-Wasserstein distance as $N \to \infty$ and $t \to \infty$ regardless of the length of the inner Markov chain performed at each iteration. This result implies that the noisy Langevin Monte Carlo algorithm is scalable with respect to the network size. Further, since the posterior $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ is strongly log-concave (with an appropriate prior) as shown in Section 3, it follows from Dalalyan and Karagulyan (2019) and Durmus et al. (2019) that noisy Langevin Monte Carlo can converge to the true posterior very fast, whose computational complexity increases only linearly with respect to the dimension p.

We note that the existing exact methods can be very inefficient for large-scale networks and high-dimensional ERGMS. For example, to generate auxiliary samples used for canceling the normalizing constant ratio $Z(\boldsymbol{\theta})/Z(\boldsymbol{\theta}')$, the AEX algorithm (Liang et al., 2016) learns a mixture trial distribution for which the number of components is required to increase exponentially with the dimension of $\boldsymbol{\theta}$. Therefore, AEX can be very inefficient for high-dimensional problems. Similarly, the methods by Liang (2007) and Atchade et al. (2013) also suffer from the curse of dimensionality.

The remaining part of this paper is organized as follows. Section 2 describes the proposed method. Section 3 proves the convergence of noisy Langevin Monte Carlo to the true posterior distribution under the large network regime. Sections 4 and 5 illustrate

[1] By Theorem 2.2 and Lemma 3 of Alquier et al. (2016), it is easy to get a lower bound for the length of the inner Markov chain at iteration t: $m_t \geq cp/[N^{\kappa}(\epsilon^{(t)})^2]$ for some constants c and κ in notations of this paper.

the performance of noisy Langevin Monte Carlo. Section 6 concludes the paper with a brief discussion.

2 Noisy Langevin Monte Carlo for Bayesian Analysis of ERGMs

2.1 The Noisy Langevin Monte Carlo Algorithm

Consider the ERGM (1.1). A straightforward calculation shows that

$$\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}|\boldsymbol{y}) = \boldsymbol{S}(\boldsymbol{y}) - \mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{S}(\boldsymbol{Y})] + \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}), \tag{2.1}$$

where $\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{S}(\boldsymbol{Y})] = \nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}) = \sum_{\tilde{\boldsymbol{y}} \in \mathcal{Y}} \boldsymbol{S}(\tilde{\boldsymbol{y}}) e^{\boldsymbol{\theta}^T \boldsymbol{S}(\tilde{\boldsymbol{y}})} / Z(\boldsymbol{\theta})$. Since an exhaustive evaluation for all possible configurations of $\tilde{\boldsymbol{y}} \in \mathcal{Y}$ is impossible even for a moderate value of N, we estimate $\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{S}(\boldsymbol{Y})]$ by averaging over the samples simulated by an inner Markov chain. As in DMH (Liang, 2010), the Markov chain is initialized at the observed network $\boldsymbol{y}$ and leaves $p(\boldsymbol{y}|\boldsymbol{\theta})$ as the invariant distribution. Let $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \ldots, \tilde{\boldsymbol{y}}_m$ denote the auxiliary samples collected from a single inner Markov chain. Then $\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{S}(\boldsymbol{Y})]$ can be estimated by

$$\widehat{\mathbb{E}}_{\boldsymbol{\theta}}[\boldsymbol{S}(\boldsymbol{Y})] = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{S}(\tilde{\boldsymbol{y}}_i), \tag{2.2}$$

which is known to be biased due to the finiteness of m. In this paper we always use m to denote the number of auxiliary samples used in estimating $\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{S}(\boldsymbol{Y})]$ at each iteration. Plugging this estimator into (2.1), we get

$$\widehat{\nabla}_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}|\boldsymbol{y}) = \boldsymbol{S}(\boldsymbol{y}) - \widehat{\mathbb{E}}_{\boldsymbol{\theta}}[\boldsymbol{S}(\boldsymbol{Y})] + \nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}).$$

The posterior $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ can then be simulated using the stochastic gradient Langevin dynamics (SGLD) algorithm (Welling and Teh, 2011) by iterating the following equation:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \frac{\epsilon^{(t)}}{2} \boldsymbol{D} \widehat{\nabla}_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}^{(t)}|\boldsymbol{y}) + \mathcal{N}(0, \epsilon^{(t)} \boldsymbol{D}), \tag{2.3}$$

where $\epsilon^{(t)}$ is a positive scalar and $\boldsymbol{D} \in \mathbb{R}^{d \times d}$ is a diagonal matrix, which together give component-specific learning rates for $\boldsymbol{\theta}$. For some experiments in this paper, we set

$$\boldsymbol{D} = \operatorname{diag}\Big\{ \frac{1}{S_1(\boldsymbol{y})}, \frac{1}{S_2(\boldsymbol{y})}, \ldots, \frac{1}{S_p(\boldsymbol{y})} \Big\},$$

where $S_k(\boldsymbol{y})$ denotes the k-th component of $\boldsymbol{S}(\boldsymbol{y})$. How to run the inner Markov chain at each iteration is described in Section 2.2, and the convergence of the algorithm is studied in Section 3. In what follows, we call (2.3) a noisy Langevin Monte Carlo algorithm, which is exchangeable with SGLD in this paper.

2.2 On Inner Markov Chain Simulations

For small networks, we simulate each auxiliary network by running the Gibbs sampler (Geman and Geman, 1984) in a full sweep, where each dyad of the network undergoes an update according to its conditional distribution. For large networks, we simulate auxiliary networks using the tie-no-tie (TNT) sampler (Morris et al., 2008) which updates only a subset of dyads of the network according to the Metropolis-Hastings rule. Typically, half of the dyads in the subset are randomly selected from the set of ties/edges of the network and the other half are randomly selected from the set of non-ties/non-edges of the network. Since the network is usually very sparse, the TNT sampler avoids the time wasted in updating the dyads with low probability to be edges and is thus efficient. Specifically, for a dyad with a tie, we propose to delete the tie with the acceptance probability given by

$$r_d = \min\left\{1, \frac{p(\boldsymbol{y}'|\boldsymbol{\theta})}{p(\boldsymbol{y}|\boldsymbol{\theta})} \frac{|\text{Edge}(\boldsymbol{y})|}{\binom{N}{2} - |\text{Edge}(\boldsymbol{y})| + 1} \frac{P(\text{adding})}{P(\text{deleting})}\right\},$$

where $\boldsymbol{y}'$ is the proposed network, $|\text{Edge}(\boldsymbol{y})|$ is the number of ties in $\boldsymbol{y}$, and $P(\text{adding}) = P(\text{deleting}) = 1/2$ according to our strategy. For a dyad without a tie, we propose to add the tie with the acceptance probability given by

$$r_a = \min\left\{1, \frac{p(\boldsymbol{y}'|\boldsymbol{\theta})}{p(\boldsymbol{y}|\boldsymbol{\theta})} \frac{\binom{N}{2} - |\text{Edge}(\boldsymbol{y})|}{|\text{Edge}(\boldsymbol{y})| + 1} \frac{P(\text{deleting})}{P(\text{adding})}\right\}.$$

For both the Gibbs sampler and the TNT sampler, the likelihood ratio

$$p(\boldsymbol{y}'|\boldsymbol{\theta})/p(\boldsymbol{y}|\boldsymbol{\theta}) = \exp\left\{\boldsymbol{\theta}^T(\boldsymbol{S}(\boldsymbol{y}') - \boldsymbol{S}(\boldsymbol{y}))\right\},$$

needs to be evaluated in each dyad update, where $\boldsymbol{y}'$ differs from $\boldsymbol{y}$ for only one edge. Let $\boldsymbol{\Delta}_{i,j}\boldsymbol{S}(\boldsymbol{y}) := \boldsymbol{S}(\boldsymbol{y}') - \boldsymbol{S}(\boldsymbol{y})$, where $\boldsymbol{y}'$ differs from $\boldsymbol{y}$ only at the (i,j)-th element. For some commonly used sufficient statistics of the ERGM (1.1), we provide recursive formulas for calculating $\boldsymbol{\Delta}_{i,j}\boldsymbol{S}(\boldsymbol{y})$ in the Appendix B of the Supplementary Material (Zhang and Liang, 2023).

3 Convergence Analysis

In this section, we rewrite a social network by $\boldsymbol{y}_N$, where the subscript indicates its size. Let $\pi_N^* = \pi(\boldsymbol{\theta}|\boldsymbol{y}_N)$ denote the posterior density function of $\boldsymbol{\theta}$, let $\pi_N^{(t)} = \pi(\boldsymbol{\theta}^{(t)}|\boldsymbol{y}_N)$ denote the density of $\boldsymbol{\theta}^{(t)}$ generated by the SGLD algorithm at iteration t, and let Θ denote the space of $\boldsymbol{\theta}$. We are interested in studying the discrepancy between π_N^* and $\pi_N^{(t)}$ in 2-Wasserstein distance. Refer to Gibbs and Su (2002) for discussions on the relation between the Wasserstein distance and other probability metrics. To study the convergence of the proposed method, the following conditions are assumed:

(A.1) The posterior π_N^* is strongly log-concave and gradient-Lipschitz:

$$U(\boldsymbol{\theta}) - U(\boldsymbol{\theta}') - \nabla U(\boldsymbol{\theta}')^T(\boldsymbol{\theta} - \boldsymbol{\theta}') \geq \frac{q_N}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2, \quad \forall\, \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta, \tag{3.1}$$

$$\|\nabla U(\boldsymbol{\theta}) - \nabla U(\boldsymbol{\theta}')\|_2 \leq Q_N\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2, \quad \forall\, \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta, \tag{3.2}$$

where $U(\boldsymbol{\theta}) = -\log \pi(\boldsymbol{\theta}|\boldsymbol{y}_N)$, and $c_0 N^\kappa \leq q_N \leq Q_N \leq c_1 N^\kappa$ holds for some positive constants c_0, c_1 and κ.

(A.2) The posterior π_N^* has bounded second moment, i.e., $\int_\Theta \boldsymbol{\theta}^T\boldsymbol{\theta}\pi(\boldsymbol{\theta}|\boldsymbol{y}_N)d\boldsymbol{\theta} = O(p)$, where p denotes the dimension of $\boldsymbol{\theta}$.

(A.3) $\mathbb{E}_{\boldsymbol{\theta}}\|\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta})\|^2 = O(N^\kappa(\|\theta\|^2 + p))$, where $\mathbb{E}_{\boldsymbol{\theta}}$ denotes expectation with respect to the distribution $p(\boldsymbol{y}_N|\boldsymbol{\theta})$, and κ denotes a positive constant as defined in (A.1).

Regarding these assumptions, we have the following justifications. For the ERGM (1.1), it is easy to derive that (see e.g., Fellows and Handcock (2017)),

$$\frac{\partial \log p(\boldsymbol{y}_N|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \boldsymbol{S}(\boldsymbol{y}_N) - \mathbb{E}_{\boldsymbol{\theta}}(\boldsymbol{S}(\boldsymbol{y}_N)), \quad -\frac{\partial^2 \log p(\boldsymbol{y}_N|\boldsymbol{\theta})}{\partial \theta_i \theta_j} = \mathrm{Cov}_{\boldsymbol{\theta}}(S_i(\boldsymbol{y}_N), S_j(\boldsymbol{y}_N)), \tag{3.3}$$

where the expectation $\mathbb{E}_{\boldsymbol{\theta}}(\cdot)$ and covariance $\mathrm{Cov}_{\boldsymbol{\theta}}(\cdot)$ are taken with respect to the distribution $p(\boldsymbol{y}_N|\boldsymbol{\theta})$. By (3.3), condition (A.1) is satisfied as long as the prior $\pi(\boldsymbol{\theta})$ is also strongly log-concave and gradient-Lipschitz. For example, we can set $\pi(\boldsymbol{\theta}) \propto 1$ or $\pi(\boldsymbol{\theta}) \propto \exp\{-\|\boldsymbol{\theta}\|^2/(2\sigma^2)\}$ for some $\sigma^2 > 0$. If the former is used, then we can set $q_N = \lambda_{\min}(\Sigma)$ and $Q_N = \lambda_{\max}(\Sigma)$, where Σ denotes the covariance matrix of $\boldsymbol{S}(\boldsymbol{y}_N)$, and $\lambda_{\min}(\Sigma)$ and $\lambda_{\max}(\Sigma)$ denote the smallest and largest eigenvalues of Σ, respectively. When the network size is large, since $S_i(\boldsymbol{y}_N)$'s generally increase with N, the eigenvalues of Σ also increases with N and becomes large. Further, since the space $\mathcal{Y}$ is finite and the same for any $\boldsymbol{\theta} \in \Theta$, the inequality $c_0 N^\kappa \leq q_N \leq Q_N \leq c_1 N^\kappa$ can hold uniformly over the parameter space Θ. However, we do note that the value of κ is model dependent, i.e., depending on the statistics $S_i(\boldsymbol{y}_N)$'s used in the model. If a Gaussian prior is used for $\boldsymbol{\theta}$, we can slightly adjust the values of q_N and Q_N by adding $1/\sigma^2$. A non-strongly log-concave prior density can also be applied to $\boldsymbol{\theta}$, while ensuring condition (A.1) holds.

Further, by (3.3) and with an appropriate prior of $\boldsymbol{\theta}$, we have

$$\mathbb{E}_{\boldsymbol{\theta}}\|\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta})\|^2 \lesssim \mathrm{trace}(\Sigma) \leq pQ_N,$$

which implies condition (A.3) is satisfied.

Theorem 3.1 concerns convergence of the noisy Langevin Monte Carlo algorithm for the ERGM, where p is allowed to increase with N, the learning rate ϵ is allowed to decrease with N, and m denotes the number of auxiliary samples used in estimation of $\nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta})$ as in (2.2). The proof of Theorem 3.1 is given in the Appendix A of the Supplementary Material (Zhang and Liang, 2023), where the bias of the estimator (2.2) has been taken into account.

Theorem 3.1. *Assume that the conditions (A.1)–(A.3) hold.*

(i) Let $\pi_N^{(t)}$ *denote the distribution of* $\boldsymbol{\theta}^{(t)}$*, and let* $\pi_N^* = \pi(\boldsymbol{\theta}|\boldsymbol{y}_N)$*. Then*

$$W_2(\pi_N^{(t)}, \pi_N^*) = (1-\omega)^t W_2(\pi_N^{(0)}, \pi_N^*) + O\big(\frac{\sqrt{p}}{mN^{\kappa/2}}\big) + O(\sqrt{\epsilon p}) + O\big(\frac{\sqrt{\epsilon p}}{m}\big), \quad (3.4)$$

for some $\omega \in (0,1)$*, where* $W_2(\cdot,\cdot)$ *denotes the second order Wasserstein distance between two distributions.*

(ii) If $\rho(\boldsymbol{\theta})$ *is integrable and* α*-Lipschitz for some constant* $\alpha > 0$*, then*

$$\sum_{t=1}^{T} \rho(\boldsymbol{\theta}^{(t)})/T - \pi_N^*(\rho) = \frac{\alpha(1-\omega)}{T\omega} W_2(\pi_N^{(0)}, \pi_N^*)[1-(1-\omega)^T] + O_p(T^{-1/2}) + O\big(\frac{\sqrt{p}}{mN^{\kappa/2}}\big) + O(\sqrt{\epsilon p}) + O\big(\frac{\sqrt{\epsilon p}}{m}\big), \quad (3.5)$$

where $\pi_N^*(\rho) = \int_\Theta \rho(\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{y}_N)d\boldsymbol{\theta}$*.*

In what follows, we give some remarks on the theorem. In the remarks, we allow the network size N to increase, while treating the dimension p, the learning rate ϵ, and the inner Markov chain length m as a function of N. However, for notational simplicity, we depressed their dependency on N in the notation. In the simplified notations, $a \prec$ (or $\succ$) b mean that both a and b depend on N and $\frac{a}{b}$ $\big($or $\frac{b}{a}\big) \to 0$ as $N \to \infty$.

Remark 3.1. *Under the high-dimensional setting, i.e., the learning rate* ϵ *decreases with* N *and* p *such that* $\epsilon \prec \min\{\frac{1}{N^\kappa}, \frac{1}{p}\}$*, and* m *increases with* p *such that* $m \succ \frac{\sqrt{p}}{N^{\kappa/2}}$ *(or* $p \prec N^\kappa$ *and* $m = O(1)$*), we have* $W_2(\pi_N^{(t)}, \pi_N^*) \to 0$ *and* $\sum_{t=1}^T \rho(\boldsymbol{\theta}^{(t)})/T - \pi_N^*(\rho) \xrightarrow{p} 0$ *as* $T \to \infty$ *and* $N \to \infty$*, where* $\xrightarrow{p} 0$ *denotes convergence in probability.*

Remark 3.2. *By Appendix A.2 of the Supplementary Material (Zhang and Liang, 2023), an approximate expression of* ω *is given by*

$$\omega \asymp \left(N^\kappa - \frac{N^{\kappa/2}}{m} - \sqrt{2}\frac{N^{\kappa/2}}{\sqrt{m}}\right)\epsilon, \quad (3.6)$$

which implies that a large value of m *can lead to a faster convergence rate of the algorithm. On the other hand, as implied by (3.4),* m *is not necessarily very large to counter the bias introduced by the estimator (2.2) especially when* N *is large. This seemly counterintuitive setting of* m *can be justified by equation (3.3), which implies that the posterior distribution is more and more concentrated as* N *increases (see also condition A.1). In theory, we need to set* $m \succ \sqrt{p}/N^{\kappa/2}$*. If* p *is held constant or has a low growth rate with* N *such that* $p \prec N^\kappa$ *holds, then we can set* $m = O(1)$*. That is, in this case, we have* $W_2(\pi_N^{(t)}, \pi_N^*) \to 0$ *holds as* $N \to \infty$ *and* $t \to \infty$ *for any choice of* m*.*

We note that the convergence of the noisy Langevin Monte Carlo algorithm has also been established in Alquier et al. (2016) (Theorem 3.2, page 36), but without giving an explicit convergence rate. By the theory of Alquier et al. (2016), the length of the inner Markov chain should satisfy the condition $m \geq cp/(N^\kappa\epsilon^2)$ *for some constant* c*, which represents a much larger number than* $\sqrt{p}/N^{\kappa/2}$ *established above.*

Remark 3.3. *As shown in (3.5), the estimation error consists of two components, variation and bias. If the computational budget is fixed, then we expect that m is inversely proportional to the total number of iterations T, i.e., $T = B/m$ for some constant B representing the fixed computational budget. In this case, we can set $m = O\left(\left(B^2\epsilon^2\alpha^4 N^\kappa W_2^4(\pi_N^{(0)}, \pi_N^*)\right)^{\frac{1}{3}}\right)$, which approximately minimizes the mean squared error of the estimate $\sum_{t=1}^T \rho(\boldsymbol{\theta}^{(t)})/T$ under the assumptions $\epsilon \prec N^{-\kappa}$ and $p \prec N^\kappa$.*

Theorem 3.1 is established with a constant learning rate. In practice, one may use a decaying learning rate, see e.g. Teh et al. (2016), where $\epsilon_t = O(1/t^\kappa)$ is suggested for some $0 < \kappa \leq 1$. For the decaying learning rate, Teh et al. (2016) established the consistency of some weighted averaging estimators for $\pi_N^*(\rho)$. Similar to Theorem 2 of Song et al. (2020), it is easy to show that the unweighted averaging estimator $\sum_{t=1}^T \rho(\boldsymbol{\theta}^{(t)})/T$ still forms a consistent estimator of $\pi_N^*(\rho)$ if the learning rate slowly decays at a rate of $\epsilon_t = O(1/t^\gamma)$ for $0 < \gamma < 1$. However, if $\gamma = 1$, a weighted averaging estimator is needed.

4 Experimental Results

This section presents five examples. A Gaussian prior or uniform prior $\pi(\boldsymbol{\theta}) \propto 1$ on Θ is employed for each of them. The detailed parameter settings are given in Appendix C of the Supplementary Material (Zhang and Liang, 2023).

4.1 A Simulation Study of Spatial Autologistic Models

To validate the proposed method, we first applied it to estimate the parameters of a spatial autologistic model with simulated data. In the spatial autologistic model, the graph is denoted by $\boldsymbol{y} \in \{-1, 1\}^{d_1 \times d_2}$, where each element $\boldsymbol{y}_{ij}$ is called a spin. Let $\mathcal{D}$ denote the set of indices of the spins, and let $n(i, j)$ denote the set of indices of the neighboring spins of $\boldsymbol{y}_{ij}$. The likelihood function of the graph $\boldsymbol{y}$ is given by

$$p(\boldsymbol{y}|\alpha, \beta) = \frac{1}{Z(\alpha, \beta)} \exp\left\{\alpha \sum_{(i,j)\in\mathcal{D}} \boldsymbol{y}_{ij} + \beta \sum_{(i,j)\in\mathcal{D}} \boldsymbol{y}_{ij}\left(\sum_{(k,l)\in n(i,j)} \boldsymbol{y}_{kl}\right)\right\},$$

where the parameter α determines the overall proportion of $\boldsymbol{y}_{ij} = \pm 1$, and the parameter β determines the intensity of interaction between $\boldsymbol{y}_{ij}$ and its neighbors. Let $\boldsymbol{\theta} = (\alpha, \beta)$ denote the parameter vector. In this example, we set $\mathcal{D}$ to a United States cancer mortality map compiled by Riggan et al. (1987) based on the case counts of liver and gallbladder (including bile ducts) cancers in white males during the decade 1950–1959. The mortality map consists of 2293 spins, each representing a county of the United States. Different values of (α, β) have been considered in the simulation, which are given in Table 1. For each value of (α, β), 50 independent graphs were simulated using a perfect sampler developed by Childs et al. (2001) for spin models. The free boundary

method	α	β	$\hat{\alpha}-\alpha$	$\hat{\beta}-\beta$	average CPU time (s)
SGLD	0	0.1	0.0003 (0.0026)	0.0018 (0.0022)	27.251
	0	0.2	0.0022 (0.0020)	0.0008 (0.0020)	35.230
	0	0.3	0.0023 (0.0015)	−0.0027 (0.0016)	32.217
	0	0.4	−0.0009 (0.0007)	−0.0025 (0.0011)	29.214
	0.1	0.1	0.0000 (0.0025)	−0.0009 (0.0023)	33.915
	0.3	0.3	0.0025 (0.0083)	0.0025 (0.0034)	31.841
	0.5	0.5	0.0017 (0.0206)	0.0072 (0.0091)	35.292
DMH	0	0.1	0.0007 (0.0027)	0.0014 (0.0023)	24.103
	0	0.2	0.0026 (0.0021)	0.0010 (0.0020)	22.081
	0	0.3	0.0018 (0.0017)	−0.0024 (0.0016)	21.348
	0	0.4	−0.0006 (0.0007)	−0.0034 (0.0013)	29.313
	0.1	0.1	−0.0005 (0.0025)	−0.010 (0.0023)	23.107
	0.3	0.3	0.0019 (0.0080)	0.0029 (0.0034)	35.136
	0.5	0.5	0.0713 (0.0252)	−0.0138 (0.0106)	31.461
AUEX	0	0.1	0.0001 (0.0026)	0.0017 (0.0022)	431.990
	0	0.2	0.0023 (0.0019)	0.0008 (0.0020)	1080.252
	0	0.3	0.0026 (0.0016)	−0.0022 (0.0016)	2886.858
	0.1	0.1	0.0002 (0.0024)	−0.0007 (0.0022)	472.623
	0.3	0.3	0.0015 (0.0081)	0.0029 (0.0035)	1251.778

Table 1: Parameter estimates and their standard deviations (reported in parentheses) for the simulated spatial autologistic model, where "SGLD" represents the proposed method.

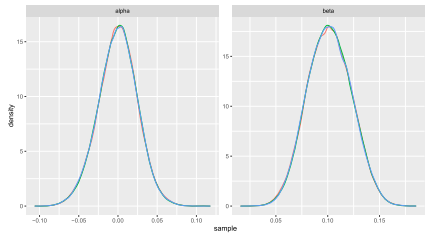
condition was used in simulations. Similar simulation studies have been considered in the literature, see e.g. Liang (2010) and Liang et al. (2016).

We have applied the proposed method, DMH (Liang, 2010) and exchange algorithm (Murray et al., 2006) (denoted by AUEX) to this example with the improper prior $\pi(\boldsymbol{\theta}) \propto 1$. The detailed parameter settings are given in Appendix C of the Supplementary Material (Zhang and Liang, 2023). Note that AUEX is exact, where the perfect sampler (Childs et al., 2001) was used for generating the auxiliary sample at each iteration. Since the perfect sampler can be extremely slow when β is close or greater than the critical value of the model, the estimates for the cases $\beta = 0.4$ and 0.5 cannot be obtained.

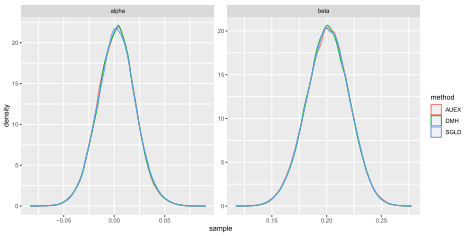
The results are summarized in Table 1, where the CPU time was measured on a computer of Intel(R) Xeon(R) Gold 6126 CPU@2.60GHz. All computations reported in this paper were done on the same computer. The comparison shows that the proposed method (denoted by SGLD) produced about the same accurate estimates as AUEX, while costing much less CPU time than AUEX; SGLD cost about the same CPU time as DMH, while producing more accurate estimates for strongly interaction models, e.g., the model with $(\alpha, \beta) = (0.5, 0.5)$.

Figure 1 compares the density plots of posterior samples generated by different methods, where AUEX is only available for the weak interaction cases. The comparison

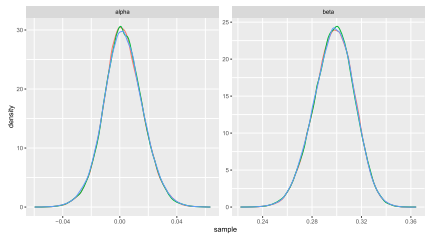
(a) $\alpha = 0, \beta = 0.1$



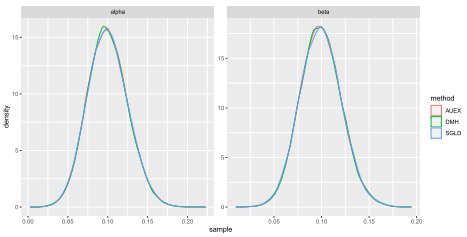
(b) $\alpha = 0, \beta = 0.2$



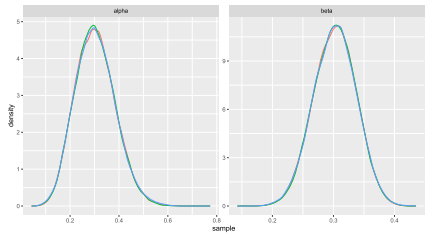
(c) $\alpha = 0, \beta = 0.3$



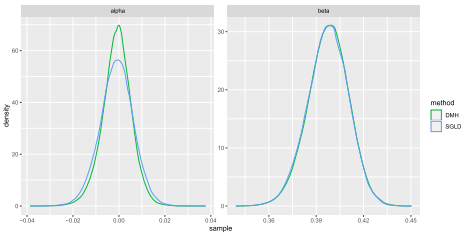
(d) $\alpha = 0.1, \beta = 0.1$



(e) $\alpha = 0.3,\ \beta = 0.3$



(f) $\alpha = 0.3, \beta = 0.3$



(g) $\alpha = 0.5, \beta = 0.5$

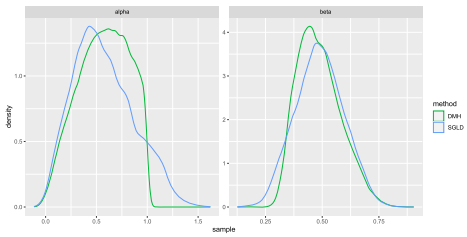


Figure 1: Density plots of posterior samples generated by AUEX, DMH and SGLD under different settings of (α, β) for the spatial autologistic model.

shows that SGLD is able to converge to the true posterior and is thus a valid method for Bayesian analysis of the models with intractable normalizing constants.

4.2 A Simulation Study of ERGM

To assess the effect of the network size N on the performance of the proposed algorithm, we conduct a series of experiments on ERGMs with different values of N. In our experiments, we simulated networks of size $N = 20$, 50, 75, and 100 from the fixed ERGM

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left\{\sum_{k=1}^{2} \theta_k E_k(\boldsymbol{y})\right\}, \tag{4.1}$$

with $\theta_1 = -2$ and $\theta_2 = 0.0042$. For each value of N, 50 networks were simulated using the Gibbs sampler. Specifically, we collected one network after every 10,000 sweeps of the Gibbs sampler, where each dyad undergoes a Metropolis-Hastings update in one sweep of the Gibbs sampler.

The SGLD, DMH, and noisy HMC (Stoehr et al., 2019) algorithms were applied to the simulated networks with a uniform prior $\pi(\boldsymbol{\theta}) \propto 1$ on the parameter space Θ, where Θ is given in the Appendix C of the Supplementary Material (Zhang and Liang, 2023). Detailed parameter settings were also given there. We note particularly that for each algorithm and for each value of N, we run the inner Markov chain for 10 sweeps. For SGLD, this corresponds to fix $m = 10$ for all values of N. The numerical results are summarized in Table 2.

Table 2 shows that the bias of the SGLD estimates decreases as the network size increases, although m is kept as a constant. This is consistent with our theory. In contrast, the estimates of DMH and noisy HMC do not follow this pattern. We suspect that this is due to the acceptance-rejection step involved in DMH and noisy HMC. As the network size increases, the length of the inner Markov chain needs to be extended

method	N	$\hat{\theta}_1 - \theta_1$	$\hat{\theta}_2 - \theta_2$	average CPU time (min)
SGLD	20	−0.1557 (0.0361)	0.0115 (0.0013)	0.1785
	50	0.0986 (0.0413)	−0.0082 (0.0031)	1.1404
	75	−0.0233 (0.0371)	0.0008 (0.0020)	2.7139
	100	0.0044 (0.0417)	0.0003 (0.0016)	4.5732
DMH	20	−0.1568 (0.0364)	0.0121 (0.0012)	0.2010
	50	0.1054 (0.0353)	−0.0084 (0.0026)	1.5187
	75	0.2271 (0.0520)	−0.0122 (0.0027)	3.6809
	100	0.1757 (0.0436)	−0.0069 (0.0017)	6.2842
noisy HMC	20	−0.1299 (0.0415)	0.0134 (0.0019)	10.0616
	50	0.0248 (0.0510)	−0.0015 (0.0040)	19.0405
	75	0.2004 (0.0440)	−0.0109 (0.0022)	8.8749
	100	0.1615 (0.0407)	−0.0063 (0.0016)	10.0393

Table 2: Means and standard deviations (reported in parentheses) of the parameter estimation bias for the networks simulated from the ERGM (4.1) with different values of N.

substantially; otherwise, the resulting auxiliary samples might be more correlated and pre-converged. For noisy HMC, this will lead to a less accurate estimator for the ratio $Z(\boldsymbol{\theta})/Z(\boldsymbol{\theta}')$ and thus the corresponding Metropolis-Hastings ratio. This is similar for DMH, whose estimator for the Metropolis-Hastings ratio also becomes less accurate if the length of the inner Markov chain is not substantially extended as N increases.

Finally, we note that for SGLD, if all samples generated by the inner Markov chain, instead of only those generated at the end of each sweep, are used in estimation of $\mathbb{E}_{\boldsymbol{\theta}}[S(\boldsymbol{y})]$, the resulting estimates of $\boldsymbol{\theta}$ can have a smaller variance but about the same bias compared to those reported in Table 2. Also, we note that SGLD and DMH cost about the same CPU time, while they both are faster than noisy HMC.

4.3 Florentine Business Network

The Florentine families' business network represents the business relations among the 16 prominent Florentine families in early 15th century Europe. The network was originally constructed by Padgett (1994) based on the data from historic documents. We modeled the network using the following ERGM:

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left\{\sum_{k=1}^{2} \theta_k E_k(\boldsymbol{y})\right\},$$

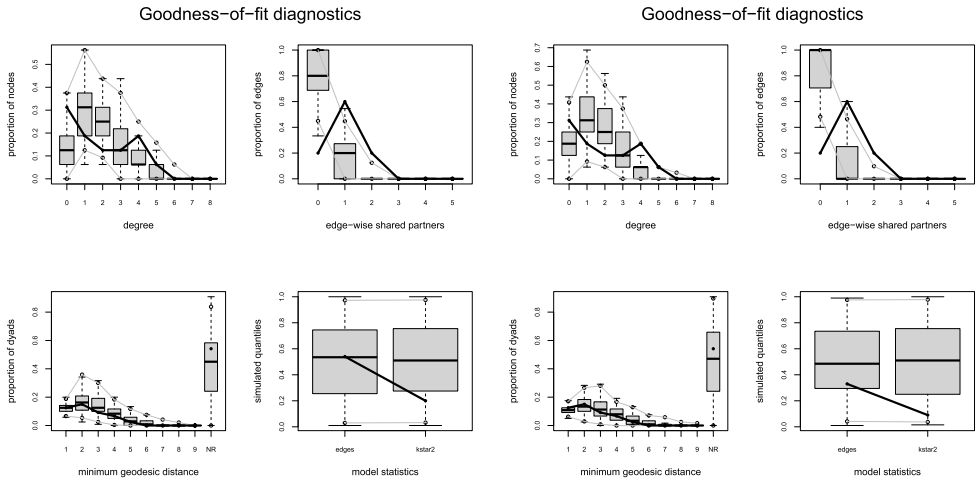
where $E_k(\boldsymbol{y})$ denotes the k-star count.

The SGLD, DMH, noisy Hamiltonian Monte Carlo (noisy HMC) (Stoehr et al., 2019), and AEX (Liang et al., 2016) were applied to this example with the improper prior $\pi(\boldsymbol{\theta}) \propto 1$. Each method was run for 5 times independently with detailed parameter settings given in Appendix C of the Supplementary Material (Zhang and Liang, 2023). These independent runs have exactly the same parameter settings but different random seeds. For SGLD and DMH, we set $m = 4$. The resulting estimates are shown in Table 3. The goodness-of-fit (GOF) plot for the averaged estimate is shown in Figure 2, which was generated using the R package "ergm" (Hunter et al., 2008). In the

method	Edges(θ_1)	k_2-star(θ_2)	CPU time (s)
SGLD	−2.2370 (0.1508)	0.0653 (0.0302)	1.956
AEX	−2.2549 (0.0414)	0.0627 (0.0135)	89.218
DMH	−2.5104 (0.2164)	0.1427 (0.0471)	1.954
noisy HMC	−2.4854 (0.0280)	0.1186 (0.0073)	108.866[a]

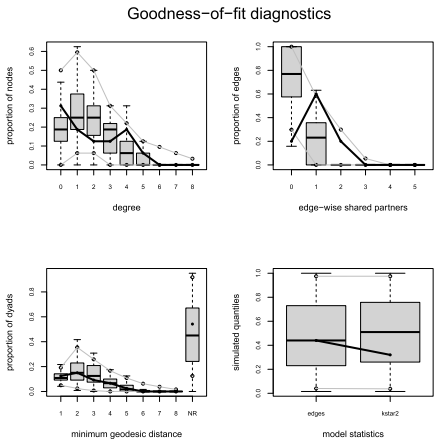
[a] This number counts only the CPU time used for running noisy HMC with the maximum *a posteriori* (MAP) estimate as the initial value; finding the MAP estimate by the stochastic approximation algorithm Robbins and Monro (1951) took 1953.3 seconds.

Table 3: Parameters estimates for Florentine Business Network, where the estimates and standard deviations (reported in parentheses) were obtained by averaging over 5 independent runs.

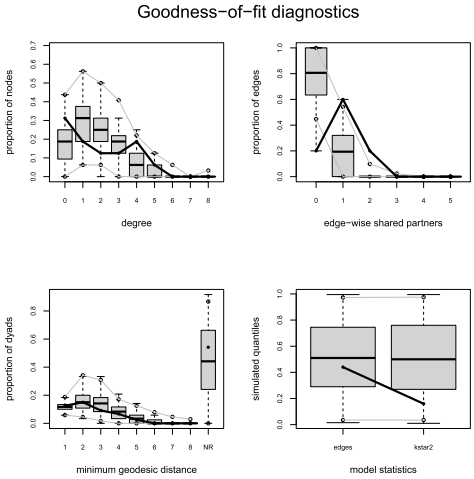


(a) SGLD

(b) AEX



(c) DMH



(d) noisyHMC

Figure 2: Goodness-of-fit plots for Florentine business network.

GOF plot, each boxplot depicts the distribution of a statistic of the networks simulated from the estimated model, and the line links the values of the statistics of the observed network. In particular, the panel of model statistics shows the reproducibility of the sufficient statistics of the observed network by the fitted ERGM, and we often treat it as an indicator for the adequacy of the fitted ERGM. Refer to Appendix B of the Supplementary Material (Zhang and Liang, 2023) for the definitions of the statistics used in the GOF plot and some other basic network statistics.

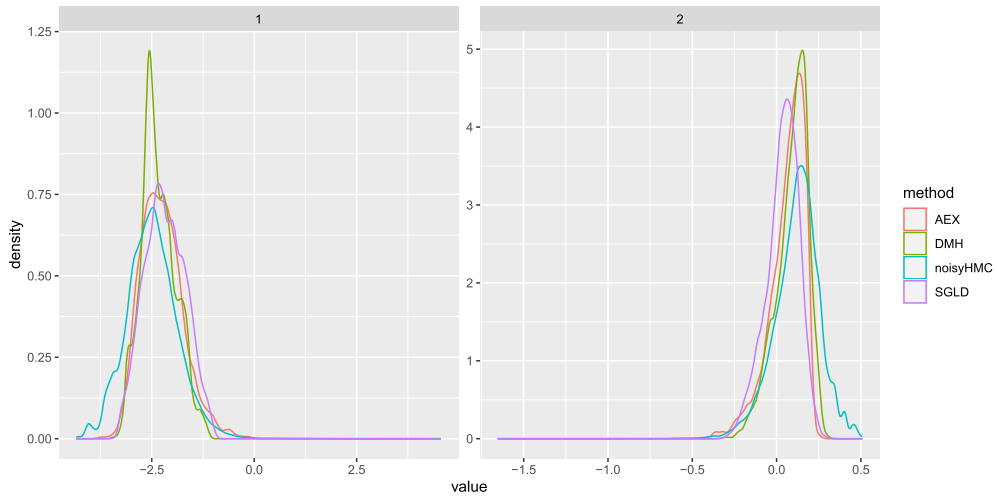


Figure 3: Density plots of posterior samples generated by AEX, DMH, noisy HMC and SGLD, where $m = 100$ for DMH and SGLD.

The numerical results are summarized in Table 3 and Figure 2. For this example, we treat the results of AEX as the standard, as AEX is known to be exact. The comparison shows that SGLD produced about the same estimates as AEX, while costing much less CPU time than the latter; SGLD cost about the same CPU time as DMH, while their estimates are much different; noisy HMC is much expensive under its default setting, while its estimate is reasonably close to that of DMH.

The results in Table 3 are understandable. We note that noisy HMC includes an acceptance-rejection step, where the normalizing constant ratio $Z(\boldsymbol{\theta})/Z(\boldsymbol{\theta}')$ is estimated based on the auxiliary samples simulated by the HMC algorithm. This is somewhat similar to DMH, where the ratio $Z(\boldsymbol{\theta})/Z(\boldsymbol{\theta}')$ is estimated based on a single auxiliary sample simulated by the MH algorithm. In contrast, SGLD does not include such an acceptance-rejection step, for which the auxiliary samples are used for estimating the gradient $\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}|\boldsymbol{y}_N)$ only. As a result, noisy HMC performs more similarly to DMH instead of SGLD. This phenomenon is also evidenced by Table 2.

For a thorough comparison of the four methods, Figure 3 compares the density plots of the posterior samples generated by them. To make the comparison more fair, we have re-run DMH and SGLD with $m = 100$ such that they have comparable (although still shorter) CPU time as AEX and noisy HMC. In terms of posterior densities, SGLD performs more similar to AEX than DMH and noisy HMC.

4.4 Kapferer's Tailor Shop Network

Kapferer's tailor shop network (Kapferer, 1972) describes interactions in a tailor shop in Zambia (Northern Rhodesia) over a period of 10 months, which consists of 39 nodes

method	Edges(θ_1)	k_2-star(θ_2)	GWESP	CPU time (min)
SGLD	−4.0721 (0.1128)	0.0209 (0.0004)	1.0946 (0.0454)	6.019
AEX	−4.1198 (0.1254)	0.0359 (0.0046)	1.0089 (0.0828)	54.044
DMH	−4.1898 (0.2501)	0.0699 (0.0008)	0.7172 (0.1003)	6.863

Table 4: Parameter estimates for Kapferer's Tailor Shop Network, where the estimates and standard deviations (reported in the parentheses) were obtained by averaging over 5 independent runs.

and 223 edges. We modeled the network using the following ERGM:

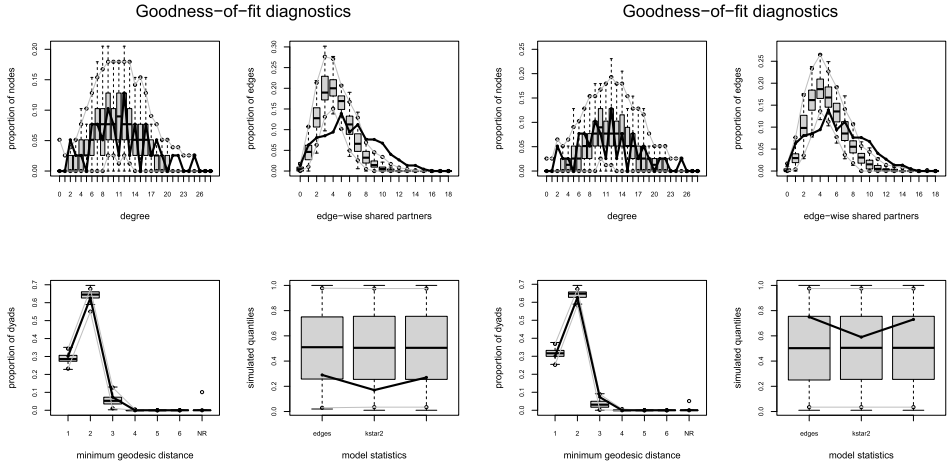
$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left\{\sum_{k=1}^{2} \theta_k E_k(\boldsymbol{y}) + \theta_3 v(\boldsymbol{y}|\tau)\right\}$$

where $\tau = \log 2$, and $v(\boldsymbol{y}|\tau)$ is the geometrically weighted edgewise shared partnership (GWESP) whose definition is given in Appendix B of the Supplementary Material (Zhang and Liang, 2023).

The SGLD, AEX and DMH were applied to this example with the improper prior $\pi(\boldsymbol{\theta}) \propto 1$. The noisy HMC was not applied to this example as its R package (Stoehr et al., 2019) does not implement the GWESP statistic. Each method was run for 5 times independently with detailed settings given in Appendix C of the Supplementary Material (Zhang and Liang, 2023). The resulting parameter estimates are shown in Table 4 and the GOF plots are shown in Figure 4. Since the true parameter values are unknown, we treat the AEX estimates as the standard. The comparison indicates that SGLD outperforms DMH; SGLD led to more accurate parameter estimates and better GOF fitting than DMH for this example. Again, SGLD cost much less CPU time than AEX, although their estimates are similar. In summary, SGLD method is as fast as DMH, while providing similar accurate estimates as AEX.

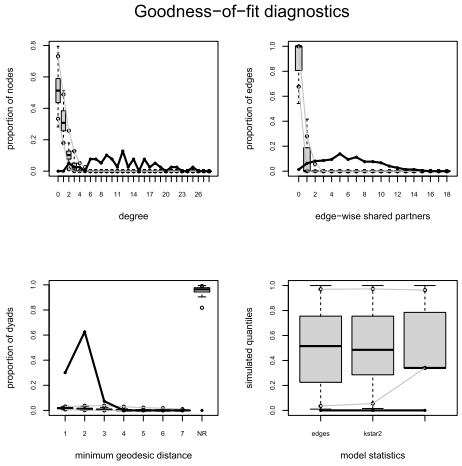
4.5 Student Friendship Network

To investigate the performance of the proposed SGLD method on higher dimensional models, we consider the student friendship network which was collected in the first wave (1994–1995) of the National Longitudinal Study of Adolescent to Adult Health (Add Health). The detailed description of the study and the dataset can be found at https://addhealth.cpc.unc.edu. The entire dataset was collected from 86 schools with 90,118 students. In this paper, we focused on the data from school 10 with 205 students. Therefore, the network consists of 205 nodes. There are three factors for each node: *grade*, *race*, and *sex*. To include nodal covariates in the ERGM of this network,



(a) SGLD

(b) AEX



(c) DMH

Figure 4: GOF plots for the Kapferer's Tailor Shop Network.

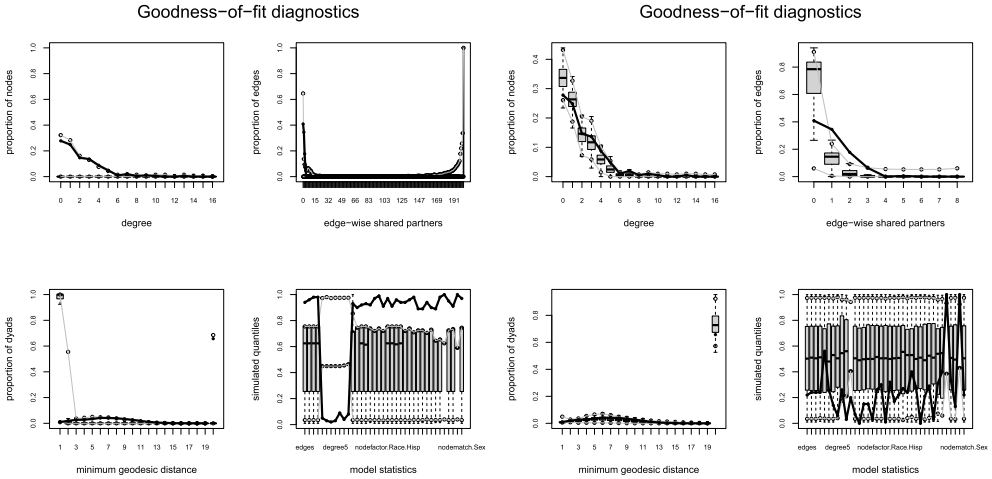
we consider the model:

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left\{ \sum_{k=1}^{3} \theta_k E_k(\boldsymbol{y}) + \theta_4 T(\boldsymbol{y}) + \sum_{k=2}^{8} \theta_{k+3} D_k(\boldsymbol{y}) + \theta_{12} u((\boldsymbol{y}|\tau) \right.$$
$$+ \sum_{k=1}^{10} \theta_{k+12} h_{\mathrm{NF},k}(\boldsymbol{y}, \boldsymbol{X}) + \sum_{k=1}^{3} \theta_{k+22} h_{\mathrm{AD},k}(\boldsymbol{y}, \boldsymbol{X})$$
$$\left. + \sum_{k=1}^{11} \theta_{k+25} h_{\mathrm{DNF},k}(\boldsymbol{y}, \boldsymbol{X}) + \theta_{37} h_{\mathrm{UHF},1}(\boldsymbol{y}, \boldsymbol{X}) \right\}, \tag{4.2}$$

where $\boldsymbol{y}$ denotes the adjacency matrix of the network, $\boldsymbol{X}$ denotes the matrix of factors of all nodes, $D_k(\boldsymbol{y})$ denotes the number of nodes in $\boldsymbol{y}$ whose degree is k, $u(\boldsymbol{y}|\tau)$ denotes the geometrically weighted degree (GWD) statistic of $\boldsymbol{y}$, $h_{\mathrm{AD},k}(\boldsymbol{y}, \boldsymbol{X})$ denotes the nodal factor effect of the network, $h_{\mathrm{AD},k}(\boldsymbol{y}, \boldsymbol{X})$ denotes the absolute difference factor effect of the network, $h_{\mathrm{DNF},k}(\boldsymbol{y}, \boldsymbol{X})$ denotes the differential homophily factor effect of the network, and $h_{\mathrm{UHF},1}(\boldsymbol{y}, \boldsymbol{X})$ denotes the uniform homophily factor effect of the network. The definitions of those statistics and nodal covariates can be found in Jin and Liang (2013).

Grade is an ordinal factor with six levels indicating the grade (7–12) of a student. For *grade*, we include the nodal factor effect for grade 8–12, the differential homophily factor effect for grade 7–12, and the absolute different effects with $C = 1, 2, 3$ in the model. *Race* is a nominal factor with five levels indicating the race (*white, black, Hispanic, native American*, and *others*) of a student. For *race*, we include the nodal factor effect for all levels but *others* and the differential homophily factor effect for all levels in the model. *Sex* is a nominal factor with two levels indicating the gender (*male*, and *female*) of a student. For *sex*, we include the nodal factor effect for *female* and the uniform homophily factor effect in the model.

SGLD and DMH were applied to this example with a multivariate Gaussian prior $\pi(\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{\theta}\|_2^2\right)$ imposed on $\boldsymbol{\theta}$. The Gibbs sampler was used to generate auxiliary networks at each iteration. Refer to Appendix C of the Supplementary Material (Zhang and Liang, 2023) for the settings of other hyperparameters. Table 5 shows the parameter estimates produced by the two methods. Figure 5 shows the GOF plots of their estimates. Figure 5 indicates that the estimates produced by SGLD fit the network much better than those produced by DMH, while Table 5 indicates that most of the estimates produced by the two methods have the same sign.



(a) DMH (b) SGLD

Figure 5: GOF plots for the Student Friendship Network.

Coefficient	DMH	SGLD
Edge counts	−1.0789 (0.0918)	−4.3493 (0.0673)
k_2-star	−0.3024 (0.0439)	−0.1264 (0.0216)
k_3-star	0.0276 (0.0093)	0.0012 (0.0030)
Triangle	2.0004 (0.0681)	1.6922 (0.0151)
Degree (2)	−0.5255 (0.0696)	−0.3924 (0.0238)
Degree (3)	−0.3112 (0.1248)	−0.2250 (0.0470)
Degree (4)	−0.3994 (0.1024)	−0.2956 (0.0552)
Degree (5)	−0.3091 (0.1293)	−0.4355 (0.0603)
Degree (6)	−0.7418 (0.1538)	−1.1688 (0.0474)
Degree (7)	−0.2040 (0.1828)	−0.2621 (0.0387)
Degree (8)	−0.2557 (0.1393)	−0.7213 (0.0475)
GWD ($\tau = 0.25$)	−1.1459 (0.1525)	−0.9131 (0.0709)
NF (*grade*: 8)	−0.4021 (0.1015)	−0.0053 (0.0438)
NF (*grade*: 9)	−0.0973 (0.0541)	0.6661 (0.0250)
NF (*grade*: 10)	0.2500 (0.1442)	0.9473 (0.0292)
NF (*grade*: 11)	−0.1670 (0.0805)	0.6462 (0.0323)
NF (*grade*: 12)	0.2386 (0.1308)	1.1464 (0.0166)
NF (*race*: *white*)	−0.8079 (0.0353)	−0.3115 (0.0373)
NF (*race*: *black*)	−0.0257 (0.0802)	0.3568 (0.0405)
NF (*race*: *Hispanic*)	−1.3333 (0.0770)	−0.8695 (0.0352)
NF (*race*: *native American*)	−1.3399 (0.1222)	−0.8631 (0.0412)
NF (*sex*: *female*)	0.0708 (0.0458)	0.0896 (0.0043)
AD (*grade*, $C = 1$)	−1.1932 (0.0587)	−0.7964 (0.0198)
AD (*grade*, $C = 2$)	−1.1378 (0.1723)	−0.7057 (0.0114)
AD (*grade*, $C = 3$)	−1.3530 (0.0690)	−0.9116 (0.0093)
DHF (*grade*: 7)	0.8620 (0.0629)	2.5629 (0.0348)
DHF (*grade*: 8)	1.3197 (0.0376)	2.2915 (0.0517)
DHF (*grade*: 9)	0.6922 (0.0694)	1.0148 (0.0103)
DHF (*grade*: 10)	0.2927 (0.0450)	0.5750 (0.0233)
DHF (*grade*: 11)	1.2738 (0.0441)	1.5025 (0.0350)
DHF (*grade*: 12)	0.7811 (0.2581)	1.0333 (0.0276)
DHF (*race*: *white*)	0.1777 (0.1570)	0.0873 (0.0085)
DHF (*race*: *black*)	−0.2814 (0.0832)	−0.5023 (0.0558)
DHF (*race*: *Hispanic*)	0.5389 (0.0477)	0.5317 (0.0060)
DHF (*race*: *native American*)	1.0661 (0.3765)	1.0832 (0.0142)
DHF (*race*: *others*)	0.0288 (0.1493)	−0.2888 (0.0844)
UHF (*sex*)	0.4408 (0.0388)	0.5113 (0.0017)
CPU time (min)	616.484	626.285

Table 5: Parameters estimates for the student friendship network, where the estimates and standard deviations (reported in the parentheses) were obtained by averaging over 5 independent runs.

5 LastFM Asia Social Network

This section explores the performance of the SGLD method on very large networks. The LastFM Asia social network is a network of the Asian users of LastFM, a music website, collected from the public API in March 2020 (Rozemberczki and Sarkar, 2020). The network consists of 7,624 nodes and 27,806 edges, where the node represent users and the edges represent mutual follower relationships between the users. We modeled the network using an ERGM with 4 parameters:

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left\{\sum_{k=1}^{2} \theta_k E_k(\boldsymbol{y}) + \theta_3 T(\boldsymbol{y}) + \theta_4 u(\boldsymbol{y}|\tau)\right\}, \tag{5.1}$$

where $E_k(\boldsymbol{y})$ denotes k-star counts, $T(\boldsymbol{y})$ denotes triangle counts, $\tau = \log(2)$, and $u(\boldsymbol{y}|\tau)$ denotes the geometrically weighted degree (GWD) statistic whose definition is given in Appendix B of the Supplementary Material (Zhang and Liang, 2023).

For the model (5.1), we let $\boldsymbol{\theta}$ be subject to a multivariate Gaussian prior $\pi(\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{\theta}\|_2^2\right)$. Since the network is large, the TNT sampler was used to generate auxiliary networks. At each iteration, the TNT sampler proceeds by updating $\tilde{m}$ dyads with edges and $\tilde{m}$ dyads without edges, and the last network was used in estimating the expectation $\mathbb{E}_{\boldsymbol{\theta}} S(\boldsymbol{Y})$.

To explore the effect of $\tilde{m}$ on the convergence of SGLD, we tried three different values of $\tilde{m} = 500$, 1000 and 2000. Figure 6 shows the trace plots of posterior samples produced by SGLD with different values of $\tilde{m}$. It indicates that the choice of $\tilde{m}$ affects the speed of convergence; a larger value of $\tilde{m}$ can significantly accelerates the convergence of the simulation (in terms of iterations). However, SGLD is able to converge to the same estimate with different values of $\tilde{m}$ if the run is sufficiently long. This is consistent with Remark 3.2.

Next, we re-ran SGLD with the settings $\tilde{m} = 500$, 1000 and 2000 for this example. Correspondingly, we set the total numbers of iterations $T = 250,000$, 150,000 and 100,000 such that each run cost about the same CPU time under each setting. Table 6 summarizes the resulting parameter estimates, where SGLD were run 5 times independently under each setting and the parameters were estimated by averaging over the last 50,000 iterations in each run. Figure 7 shows the GOF plots obtained under each setting of $\tilde{m}$. The comparison shows that a larger value of $\tilde{m}$ tends to produce more stable parameter estimates. This is consistent with Remark 3.3: the best choice of $\tilde{m}$ might depend on the network size.

Finally, we explore the importance of each component of the sufficient statistics in (5.1) in modeling the network. We imposed a shrinkage prior on $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)$, where θ_k's are *a priori* independent and each follows a mixture Gaussian prior

$$\pi(\theta_k) = \lambda \mathcal{N}(0, \sigma_1^2) + (1 - \lambda)\mathcal{N}(0, \sigma_0^2), \tag{5.2}$$

where σ_0^2 is very small, σ_1^2 is relatively large, and λ specifies the mixture probability of the two components. Since the log-prior $\log \pi(\boldsymbol{\theta})$ is differentiable with respect to $\boldsymbol{\theta}$

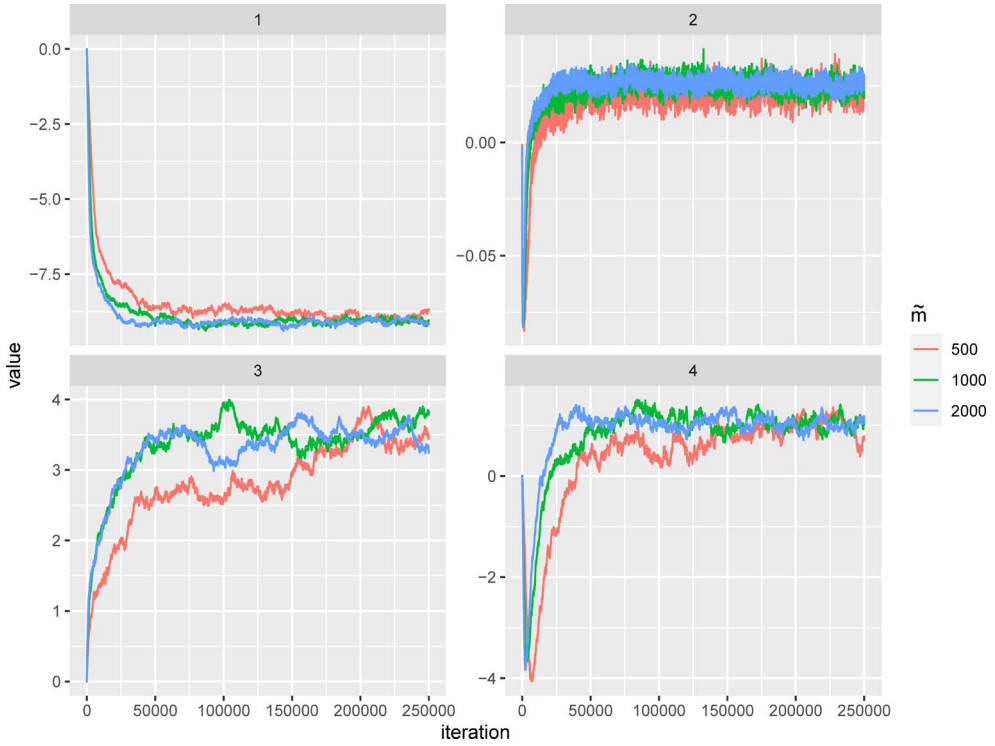
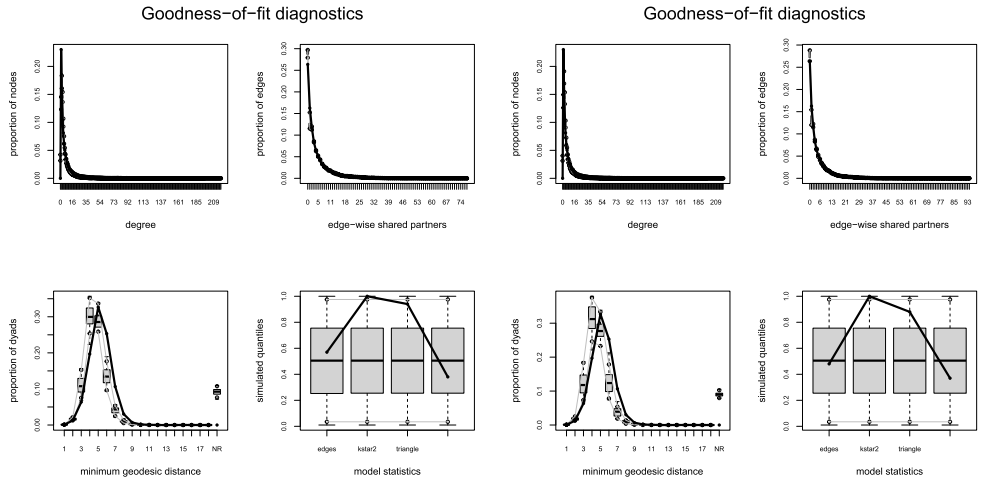


Figure 6: Trace plots of SGLD produced with different values of $\tilde{m}$ in the TNT sampler, where plot i corresponds to the parameter θ_i of the model (5.1).

terms	SGLD ($\tilde{m} = 500$)	SGLD ($\tilde{m} = 1000$)	SGLD ($\tilde{m} = 2000$)	SGLD ($\tilde{m} = 500$, three-parameter model: $\theta_1, \theta_3, \theta_4$)
Edges (θ_1)	−8.8886 (0.0396)	−9.0905 (0.0255)	−9.0889 (0.0175)	−8.2744 (0.0335)
k_2-star (θ_2)	0.0229 (0.0003)	0.0262 (0.0004)	0.0261 (0.0003)	–
Triangles (θ_3)	3.2966 (0.1291)	3.4597 (0.0751)	3.4433 (0.0679)	3.2166 (0.0312)
GWD (θ_4)	0.8640 (0.0659)	1.0533 (0.0437)	1.0312 (0.0236)	0.3043 (0.0814)
CPU time	526.245 min	486.999 min	516.831 min	504.908 min

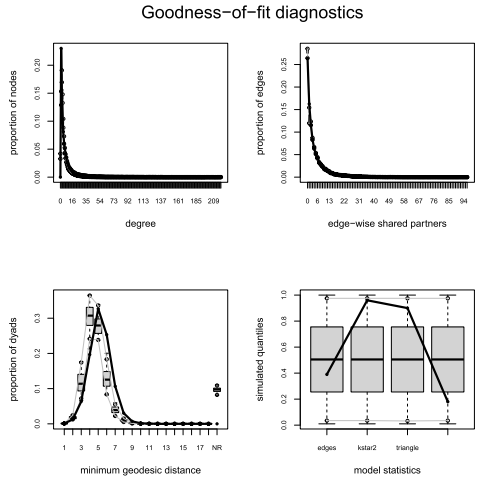
Table 6: Parameter estimates produced by SGLD for the LastFM Asia Social Network, where the estimates and standard errors (reported in the parentheses) were obtained by averaging over 5 independent runs.

and the network is large, the conditions (A.1)–(A.3) can still be satisfied as discussed in Section 3. For this example, we set $\lambda = 0.5$, $\sigma_1 = 10$ and tried three different values of $\sigma_0 = 0.01$, 0.001, and 0.0001.

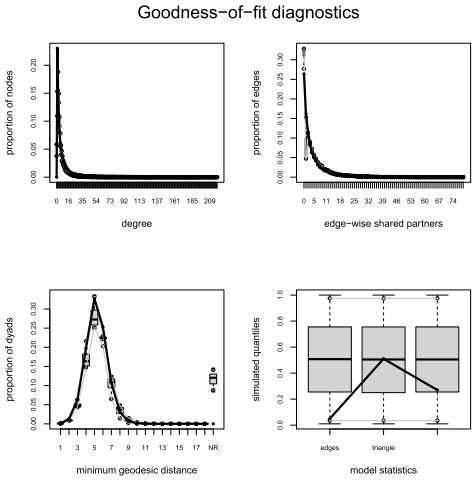


(a) SGLD ($\tilde{m} = 500$)

(b) SGLD ($\tilde{m} = 1000$)



(c) SGLD ($\tilde{m} = 2000$)



(d) SGLD ($\tilde{m} = 500$, three-parameter model)

Figure 7: GOF plots produced by SGLD for the LastFM Asia network with different choices of $\tilde{m}$ in the TNT sampler.

With such a shrinkage prior, variable selection can be done for the model (5.1) using the marginal inclusion probability approach, see e.g. Barbieri and Berger (2004) and Liang et al. (2013). In this approach, variable selection can proceeds as follows. Let $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(T)}$ denote T samples from the posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$. Let $\theta_k^{(i)}$ denotes the k-th component of $\boldsymbol{\theta}^{(i)}$, which corresponds to the component $S_k(\boldsymbol{y})$ of the

sufficient statistics. Let $\delta_k^{(i)} = I\left(|\theta_k^{(i)}| > \frac{\sqrt{2}\sigma_0\sigma_1}{\sqrt{\sigma_1^2-\sigma_0^2}}\sqrt{\log\left(\frac{1-\lambda}{\lambda}\frac{\sigma_1}{\sigma_0}\right)}\right)$ be the indicator that $\theta_k^{(i)}$ more likely belongs to the component $N(0,\sigma_1^2)$ than the component $N(0,\sigma_0^2)$. Then $\hat{p}_k = \sum_{i=1}^T \delta_k^{(i)}/T$ provides a consistent estimator for the marginal inclusion probability of the statistic $S_k(\boldsymbol{y})$, and variable selection for the model (5.1) can be made accordingly. Alternatively, we can first average $\boldsymbol{\theta}^{(i)}$'s to get a consistent estimator of $\boldsymbol{\theta}$ and then select variables for the model based on the indicators $\delta_k = I\left(|\hat{\theta}_k| > \frac{\sqrt{2}\sigma_0\sigma_1}{\sqrt{\sigma_1^2-\sigma_0^2}}\sqrt{\log\left(\frac{1-\lambda}{\lambda}\frac{\sigma_1}{\sigma_0}\right)}\right)$, where $\hat{\theta}_k = \sum_{i=1}^T \theta_k^{(i)}/T$.

By the alternative approach described above, a 3-component model $(E_1(\boldsymbol{y}), T(\boldsymbol{y}), u(\boldsymbol{y}|\tau))$ was selected if we set $\sigma_0 = 0.01$ in the prior (5.2), and the full model was selected if we set $\sigma = 0.001$ and 0.0001. The resulting parameter estimates are shown in Table 6 and GOF plot is shown in Figure 7. A comparison with the GOF plots of the full model indicates that the reduced model fits the network even better, particularly in minimum geodesic distance.

It is interesting to note that the model statistics panels of the GOF plots (Figure 7(a)–(c)) indicates that the statistic kstar2 ($E_2(\boldsymbol{y})$) is poorly reproduced by the fitted full model. Then, with regularization, this statistic was dropped off as shown in Figure 7(d). In general, if the model statistics panel shows a poor fit, then we might need to modify the model by adding to or deleting from the model some statistics.

6 Conclusion

In this paper, we have proposed to use the SGLD algorithm for Bayesian analysis of ERGMs. We proved that SGLD converges to the true posterior in 2-Wasserstein distance as the network size $N \to \infty$ and the iteration number $t \to \infty$ regardless of the length of the inner Markov chain performed at each iteration, provided the model size p grows at a rate of $o(N^\kappa)$. The SGLD algorithm also enjoys its scalability with respect to the model size p as a nice property carried over from stochastic gradient MCMC algorithms. We tested the performance of the proposed algorithm on simulated and real networks.

Compared with the existing exact algorithms such as auxiliary variable MCMC (Møller et al., 2006), the exchange algorithm (Murray et al., 2006), and AEX (Liang et al., 2016), the proposed algorithm is much more efficient for large-scale networks. Compared to the existing inexact algorithms like DMH, the proposed method is more stable and accurate while costing about the same CPU time.

Further efforts can be made on developing more efficient algorithms for auxiliary network simulations, the most time consuming part of the SGLD algorithm. Parallel computing could be a good direction to follow. We will also consider to apply the proposed algorithm to more general network models such as mixture ERGMs (Salter-Townshend and Brendan Murphy, 2015) and more sophisticated dynamic network models (Kim et al., 2018), and many other models with intractable normalizing constants, such as spatial autologistic models (Besag, 1974), Gaussian Markov random fields (Besag and

Moran, 1975), and spatial interaction point process models (Goldstein et al., 2015). For the spatial interaction point process model, we particularly note that although $\nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta})$ cannot be written in the form $\mathbb{E}_{\boldsymbol{\theta}}[S(\boldsymbol{Y})]$ as in equation (2.1) and the sample space for the point process is continuous, the SGLD algorithm can still be directly applied. For example, let's consider a pairwise interaction point process model with the density function $p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{\exp\left(-\sum_{i=1}^{n}\sum_{j=i+1}^{n}\phi(\|x_i - x_j\|;\boldsymbol{\theta})\right)}{Z(\boldsymbol{\theta})}$, where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and n denotes the number of points. For this model, we have

$$\nabla_{\boldsymbol{\theta}} \log(Z(\boldsymbol{\theta})) = -\sum_{i=1}^{n}\sum_{j=i+1}^{n} \mathbb{E}_{\boldsymbol{\theta}}[\nabla_{\boldsymbol{\theta}}\phi(|x_i - x_j\|, \boldsymbol{\theta})],$$

and, therefore, $\nabla_{\boldsymbol{\theta}} \log(Z(\boldsymbol{\theta}))$ can still be estimated using the auxiliary samples simulated from the point process at the parameter point $\boldsymbol{\theta}$.

Supplementary Material

Supplement to "Bayesian Analysis of Exponential Random Graph Models Using Stochastic Gradient Markov Chain Monte Carlo" (DOI: 10.1214/23-BA1364SUPP; .pdf). Appendix A: Proof of theorems. Appendix B: Recursive formulas for sufficient statistics calculation. Appendix C: Experimental settings.

References

Alquier, P., Friel, N., Everitt, R., and Boland, A. (2016). "Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels." *Statistics and Computing*, 26(1-2): 29–47. MR3439357. doi: https://doi.org/10.1007/s11222-014-9521-x. 596, 597, 601

Atchade, Y. F., Lartillot, N., and Robert, C. P. (2013). "Bayesian computation for intractable normalizing constants." *Brazilian Journal of Statistics*, 27: 416–436. MR3105037. doi: https://doi.org/10.1214/11-BJPS174. 596, 597

Barbieri, M. and Berger, J. (2004). "Optimal Predictive Model Selection." *The Annals of Statistics*, 32: 870–897. MR2065192. doi: https://doi.org/10.1214/009053604000000238. 615

Besag, J. (1974). "Spatial interaction and the statistical analysis of lattice systems." *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2): 192–225. MR0373208. 616

Besag, J. E. and Moran, P. A. (1975). "On the estimation and testing of spatial interaction in Gaussian lattice processes." *Biometrika*, 62(3): 555–562. MR0391451. doi: https://doi.org/10.1093/biomet/62.3.555. 616

Bhatia, K., Ma, Y.-A., Dragan, A. D., Bartlett, P. L., and Jordan, M. I. (2019). "Bayesian Robustness: A Nonasymptotic Viewpoint." *arXiv preprint arXiv:1907.11826*. 597

Caimo, A. and Friel, N. (2011). "Bayesian inference for exponential random graph models." *Social Networks*, 33: 41–55. 596

Childs, A. M., Patterson, R. B., and MacKay, D. J. (2001). "Exact sampling from non-attractive distributions using summary states." *Physics Review E*, 63: 036113. 602, 603

Dalalyan, A. S. and Karagulyan, A. (2019). "User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient." *Stochastic Processes and their Applications*, 129(12): 5278 – 5311. URL http://www.sciencedirect.com/science/article/pii/S0304414918304824 MR4025705. doi: https://doi.org/10.1016/j.spa.2019.02.016. 597

Durmus, A., Majewski, S., and Miasojedow, B. (2019). "Analysis of Langevin Monte Carlo via Convex Optimization." *J. Mach. Learn. Res.*, 20: 73:1–73:46. MR3960927. 597

Everitt, R. G. (2012). "Bayesian parameter estimation for latent Markov random fields and social networks." *Journal of Computational and Graphical Statistics*, 21: 940–960. MR3005805. doi: https://doi.org/10.1080/10618600.2012.687493. 596, 597

Fellows, I. and Handcock, M. (2017). "Removing Phase Transitions from Gibbs Measures." In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, 289–297. Fort Lauderdale, FL, USA: PMLR. 600

Geman, S. and Geman, D. (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6: 721–741. 599

Geyer, C. J. and Thompson, E. A. (1992). "Constrained Monte Carlo Maximum Likelihood for Dependent Data." *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(3): 657–699. URL http://www.jstor.org/stable/2345852 MR1185217. 596

Gibbs, A. and Su, F. (2002). "On Choosing and Bounding Probability Metrics." *International Statistical Review*, 70(3): 419–435. 599

Goldstein, J., Haran, M., Simeonov, I., Fricks, J., and Chiaromonte, F. (2015). "An attraction–repulsion point process model for respiratory syncytial virus infections." *Biometrics*, 71(2): 376–385. MR3366242. doi: https://doi.org/10.1111/biom.12267. 617

Hunter, D. R., Goodreau, S. M., and Handcock, M. S. (2008). "Goodness of fit of social network models." *Journal of the American Statistical Association*, 103: 248–258. MR2394635. doi: https://doi.org/10.1198/016214507000000446. 606

Jin, I. H. and Liang, F. (2013). "Fitting social network models using varying truncation stochastic approximation MCMC algorithm." *Journal of computational and graphical statistics*, 22(4): 927–952. MR3173750. doi: https://doi.org/10.1080/10618600.2012.680851. 611

Jin, I. H. and Liang, F. (2014). "Use of SAMC for Bayesian Analysis of Statistical Models with Intractable Normalizing Constants." *Computational Statistics and Data Analysis*, 71: 402–416. MR3131979. doi: https://doi.org/10.1016/j.csda.2012.07.005. 596

Kapferer, B. (1972). *Strategy and transaction in an African factory*. Manchester: Manchester University Press. 608

Kim, B., Lee, K. H., Xue, L., and Niu, X. (2018). "A review of dynamic network models with latent variables." *Statistics surveys*, 12: 105. MR3850294. doi: https://doi.org/10.1214/18-SS121. 616

Liang, F. (2007). "Continuous contour Monte Carlo for marginal density estimation with an application to a spatial statistical models." *Journal of Computational and Graphical Statistics*, 16: 608–632. MR2351082. doi: https://doi.org/10.1198/106186007X238459. 596, 597

Liang, F. (2010). "A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants." *Journal of Statistical Computing and Simulation*, 80: 1007–1022. MR2742519. doi: https://doi.org/10.1080/00949650902882162. 596, 598, 603

Liang, F. and Jin, I. H. (2013). "A Monte Carlo Metropolis-Hastings Algorithm for Sampling from Distributions with Intractable Normalizing Constants." *Neural Computation*, 25: 2199–2234. MR3100001. doi: https://doi.org/10.1162/NECO_a_00466. 596, 597

Liang, F., Jin, I. H., Song, Q., and Liu, J. S. (2016). "An adaptive exchange algorithm for sampling from distributions with intractable normalizing constants." *Journal of the American Statistical Association*, 111(513): 377–393. MR3494666. doi: https://doi.org/10.1080/01621459.2015.1009072. 597, 603, 606, 616

Liang, F., Song, Q., and Yu, K. (2013). "Bayesian Subset Modeling for High-Dimensional Generalized Linear Models." *Journal of the American Statistical Association*, 108(502): 589–606. MR3174644. doi: https://doi.org/10.1080/01621459.2012.761942. 615

Møller, J., Pettitt, A. N., Reeves, R., and Berthelsen, K. K. (2006). "An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants." *Biometrika*, 93(2): 451–458. MR2278096. doi: https://doi.org/10.1093/biomet/93.2.451. 596, 616

Morris, M., Handcock, M., and Hunter, D. (2008). "Specification of Exponential-Family Random Graph Models: Terms and Computational Aspects." *Journal of statistical software*, 24 4: 1548–7660. 599

Murray, I., Ghahramani, Z., and MacKay, D. J. (2006). "MCMC for doubly-intractable distributions." In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence (UAI)*, 359–366. 596, 603, 616

Padgett, J. (1994). *Marriage and elite structure in Renaissance Florence, 1282–1500*. Social Science History Association. 606

Park, J. and Haran, M. (2018). "Bayesian inference in the presence of intractable normalizing functions." *Journal of the American Statistical Association*, 113(523): 1372–1390. MR3862364. doi: https://doi.org/10.1080/01621459.2018.1448824. 596

Propp, J. G. and Wilson, D. B. (1996). "Exact Sampling with Coupled Markov Chains and Applications to Statistical Mechanics." *Random Structures and Algorithms*, 9: 223–252. MR1611693. doi: https://doi.org/10.1002/(SICI)1098-2418(199608/09)9:1/2<223::AID-RSA14>3.3.CO;2-R. 596

Riggan, W. B., Creason, J. P., Nelson, W. C., Manton, K. G., Woodbury, M. A., Stallard, E., Pellom, A. C., and Beaubier, J. (1987). *U.S. Cancer Mortality Rates and Trends, 1950–1979. (Vol. IV: Maps)*. U.S. Government Printing Office.: U.S. Government Printing Office. 602

Robbins, H. and Monro, S. (1951). "A Stochastic Approximation Method." *The Annals of Mathematical Statistics*, 22(3): 400–407. MR0042668. doi: https://doi.org/10.1214/aoms/1177729586. 606

Robins, G., Pattison, P., Kalish, Y., and Lusher, D. (2007a). "An introduction to exponential random graph (p*) models for social networks." *Social Networks*, 29(2): 173–191. Special Section: Advances in Exponential Random Graph (p*) Models. URL http://www.sciencedirect.com/science/article/pii/S0378873306000372 595

Robins, G. E., Snijers, T. A. B., Wang, P., Handcock, M. S., and Pattison, P. E. (2007b). "Recent development in exponential random graph models for social networks." *Social Networks*, 29: 192–215. MR2873466. 595

Rozemberczki, B. and Sarkar, R. (2020). "Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models." In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, 1325–1334. ACM. 613

Salter-Townshend, M. and Brendan Murphy, T. (2015). "Role analysis in networks using mixtures of exponential random graph models." *Journal of Computational and Graphical Statistics*, 24(2): 520–538. MR3357393. doi: https://doi.org/10.1080/10618600.2014.923777. 616

Song, Q., Sun, Y., Ye, M., and Liang, F. (2020). "Extended stochastic gradient Markov chain Monte Carlo for large-scale Bayesian variable selection." *Biometrika*, in press. MR4186501. doi: https://doi.org/10.1093/biomet/asaa029. 597, 602

Stoehr, J., Benson, A., and Friel, N. (2019). "Noisy Hamiltonian Monte Carlo for doubly intractable distributions." *Journal of Computational and Graphical Statistics*, 28(1): 220–232. MR3939384. doi: https://doi.org/10.1080/10618600.2018.1506346. 596, 597, 605, 606, 609

Teh, W., Thiery, A., and Vollmer, S. (2016). "Consistency and fluctuations for stochastic gradient Langevin dynamics." *Journal of Machine Learning Research*, 17: 1–33. MR3482927. 602

Welling, M. and Teh, Y. W. (2011). "Bayesian learning via stochastic gradient Langevin dynamics." In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 681–688. 597, 598

Zhang, Q. and Liang, F. (2023). "Supplementary Material for " Bayesian analysis of exponential random graph models using stochastic gradient Markov chain Monte Carlo"." *Bayesian Analysis*. doi: https://doi.org/10.1214/23-BA1364SUPP. 599, 600, 601, 602, 603, 605, 606, 607, 609, 611, 613

Acknowledgments

The authors thank the editor, associate editor and three referees for their constructive comments which have led to significant improvement of this paper.