LETTER Communicated by Chuanhai Liu

A Monte Carlo Metropolis-Hastings Algorithm for Sampling from Distributions with Intractable Normalizing Constants

Faming Liang
fliang@stat.tamu.edu.
Department of Statistics, Texas A&M University, College Station, TX 77843, U.S.A.

Ick-Hoon Jin
ijin@mdanderson.org
Department of Biostatistics, University of Texas, M. D. Anderson Cancer Center, Houston, TX 77030, U.S.A.

Simulating from distributions with intractable normalizing constants has been a long-standing problem in machine learning. In this letter, we propose a new algorithm, the Monte Carlo Metropolis-Hastings (MCMH) algorithm, for tackling this problem. The MCMH algorithm is a Monte Carlo version of the Metropolis-Hastings algorithm. It replaces the unknown normalizing constant ratio by a Monte Carlo estimate in simulations, while still converges, as shown in the letter, to the desired target distribution under mild conditions. The MCMH algorithm is illustrated with spatial autologistic models and exponential random graph models. Unlike other auxiliary variable Markov chain Monte Carlo (MCMC) algorithms, such as the Møller and exchange algorithms, the MCMH algorithm avoids the requirement for perfect sampling, and thus can be applied to many statistical models for which perfect sampling is not available or very expensive. The MCMH algorithm can also be applied to Bayesian inference for random effect models and missing data problems that involve simulations from a distribution with intractable integrals.

1 Introduction

In scientific computation, one often encounters problems of making inference for a model whose likelihood function contains an intractable normalizing constant. Examples of such models include the autologistic model used in ecology study (Wu & Huffer, 1997), the Potts model used in image analysis (Hurn, Husby, & Rue, 2003), the autonormal model used in

*Liang is Professor, Department of Statistics, Texas A&M University, College Station, TX 77843-3143. Tel: (979)-845-8885; email: fliang@stat.tamu.edu. Jin is Postdoctoral Fellow, Department of Biostatistics, The University of Texas MD Anderson Cancer Center, Houston, TX 77030-4009, USA; email: ijin@mdanderson.org.

Neural Computation **25, 2199–2234** (2013)
doi:10.1162/NECO_a_00466

agriculture experiments (Besag, 1974), and the exponential random graph model used in social network study (Snijders, Pattison, Robins, & Handcock, 2006), among others. Under the Bayesian framework, the problems can be posed as follows. Suppose we have a data set X generated from a statistical model with the likelihood function

$$f(x|\theta) = \frac{g(x,\theta)}{\kappa(\theta)}, \quad x \in \mathcal{X},\ \theta \in \Theta, \tag{1.1}$$

where θ is the parameter and $\kappa(\theta)$ is the normalizing constant that depends on θ and is not available in closed form. Let $\pi(\theta)$ denote the prior density imposed on θ. The posterior density of θ is then given by

$$\pi(\theta|x) \propto \frac{1}{\kappa(\theta)} g(x,\theta)\pi(\theta). \tag{1.2}$$

Since the closed form of $\kappa(\theta)$ is not available, inference for θ poses a great challenge on the current statistical methods.

The Metropolis-Hastings (MH) algorithm cannot be applied to simulate from $\pi(\theta|x)$, because the acceptance probability would involve an unknown ratio $\kappa(\theta)/\kappa(\vartheta)$, where ϑ denotes the proposed value. To circumvent this difficulty, various approximation methods to the likelihood function or the normalizing constant function have been proposed in the literature. Besag (1974) proposed to approximate the likelihood function by a pseudo-likelihood function that is tractable. The method is easy to use, but it typically performs less well for the models for which neighboring dependence is strong. Geyer and Thompson (1992) proposed an importance sampling–based approach to approximation $\kappa(\theta)$, which can be briefly described as follows. Let θ^* denote an initial guess of θ. Let $y_1, \ldots, y_m$ denote random samples simulated from $f(y|\theta^*)$, which can be obtained via a Markov chain Monte Carlo (MCMC) simulation. Then

$$\log f_m(x|\theta) = \log(g(x,\theta)) - \log(\kappa(\theta^*)) - \log\left(\frac{1}{m}\sum_{i=1}^{m}\frac{g(y_i,\theta)}{g(y_i,\theta^*)}\right) \tag{1.3}$$

approaches $\log f(x|\theta)$ as $m \to \infty$. The estimator $\hat{\theta} = \arg\max_\theta \log f_m(x|\theta)$ is called the MCMLE of θ. It is known that the performance of the method depends on the choice of θ^*. If θ^* is too far from the true MLE, the method typically does not produce a good estimate of θ. Liang (2007) proposed an alternative Monte Carlo approach to approximate $\kappa(\theta)$, where $\kappa(\theta)$ is viewed as a marginal density function of the unnormalized distribution $g(x,\theta)$ and estimated using an adaptive kernel smoothing approach with Monte Carlo samples.

Toward Bayesian analysis for the model, equation 1.1, a significant step was made by Møller, Pettitt, Reeves, and Berthelsen (2006), who propose augmenting the distribution $f(x|\theta)$ by an auxiliary variable such that the normalizing constant ratio $\kappa(\theta)/\kappa(\vartheta)$ can be canceled in simulations. This algorithm was improved by Murray, Ghahramani, and MacKay (2006), who, based on the idea of parallel tempering (Geyer, 1991), proposed the following algorithm:

Exchange Algorithm

- Propose a candidate point ϑ from a proposal distribution denoted by $Q(\theta, \vartheta)$.
- Generate an auxiliary variable $y \sim f(y|\vartheta)$ using a perfect sampler (Propp & Wilson, 1996).
- Accept ϑ with probability $\min\{1, r(\theta, y, \vartheta)\}$, where

$$r(\theta, y, \vartheta) = \frac{\pi(\vartheta) f(x|\vartheta) f(y|\theta) Q(\vartheta, \theta)}{\pi(\theta) f(x|\theta) f(y|\vartheta) Q(\theta, \vartheta)} = \frac{\pi(\vartheta) g(x, \vartheta) g(y, \theta) Q(\vartheta, \theta)}{\pi(\theta) g(x, \theta) g(y, \vartheta) Q(\theta, \vartheta)}.$$

Since a swapping operation between (θ, x) and (ϑ, y) is involved, the algorithm is called the exchange algorithm. Both the Møller and the exchange algorithm are called auxiliary variable MCMC algorithms in the literature. The exchange algorithm generally improves the performance of the Møller algorithm. As Murray et al. (2006) reported, the exchange algorithm tends to have a higher acceptance probability than the Møller algorithm. Although the Møller and exchange algorithms work well for some discrete models, such as the Ising and autologistic models, they cannot be applied to many other models for which perfect sampling is not available. In addition, even for the Ising and autologistic models, perfect sampling may be very expensive when the temperature is near or below the critical point. To tackle this difficulty, Liang (2010) proposed replacing the exact sample by MH sample in the exchange algorithm, but the ergodicity of the algorithm is hard to establish.

Another way for conducting Bayesian inference of θ is to approximate the normalizing constant function $\kappa(\theta)$ in an offline way and then substitute it into equation 1.1 as a known function for posterior simulations. For example, Green and Richardson (2002) estimated $\kappa(\theta)$ at a number of discrete points, and Liang (2007) estimated the function $\kappa(\theta)$ as a marginal of the unnormalized distribution $g(x, \theta)$. However, these methods usually work only for the case that the dimension of θ is low.

In this letter, we propose a new algorithm, the Monte Carlo Metropolis-Hastings (MCMH) algorithm, for sampling from distributions with intractable normalizing constants. The MCMH algorithm is a Monte Carlo version of the Metropolis-Hastings algorithm. At each iteration, it replaces the unknown normalizing constant ratio $\kappa(\theta)/\kappa(\vartheta)$ by a Monte Carlo

estimate. Under mild conditions, we show that the MCMH algorithm can still converge to the desired stationary distribution $\pi(\theta|x)$. Unlike the Møller and exchange algorithms, the MCMH algorithm avoids the requirement for perfect sampling and thus can be applied to many statistical models for which perfect sampling is unavailable or very expensive.

The remainder of this letter is organized as follows. In section 2, we describe the MCMH algorithm and study its convergence theory. In section 3, we test the MCMH algorithm on spatial autologistic models. In section 4, we test the MCMH algorithm on social network models. In section 5, we discuss the relation between MCMH and the group independence MH algorithm introduced by Beaumont (2003), and the potential applications of MCMH in marginal inference. We conclude in section 6.

2 The Monte Carlo Metropolis-Hastings Algorithm

2.1 The Algorithm. Consider the problem of sampling from the distribution, equation 1.2. Let θ_t denote the current draw of θ by the algorithm. Let $y_1^{(t)}, \ldots, y_m^{(t)}$ denote the auxiliary samples simulated from the distribution $f(y|\theta_t)$, which can be drawn by either an MCMC algorithm or an exact sampling algorithm. The MCMH algorithm works by iterating between the following steps:

Monte Carlo MH Algorithm-I

1. Draw ϑ from a proposal distribution $Q(\theta_t, \vartheta)$.
2. Calculate the Monte Carlo MH ratio:
   2a. Estimate the normalizing constant ratio $R(\theta_t, \vartheta) = \kappa(\vartheta)/\kappa(\theta_t)$ by
   $$\widehat{R}_m(\theta_t, \boldsymbol{y}_t, \vartheta) = \frac{1}{m}\sum_{i=1}^{m} \frac{g(y_i^{(t)}, \vartheta)}{g(y_i^{(t)}, \theta_t)},$$
   where $\boldsymbol{y}_t = (y_1^{(t)}, \ldots, y_m^{(t)})$ denotes the collection of auxiliary samples.
   2b. Calculate
   $$\tilde{r}_m(\theta_t, \boldsymbol{y}_t, \vartheta) = \frac{1}{\widehat{R}_m(\theta_t, \boldsymbol{y}_t, \vartheta)} \frac{g(x, \vartheta)\pi(\vartheta)}{g(x, \theta_t)\pi(\theta_t)} \frac{Q(\vartheta, \theta_t)}{Q(\theta_t, \vartheta)}.$$
3. Set $\theta_{t+1} = \vartheta$ with probability $\tilde{\alpha}(\theta_t, \boldsymbol{y}_t, \vartheta) = \min\{1, \tilde{r}_m(\theta_t, \boldsymbol{y}_t, \vartheta)\}$, and set $\theta_{t+1} = \theta_t$ with the remaining probability.
4. If the proposal is rejected in step 3, set $\boldsymbol{y}_{t+1} = \boldsymbol{y}_t$. Otherwise draw samples $\boldsymbol{y}_{t+1} = (y_1^{(t+1)}, \ldots, y_m^{(t+1)})$ from $f(y|\theta_{t+1})$ using either an MCMC algorithm or an exact sampling algorithm.

Since the unknown normalizing constant ratio is estimated using the Monte Carlo method, this algorithm is termed Monte Carlo MH. Clearly the samples $\{(\theta_t, \boldsymbol{y}_t)\}$ form a Markov chain whose transition kernel is given

by

$$\begin{aligned}\tilde{P}_m(\theta, \boldsymbol{y}; d\vartheta, d\boldsymbol{z}) &= \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta) Q(\theta, d\vartheta) f_\vartheta^m(d\boldsymbol{z}) + \delta_{\theta,\boldsymbol{y}}(d\vartheta, d\boldsymbol{z}) \\ &\quad \times \left[1 - \int_{\Theta\times\mathbb{Y}} \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta') Q(\theta, d\vartheta') f_{\vartheta'}^m(d\boldsymbol{z}')\right] \\ &= \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta) Q(\theta, d\vartheta) f_\vartheta^m(d\boldsymbol{z}) + \delta_{\theta,\boldsymbol{y}}(d\vartheta, d\boldsymbol{z}) \\ &\quad \times \left[1 - \int_{\Theta} \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta') Q(\theta, d\vartheta')\right], \end{aligned} \tag{2.1}$$

where $f_\theta^m(\boldsymbol{y}) = f(y_1, \ldots, y_m|\theta)$ denotes the joint density of $y_1, \ldots, y_m$, and $\mathbb{Y} = \mathcal{X}^m$ denotes the sample space of $\boldsymbol{y}$. It is interesting to note that if $m = 1$ and the sample is drawn using an exact sampling algorithm, MCMH-I is reduced to the exchange algorithm.

The goal of this letter is to study the convergence of $\{\theta_t\}$, a marginal chain of $\{(\theta_t, \boldsymbol{y}_t)\}$. In general, if $\{(X_t, Y_t)\}$ forms a Markov chain, then the marginal path $\{X_t\}$ forms an adaptive Markov chain for which each state depends on all of its past states; that is, X_t depends on $X_{t-1}, \ldots, X_1, X_0$ for all $t \geq 1$. For the MCMH-I algorithm, the transition kernel of the marginal chain $\{\theta_t\}$ is given by

$$\begin{aligned}\tilde{P}_m(\theta_t, d\vartheta) &= \int_{\mathbb{Y}}\int_{\mathbb{Y}} \tilde{P}_m(\theta_t, \boldsymbol{y}_t; d\vartheta, d\boldsymbol{z}) f_{\theta_t}^m(d\boldsymbol{y}_t) \\ &= \int_{\mathbb{Y}} \tilde{\alpha}(\theta_t, \boldsymbol{y}_t, \vartheta) Q(\theta_t, d\vartheta) f_{\theta_t}^m(d\boldsymbol{y}_t) + \delta_{\theta_t}(d\vartheta) \\ &\quad \times \left[1 - \int_{\Theta\times\mathbb{Y}} \tilde{\alpha}(\theta_t, \boldsymbol{y}_t, \vartheta') Q(\theta_t, d\vartheta') f_{\theta_t}^m(d\boldsymbol{y}_t)\right]. \end{aligned} \tag{2.2}$$

It is easy to see that $\tilde{P}_m(\theta_t, d\vartheta)$ is independent of $\{\theta_{t-1}, \ldots, \theta_0\}$. This implies that the ergodicity of $\{\theta_t\}$ can be analyzed as a Markov chain. Note that the independence of $\tilde{P}_m(\theta_t, d\vartheta)$ on past states is not generally true for marginal Markov chains. It is true for MCMH-I as for which $\boldsymbol{y}_t$ is generated from $f_{\theta_t}(\boldsymbol{y})$; that is, $\boldsymbol{y}_t$ is independent of $\theta_0, \ldots, \theta_{t-1}$ conditioned on θ_t.

The central issue of MCMH-I is to use the auxiliary samples $y_1^{(t+1)}$, $\ldots, y_m^{(t+1)}$ generated from a short run of MCMC to estimate the normalizing constant ratio $R(\theta_t, \vartheta)$. For convenience, we call the Markov chain used for generating auxiliary samples an auxiliary Markov chain. In practice, the auxiliary sample size m is not necessarily very large. For example, a value between 20 and 50 has been very good for the examples studied in this letter. The auxiliary samples can be generated by the auxiliary Markov chain in m consecutive or thinned iterations. Considering the general dependence of MCMC samples, multiple auxiliary Markov chains are generally

preferred. Since the multiple chains can be run on a parallel architecture, the computational time can then be significantly reduced.

To shorten the burn-in period of auxiliary Markov chains, we propose an importance resampling-based initialization procedure for creating their starting values. The initialization procedure can be described as follows:

1. Resample $z_0^{(t+1)}$ from $\boldsymbol{y}_t = (y_1^{(t)}, \ldots, y_m^{(t)})$ by setting $z_0^{(t+1)} = y_i^{(t)}$ with a probability proportional to the importance weight given by

$$w_i = g(y_i^{(t)}, \theta_{t+1})/g(y_i^{(t)}, \theta_t), \quad i = 1, 2, \ldots, m. \tag{2.3}$$

2. Run an MCMC procedure for m_0 iterations and set the initial point $y_0^{(t+1)} = z_{m_0}^{(t+1)}$, where the MCMC procedure starts with $z_0^{(t+1)}$ and admits $f(z|\theta_{t+1})$ as the invariant distribution.

The value of m_0 should be chosen large enough such that $z_0^{(t+1)}$ and $z_{m_0}^{(t+1)}$ are independent. In practice, m_0 can be determined through some pilot runs of the auxiliary Markov chain at different values of θ. To save computational time, we may set $m_0 = 0$, setting $y_0^{(t+1)} = z_0^{(t+1)}$ by omitting step (2) of the initialization procedure. This may introduce to $\boldsymbol{y}_{t+1}$ a slight dependence on $\boldsymbol{y}_t$, but the dependence vanishes when m is large.

2.2 Some Variants of the MCMH Algorithm. The MCMH algorithm can have many variants. A simple one is to draw auxiliary samples at each iteration, regardless of acceptance or rejection of the last proposal. This variant can be described as follows:

Monte Carlo MH Algorithm-II

1. Draw ϑ from some proposal distribution $Q(\theta_t, \vartheta)$.
2. Calculate the Monte Carlo MH ratio:
   2a. Draw auxiliary samples $\boldsymbol{y}_t = (y_1^{(t)}, \ldots, y_m^{(t)})$ from $f(y|\theta_t)$ using an MCMC algorithm or an exact sampling algorithm.
   2b. Estimate the normalizing constant ratio $R(\theta_t, \vartheta) = \kappa(\vartheta)/\kappa(\theta_t)$ by
$$\widehat{R}_m(\theta_t, \boldsymbol{y}_t, \vartheta) = \frac{1}{m} \sum_{i=1}^{m} \frac{g(y_i^{(t)}, \vartheta)}{g(y_i^{(t)}, \theta_t)}.$$
   2c. Calculate
$$\tilde{r}_m(\theta_t, \boldsymbol{y}_t, \vartheta) = \frac{1}{\widehat{R}_m(\theta_t, \boldsymbol{y}_t, \vartheta)} \frac{g(x, \vartheta)\pi(\vartheta)}{g(x, \theta_t)\pi(\theta_t)} \frac{Q(\vartheta, \theta_t)}{Q(\theta_t, \vartheta)}.$$
3. Set $\theta_{t+1} = \vartheta$ with probability $\tilde{\alpha}(\theta_t, \boldsymbol{y}_t, \vartheta) = \min\{1, \tilde{r}_m(\theta_t, \boldsymbol{y}_t, \vartheta)\}$, and set $\theta_{t+1} = \theta_t$ with the remaining probability.

MCMH-II has a different Markovian structure from MCMH-I. In MCMH-II, $\{\theta_t\}$ forms a Markov chain with the transition kernel given by

$$\tilde{P}_m(\theta, d\vartheta) = \int_{\mathbb{Y}} \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta) Q(\theta, d\vartheta) f_\theta^m(d\boldsymbol{y}) + \delta_\theta(d\vartheta) \times \left[1 - \int_{\Theta\times\mathbb{Y}} \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta') Q(\theta, d\vartheta') f_\theta^m(d\boldsymbol{y})\right], \tag{2.4}$$

which is identical to the marginal transition kernel, equation 2.4, except for notations. Hence, the two algorithms will have the same convergence rate for $\{\theta_t\}$. Intuitively, one may expect that MCMH-I converges more slowly than MCMH-II, as MCMH-I recycles the auxiliary samples when rejection occurs and thus the successive samples generated by it may have significantly higher correlation than those generated by MCMH-II. In fact, the random error of $\widehat{R}_m(\theta_t, \boldsymbol{y}_t, \vartheta)$ depends mainly on θ_t and ϑ instead of $\boldsymbol{y}_t$ when m is large. This may help us to understand why MCMH-I and MCMH-II show the same convergence rate in numerical examples.

Similar to MCMH-II, we can propose another variant of MCMH, which, in step 2, draws auxiliary samples from $f(y|\vartheta)$ instead of $f(y|\theta_t)$. Then

$$\widehat{R}_m^*(\theta_t, \boldsymbol{y}_t, \vartheta) = \frac{1}{m}\sum_{i=1}^{m} \frac{g\left(y_i^{(t)}, \theta_t\right)}{g\left(y_i^{(t)}, \vartheta\right)},$$

forms an unbiased estimator of the ratio $\kappa(\theta_t)/\kappa(\vartheta)$, and the Monte Carlo MH ratio can be calculated as

$$\tilde{r}_m^*(\theta_t, \boldsymbol{y}_t, \vartheta) = \widehat{R}_m^*(\theta_t, \boldsymbol{y}_t, \vartheta) \frac{g(x, \vartheta)\pi(\vartheta)}{g(x, \theta_t)\pi(\theta_t)} \frac{Q(\vartheta, \theta_t)}{Q(\theta_t, \vartheta)}.$$

This algorithm is called MCMH-III in this letter. It has a similar Markovian structure to MCMH-II; that is, $\{\theta_t\}$ forms a Markov chain with the transition kernel given by

$$\tilde{P}'_m(\theta, d\vartheta) = \int_{\mathbb{Y}} \tilde{\alpha}^*(\theta, \boldsymbol{y}, \vartheta) Q(\theta, d\vartheta) f_\vartheta^m(d\boldsymbol{y}) + \delta_\theta(d\vartheta) \times \left[1 - \int_{\Theta\times\mathbb{Y}} \tilde{\alpha}^*(\theta, \boldsymbol{y}, \vartheta') Q(\theta, d\vartheta') f_{\vartheta'}^m(d\boldsymbol{y})\right], \tag{2.5}$$

where $\tilde{\alpha}_m^*(\theta, \boldsymbol{y}, \vartheta) = \min\{1, \tilde{r}_m^*(\theta_t, \boldsymbol{y}, \vartheta)\}$. We may expect that when m is small, MCMH-III performs a little better than MCMH-II as $\widehat{R}_m^*(\theta_t, \boldsymbol{y}_t, \vartheta)$ forms an unbiased estimator of $\kappa(\theta_t)/\kappa(\vartheta)$ while $1/\widehat{R}(\theta_t, \boldsymbol{y}_t, \vartheta)$ does not; when m is large, these two algorithms perform similarly. This is consistent with our numerical results as shown in Table 1. In appendix B, we calculate

Table 1: Computational Results for the U.S. Cancer Mortality Data.

Algorithm	Setting	$\widehat{\alpha}$	$\widehat{\beta}$	CPU	RE(%)
MCMH I	$m = 20$	$-0.3020\ (3.54 \times 10^{-4})$	$0.1230\ (1.73 \times 10^{-4})$	11	100
	$m = 50$	$-0.3015\ (2.93 \times 10^{-4})$	$0.1231\ (1.44 \times 10^{-4})$	24	66.2
	$m = 100$	$-0.3022\ (2.61 \times 10^{-4})$	$0.1228\ (1.29 \times 10^{-4})$	46	43.0
MCMH II	$m = 20$	$-0.3016\ (3.64 \times 10^{-4})$	$0.1233\ (1.82 \times 10^{-4})$	26	38.2
	$m = 50$	$-0.3016\ (3.05 \times 10^{-4})$	$0.1230\ (1.56 \times 10^{-4})$	63	21.5
	$m = 100$	$-0.3018\ (2.44 \times 10^{-4})$	$0.1229\ (1.21 \times 10^{-4})$	129	17.4
MCMH III	$m = 20$	$-0.3020\ (2.81 \times 10^{-4})$	$0.1229\ (1.42 \times 10^{-4})$	26	62.8
	$m = 50$	$-0.3013\ (2.37 \times 10^{-4})$	$0.1231\ (1.31 \times 10^{-4})$	63	30.5
	$m = 100$	$-0.3015\ (2.64 \times 10^{-4})$	$0.1231\ (1.25 \times 10^{-4})$	129	16.3
DMH	$m = 20$	$-0.3018\ (3.47 \times 10^{-4})$	$0.1228\ (1.80 \times 10^{-4})$	18	56.5
	$m = 50$	$-0.3015\ (3.18 \times 10^{-4})$	$0.1230\ (1.65 \times 10^{-4})$	43	28.1
	$m = 100$	$-0.3019\ (3.54 \times 10^{-4})$	$0.1225\ (1.81 \times 10^{-4})$	86	11.7
Exchange	—	$-0.3013\ (3.08 \times 10^{-4})$	$0.1230\ (1.60 \times 10^{-4})$	33	39.0

Notes: The numbers in the parentheses denote the standard (Monte Carlo) error of the estimates, which are evaluated based on 100 repeated runs. CPU: CPU time in seconds cost by a single run on a 3.0 GHz personal computer. RE (relative efficiency): Calculated in $(\sigma_1/\sigma_2)^2 * T_1/T_2 \times 100\%$, where σ_i and T_i $(i = 1, 2)$ denote the standard Monte Carlo error of $\widehat{\beta}$ produced and the CPU time cost by method i, and MCMH-I (with $m = 20$) is used as the standard (method 1) in the calculation.

the asymptotic variances of $\widehat{R}^*_m(\theta_t, \boldsymbol{y}_t, \vartheta)$ and $1/\widehat{R}(\theta_t, \boldsymbol{y}_t, \vartheta)$. Our results show that there is no a fixed ordering for their asymptotic variances, depending on the values of θ_t and ϑ.

In addition to $f(y|\theta_t)$ and $f(y|\vartheta)$, the auxiliary samples can be generated from a third distribution, which has the same support set as $f(y|\theta_t)$ and $f(y|\vartheta)$. In this case, the ratio importance sampling method (Torrie & Valleau, 1997; Chen & Shao, 1997) can be used for estimating the normalizing constant ratio, $\kappa(\theta_t)/\kappa(\vartheta)$. The existing normalizing constant ratio estimation techniques, such as bridge sampling (Meng & Wong, 1996) and path sampling (Gelman & Meng, 1998), are also applicable to MCMH with an appropriate strategy for generating auxiliary samples.

2.3 Convergence. In this section, we first prove the ergodicity of MCMH-II, showing

$$\tilde{P}^k_m(\theta_0, \cdot) - \pi(\cdot|x)\| \to 0, \quad \text{as } m \to \infty \quad \text{and} \quad k \to \infty,$$

where k denotes the number of iterations, $\pi(\cdot|x)$ denotes the target distribution defined in equation 1.2, and $\|\cdot\|$ denotes the total variation norm as specified in Tierney (1994). Then, we extend the results to MCMH-I and MCMH-III. The main results are presented below (the proofs are in

appendix A). Define

$$\gamma_m(\theta, \boldsymbol{y}, \vartheta) = \frac{R(\theta, \vartheta)}{\widehat{R}(\theta, \boldsymbol{y}, \vartheta)}. \tag{2.6}$$

In the context where confusion is impossible, we denote $\gamma_m = \gamma_m(\theta, \boldsymbol{y}, \vartheta)$. Define $\lambda_m = |\log(\gamma_m(\theta, \boldsymbol{y}, \vartheta))|$, and define

$$\rho(\theta) = 1 - \int_{\Theta \times \mathbb{Y}} \tilde{\alpha}_m(\theta, \boldsymbol{y}, \vartheta) Q(\theta, d\vartheta) f_\theta^m(d\boldsymbol{y}), \tag{2.7}$$

which represents the mean rejection probability of an MCMH-II transition from θ.

To show the convergence of MCMH-II, we also consider the transition kernel,

$$P(\theta, \vartheta) = \alpha(\theta, \vartheta) Q(\theta, \vartheta) + \delta_\theta(d\vartheta) \left[1 - \int_\Theta \alpha(\theta, \vartheta') Q(\theta, \vartheta') d\vartheta'\right], \tag{2.8}$$

which is induced by the proposal $Q(\cdot, \cdot)$. In addition, we assume the following conditions:

A_1: Assume that P defines an irreducible and aperiodic Markov chain such that $\pi(\cdot|x)P = \pi(\cdot|x)$. Therefore, for any $\theta_0 \in \Theta$, $\lim_{k\to\infty} \|P^k(\theta_0, \cdot) - \pi(\cdot|x)\| = 0$.

A_2: For any $(\theta, \vartheta) \in \Theta \times \Theta$,

$$0 < \gamma_m(\theta, \boldsymbol{y}, \vartheta) < \infty, \quad f_\theta^m(\cdot) - a.s.$$

A_3: For any $\theta \in \Theta$ and any $\epsilon > 0$,

$$\lim_{m\to\infty} Q\left(\theta, f_\theta^m(\lambda_m(\theta, \boldsymbol{y}, \vartheta) > \epsilon)\right) = 0,$$

where $Q\left(\theta, f_\theta^m(\lambda_m(\theta, \boldsymbol{y}, \vartheta) > \epsilon)\right) = \int_{\{(\vartheta, \boldsymbol{y}): \lambda_m(\theta, \boldsymbol{y}, \vartheta) > \epsilon\}} f_\theta^m(d\boldsymbol{y}) Q(\theta, d\vartheta)$.

Condition A_1 can be simply satisfied by choosing an appropriate proposal distribution $Q(\cdot, \cdot)$, following from the standard theory of the Metropolis-Hastings algorithm (Tierney, 1994). Condition A_2 is equivalent to assuming $0 < \widehat{R}(\theta, \boldsymbol{y}, \vartheta) < \infty$, which ensures the MCMH ratio to be well defined in simulations. Condition A_3 is equivalent to assuming that for any $\theta \in \Theta$ and any $\epsilon > 0$, there exists a positive integer M such that for any $m > M$,

$$Q\left(\theta, f_\theta^m(\lambda_m(\theta, \boldsymbol{y}, \vartheta) > \epsilon)\right) \leq \epsilon.$$

That is, it requires that $\widehat{R}(\theta, \boldsymbol{y}, \vartheta)$ is a consistent estimator of $R(\theta, \vartheta)$ and the step size of the proposal $Q(\theta, \vartheta)$ is reasonably small (i.e., ϑ lies in a small neighborhood of θ). Note that conditions A_2 and A_3 have implicitly incorporated into our consideration the approximation caused by selection of the initial auxiliary sample at each iteration. It follows from the standard theory of MCMC that when m is large, conditions A_2 and A_3 can still hold even if $y_0^{(t)}$ is not an exact sample from the distribution $f(y|\theta_t)$.

Lemma 1 states that the marginal kernel $\tilde{P}_m$ has a stationary distribution. It is proved in a similar way to theorem 1 of Andrieu and Roberts (2009). (The relation between this work and Beaumont, 2003, and Andrieu & Roberts, 2009, is discussed in section 5.)

Lemma 1. *Assume conditions A_1 and A_2 hold. Then for any $m \in \mathbb{N}$ such that for any $\theta \in \Theta$, $\rho(\theta) > 0$, $\tilde{P}_m$ is also irreducible and aperiodic, and hence there exists a stationary distribution $\tilde{\pi}_m(\theta|x)$ such that for any $\theta_0 \in \Theta$,*

$$\lim_{k\to\infty} \| \tilde{P}_m^k(\theta_0, \cdot) - \tilde{\pi}_m(\cdot|x)\| = 0.$$

Lemma 2 concerns the distance between the kernel $\tilde{P}_m$ and the kernel P. It states that the two kernels can be arbitrarily close to each other, provided that m is large enough:

Lemma 2. *Assume condition A_3 holds. Let $\epsilon \in (0, 1]$. Then for any $\theta \in \Theta$, there exists $M(\theta) \in \mathbb{N}$ such that for any $\psi : \Theta \to [-1, 1]$ and any $m > M(\theta)$,*

$$|\tilde{P}_m\psi(\theta) - P\psi(\theta)| \le 4\epsilon.$$

Theorem 1 concerns the ergodicity of MCMH-II. It states that the kernel $\tilde{P}_m$ asymptotically shares the same stationary distribution with the MH kernel P:

Theorem 1. *Assume conditions A_1, A_2, and A_3 hold for MCMH-II. Then for any $\varepsilon \in (0, 1]$ and any $\theta_0 \in \Theta$, there exist $M(\varepsilon, \theta_0) \in \mathbb{N}$ and $K(\varepsilon, \theta_0, m) \in \mathbb{N}$ such that for any $m > M(\varepsilon, \theta_0)$ and $k > K(\varepsilon, \theta_0, m)$*

$$\| \tilde{P}_m^k(\theta_0, \cdot) - \pi(\cdot|x)\| \le \varepsilon,$$

where $\pi(\cdot|x)$ denotes the posterior density of θ.

Theorem 2. *Assume conditions A_1, A_2, and A_3 hold for MCMH-I. Then the marginal chain $\{\theta_t\}$ induced by MCMH-I has the same stationary distribution as the Markov chain $\{\theta_t\}$ induced by MCMH-II.*

To study the ergodicity of MCMH-III, we define

$$\gamma'_m(\theta, \boldsymbol{y}, \vartheta) = \frac{\widehat{R}^*(\theta, \boldsymbol{y}, \vartheta)}{R(\vartheta, \theta)}, \quad \lambda'_m(\theta, \boldsymbol{y}, \vartheta) = |\log(\gamma'_m(\theta, \boldsymbol{y}, \vartheta))|$$

and

$$\rho'(\theta) = 1 - \int_{\Theta \times \mathbb{Y}} \tilde{\alpha}^*_m(\theta, \boldsymbol{y}, \vartheta) Q(\theta, d\vartheta) f^m_\vartheta(d\boldsymbol{y}),$$

where $\tilde{\alpha}^*_m(\theta, \boldsymbol{y}, \vartheta) = \min\{1, \tilde{r}^*_m(\theta_t, \boldsymbol{y}, \vartheta)\}$. If we assume that condition A_1 and the following two conditions hold:

A'_2: For any $(\theta, \vartheta) \in \Theta \times \Theta$,

$$0 < \gamma'_m(\theta, \boldsymbol{y}, \vartheta) < \infty, \quad f^m_\theta(\cdot) - a.s.,$$

A'_3: For any $\theta \in \Theta$ and any $\epsilon > 0$,

$$\lim_{m \to \infty} Q\left(\theta, f^m_\vartheta(\lambda'_m(\theta, \boldsymbol{y}, \vartheta) > \epsilon)\right) = 0,$$

where $Q\left(\theta, f^m_\vartheta(\lambda'_m(\theta, \boldsymbol{y}, \vartheta) > \epsilon)\right) = \int_{\{(\vartheta, \boldsymbol{y}): \lambda'_m(\theta, \boldsymbol{y}, \vartheta) > \epsilon\}} f^m_\vartheta(d\boldsymbol{y}) Q(\theta, d\vartheta)$,

then the ergodicity of MCMH-III can be established in a similar way to that of MCMH-II. In summary, we have the following theorem:

Theorem 3. *Assume conditions A_1, A'_2, and A'_3 hold for MCMH-III. Then the Markov chain $\{\theta_t\}$ induced by MCMH-III has the same stationary distribution as that induced by MCMH-II.*

Although theorems 1 through 3 are established under the condition that $m_0 > 0$, that is, $y_0^{(t+1)}$ is independent of $\boldsymbol{y}_t$, they may still hold for the case $m_0 = 0$. This is because these theorems are all established for large values of m, while the dependence of $z_0^{(t+1)}$ on $\boldsymbol{y}_t$ vanishes when m is large.

Theorems 1 through 3 imply, by standard MCMC theory (see, e.g., Tierney, 1994), that for an integrable function $h(\theta)$, the path-averaging estimator $\sum_{k=1}^n h(\theta_k)/n$ will converge to its posterior mean almost surely; that is, as $k \to \infty$,

$$\frac{1}{n} \sum_{k=1}^{n} h(\theta_k) \to \int h(\theta) \pi(\theta | x) d\theta, \quad a.s.,$$

provided that $\int |h(\theta)|\pi(\theta|x)d\theta < \infty$ and m has been sufficiently large so that the error in replacing $\tilde{\pi}_m(\theta|x)$ by $\pi(\theta|x)$ is ignorable. Here $\tilde{\pi}_m$ denotes the stationary distribution established in lemma 1 for a fixed value of m.

3 Bayesian Analysis for Spatial Autologistic Models

The autologistic model (Besag, 1974) has been widely used for spatial data analysis (see Preisler, 1993; Wu & Huffer, 1997; Sherman, Apanasovich, & Carroll, 2006). Let $\boldsymbol{x} = \{x_i : i \in D\}$ denote the observed binary data, where $x_i \in \{-1, 1\}$ is called a spin and D is the set of indices of the spins. Let $|D|$ denote the total number of spins in D, and let $n(i)$ denote the set of neighbors of spin i. The likelihood function of the model is

$$f(\boldsymbol{x}|\alpha, \beta) = \frac{1}{Z(\alpha, \beta)} \exp\left\{\alpha \sum_{i\in D} x_i + \frac{\beta}{2} \sum_{i\in D} x_i \left(\sum_{j\in n(i)} x_j\right)\right\}, \quad (\alpha, \beta) \in \Theta, \tag{3.1}$$

where the parameter α determines the overall proportion of $x_i = +1$, the parameter β determines the intensity of interaction between x_i and its neighbors, and $Z(\alpha, \beta)$ is the intractable normalizing constant defined by

$$Z(\alpha, \beta) = \sum_{\text{for all possible } \boldsymbol{x}} \exp\left\{\alpha \sum_{j\in D} x_j + \frac{\beta}{2} \sum_{i\in D} x_i \Big(\sum_{j\in n(i)} x_j\Big)\right\}.$$

An exact evaluation of $Z(\alpha, \beta)$ is prohibited even for a moderate system.

To conduct a Bayesian analysis for the model, a uniform prior on

$$(\alpha, \beta) \in \Theta = [-1, 1] \times [0, 1]$$

is assumed for the parameters. Then MCMH can be applied to simulate from the posterior distribution $\pi(\alpha, \beta|\boldsymbol{x})$. The proposal distribution $Q(\cdot, \cdot)$ we used here is a gaussian random walk proposal $N_2((\alpha_t, \beta_t)^T, s^2 I_2)$, where s is the step size and I_2 is the 2×2 identity matrix. Each auxiliary sample is generated by a single cycle of Gibbs updates. The acceptance rate of the MCMH moves can be controlled by the value of s. In this section, we set $s = 0.03$ for all examples, although it may not be optimal.

3.1 U.S. Cancer Mortality Data. U.S. cancer mortality maps have been compiled by Riggan et al. (1987) for investigating the possible association of cancer with unusual demographic, environmental, industrial characteristics, or employment patterns. Figure 1a shows the mortality map of liver and gall bladder (including bile ducts) cancers for white males during the

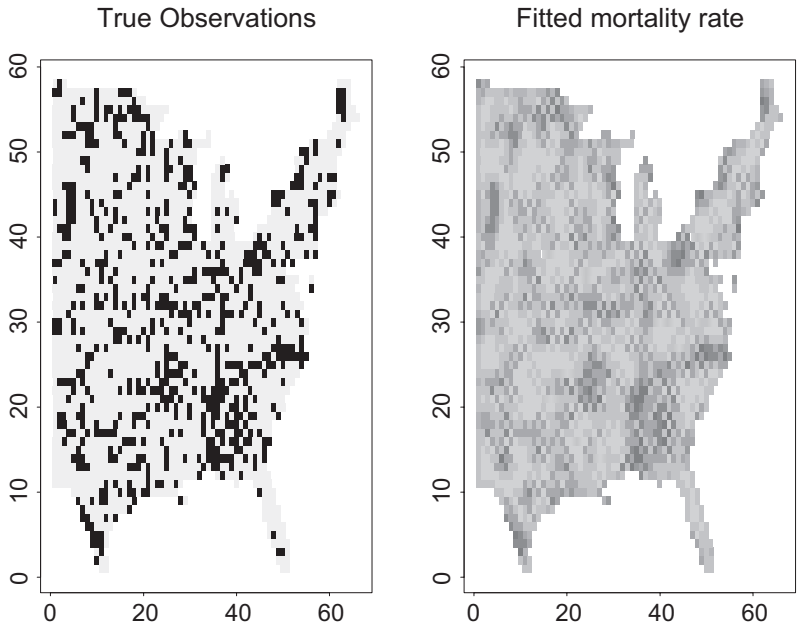

Figure 1: U.S. cancer mortality data. (Left) Mortality map of liver and gall bladder cancers (including bile ducts) for white males, 1950–1959. Black squares denote counties of high cancer mortality rate, and white squares denote counties of low cancer mortality rate. (Right) Estimated cancer mortality rates using the autologistic model with the model parameters being replaced by its approximate Bayesian estimates. The cancer mortality rate of each county is represented by the gray level of the corresponding square.

decade 1950 to 1959, which indicates some apparent geographic clustering. (See Sherman et al., 2006, for more descriptions of the data.) Following Sherman et al. (2006), we modeled the data by a spatial autologistic model. The total number of spins is $|D| = 2293$. A free boundary condition is assumed for the model, under which the boundary points have fewer neighboring points than the interior points. This assumption is natural to these data, as the boundary of the lattice has an irregular shape, as shown in Figure 1.

The MCMH algorithms were first applied to this example with $m_0 = 0$ and different choices of $m = 20$, 50, and 100. For each value of m, each algorithm was run 100 times independently. Each run started with a random point drawn uniformly on the region $[-1, 1] \times [0, 1]$ (independent of other runs) and consisted of 5000 iterations, with the first 1000 iterations being discarded for the burn-in process and 4000 samples of θ were collected from the remaining iterations. The overall acceptance rates of the MCMH-I moves are 0.41, 0.37, and 0.36 for the runs with $m = 20$, $m = 50$,

and $m = 100$, respectively. For MCMH-II, they are 0.42, 0.38, and 0.36, respectively, and for MCMH-III, they are 0.28, 0.29, and 0.29, respectively. This implies that our implementations for all the three MCMH algorithms are efficient. A diagnosis based on the Gelman-Rubin statistic (Gelman & Rubin, 1992) showed that for each value of m, the simulations converged very fast, usually within a few hundred iterations. The details of the diagnosis are omitted here. The numerical results were summarized in Table 1. MCMH-I and MCMH-II produced very similar results for this example, while MCMH-I cost less than 50% CPU times than MCMH-II. As expected, MCMH-III costs the same CPU time as MCMH-II but produced more accurate estimates than MCMH-II when m is small. A common feature of the MCMH algorithms is that they can produce more accurate estimates with a larger value of m, although at the price of longer CPU times. It is worth noting that the MCMH estimator seems unbiased even with a value of m as small as 20.

To assess the validity of the MCMH algorithms, the exchange algorithm was applied to this example. This algorithm is an auxiliary variable MCMC algorithm, which requires a perfect sampler for generating auxiliary variables but can sample correctly from the posterior distribution when the number of iterations becomes large. Hence, the estimates produced by the exchange algorithm can be used as a test standard for assessing whether the results produced by MCMH are correct. The perfect sampler used here is the summary state algorithm (Childs, Patterson, & MacKay, 2001), which is known to be suitable for high-dimensional binary spaces. The exchange algorithm was also run 100 times independently, and each run consisted of 5000 iterations. The first 1000 iterations were discarded for the burn-in process, and the remaining 4000 iterations were used for estimating θ. The overall acceptance rate was 0.2, which indicates that the algorithm has been implemented efficiently. The numerical results were summarized in Table 1. The comparison indicates that the MCMH algorithms are valid. To calibrate the efficiency of these algorithms, we calculate their relative efficiency based on their CPU cost and their estimates of β produced in 100 runs. The results were reported in Table 1, which indicate that among the four algorithms, MCMH-I is the most efficient for this example. The exchange algorithm is only about 40% efficient as MCMH-I, and MCMH-III is only about 60% efficient as MCMH-I.

Furthermore, we compare MCMH with the double Metropolis-Hastings (DMH) algorithm (Liang, 2010). DMH can be viewed as an approximate exchange algorithm, which replaces at each iteration the exact sample by a sample simulated from a short run of the MH algorithm. To simulate an auxiliary sample from $f(y|\vartheta)$, DMH initializes the MH algorithm at the observation x (i.e., setting $y_0^{(t)} = x$) and then iterates for m steps. Table 1 reported the results of DMH produced with $m = 20, 50$, and 100. It is easy to see that DMH is inferior to the MCMH algorithms in efficiency. Compared to MCMH, a significant disadvantage of DMH is that its performance is

Table 2: Variance Comparison for MCMH Stationary Distributions.

Algorithm	Setting	σ_α^2	σ_β^2	$\sigma_{\alpha\beta}$
MCMH-I	$m = 20$	1.80×10^{-3}	6.21×10^{-4}	6.99×10^{-4}
	$m = 50$	1.46×10^{-3}	5.14×10^{-4}	6.36×10^{-4}
	$m = 100$	1.40×10^{-3}	4.07×10^{-4}	5.22×10^{-4}
Exchange	—	9.21×10^{-4}	2.90×10^{-4}	4.05×10^{-4}

Note: σ_α^2: Variance corresponding to the component α. σ_β^2: Variance corresponding to the component β; and $\sigma_{\alpha\beta}$: the covariance corresponding to the components α and β.

bounded by the exchange algorithm; DMH cannot produce more accurate parameter estimates than the exchange algorithm even with a large value of m. However, MCMH can do so. As shown in Table 1, MCMH-I and MCMH-II can produce more accurate estimates than the exchange algorithm with $m = 100$, and MCMH-III can do so even with $m = 20$.

The estimates produced by the MCMH algorithms, the exchange algorithm, and DMH are also very close to those reported in the literature. Liang (2007) analyzed these data using contour Monte Carlo and produced the estimate $(-0.3008, 0.1231)$. Contour Monte Carlo first approximates the normalizing constant function on a given region and then estimates the parameters based on the approximated normalizing constant function. As Liang (2007) reported, the algorithm took hours of CPU time to approximate the normalizing constant function. Sherman et al. (2006) analyzed the data using the Monte Carlo maximum likelihood algorithm (Geyer & Thompson, 1992) and produced the estimate $(-0.304, 0.117)$, which is a bit far from the estimate of the exchange algorithm. This may reflect the difference between the posterior mode and the posterior mean. Since the uniform prior is used for θ, the posterior mode coincides with the MLE for this example.

To assess the effect of m on the stationary distribution $\tilde{\pi}_m(\theta|x)$, we compare the variances of $\tilde{\pi}_m(\theta|x)$ for $m = 20$, 50, and 100 and the variance of $\pi(\theta|x)$. The latter was estimated using the exchange algorithm. Each of the MCMH-I and exchange algorithms was run 50 times for each value of m. Each run consisted of 5000 iterations, with samples collected at every 200th iteration after the first 1000 burn-in iterations. A total of 1000 samples were collected from 50 runs for value of m. The autocorrelation plots (omitted here) indicate that the samples collected in this way are approximately independent. The resulting variance estimates are given in Table 2. The numerical results imply that as m increases, $\tilde{\pi}_m(\theta|x)$ gets closer and closer to $\pi(\theta|x)$.

Finally, we assess the effect of m_0. Figure 2 shows the autocorrelation plots of the sufficient statistics $(\sum_{i\in D} x_i, \sum_{i\in D} x_i(\sum_{j\in n(i)} x_j)$ for the samples generated by the Gibbs sampler at $(\alpha, \beta) = (-0.3, 0.123)$ during 50,000

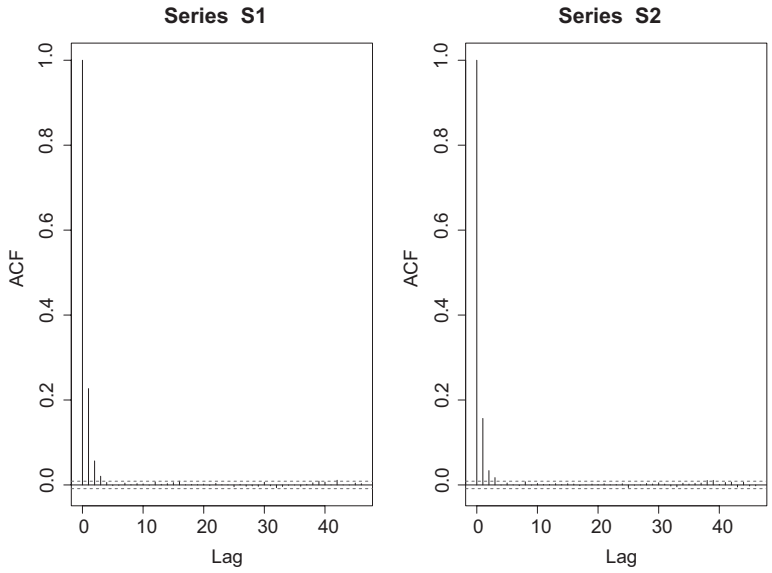

Figure 2: Autocorrelation plots of the samples generated by the Gibbs sampler for the autologistic model at $\alpha = -0.3$ and $\beta = 0.123$. (Left) Autocorrelation for the statistic $S_1 = \sum_{i \in D} x_i$. (Right) Autocorrelation for the statistic $S_2 = \sum_{i \in D} x_i \sum_{j \in n(i)} x_j$.

Table 3: Assessment of the Effect of m_0 for the MCMH-I Algorithm.

Setting	$\widehat{\alpha}$	$\widehat{\beta}$	CPU	RE(%)
$m = 20$	$-0.3016\ (3.32 \times 10^{-4})$	$0.1232\ (1.62 \times 10^{-4})$	17	73.8
$m = 50$	$-0.3019\ (3.15 \times 10^{-4})$	$0.1230\ (1.55 \times 10^{-4})$	29	47.3
$m = 100$	$-0.3015\ (2.33 \times 10^{-4})$	$0.1232\ (1.23 \times 10^{-4})$	50	43.5

Notes: The numbers in the parentheses denote the standard (Monte Carlo) error of the estimates, which are evaluated based on 100 repeated runs. CPU: CPU time in seconds cost by a single run on a 3.0 GHz personal computer. RE is calculated as in Table 1.

consecutive iterations. It indicates that the samples have a short correlation length, and an independent sample can be generated with about 5 iterations. To ensure the condition $y_0^{(t+1)}$ is independent of $\boldsymbol{y}_t$ and holds for all points near $(-0.3, 0.123)$, we set $m_0 = 20$ and then rerun MCMH-I with $m = 20$, 50, and 100. The results were summarized in Table 3. For comparison, the relative efficiency of the respective settings is also calculated as in Table 1 with the result of MCMH-I ($m_0 = 0$, $m = 20$) as the standard. When we Compare with the results of MCMH-I given in Table 1, it is easy to see

that the effect of m_0 decays as m increases; that is, when m is large, the two settings of m_0 will lead to about the same results.

The exchange algorithm works very well for this example. In section 3.2, we present a simulated example, where the exchange algorithm does not work well in some cases, while the MCMH algorithms still work well.

3.2 Simulation Studies. To assess the general accuracy of the estimates produced by MCMH, we simulated 50 independent samples for the U.S. cancer mortality data under each setting of (α, β) given in Table 4. All simulations were done using the summary state algorithm (Childs et al., 2001). Since the boundary of the lattice is irregular, the free boundary condition was again assumed in the simulations. We then reestimated the parameters using the MCMH-I and MCMH-III algorithms (with $m = 20$) and the exchange algorithm. All the three algorithms were run as for the previous example. The computational results are summarized in Table 4. Since MCMH-II always performs similar to MCMH-I but costs longer CPU times, its results are omitted in Table 4.

Table 4 indicates that the MCMH-I and MCMH-III algorithms can produce almost the same results as the exchange algorithm, but with much shorter CPU time for most cases. The variation of the CPU times of MCMH-I is due to the variation of its acceptance rate. MCMH-I has the lowest acceptance rate 0.35 at $\theta = (0, 0.4)$ and the highest acceptance rate 0.68 at $\theta = (0.5, 0.5)$. This variation can be easily smoothed by a fine tune of the step size s at different values of θ. For the exchange algorithm, the CPU time increases exponentially as β increases. Childs et al. (2001) studied the behavior of the exact sampler for the Ising model, a simplified autologistic model with α being constrained to 0. For the Ising model, they fitted an exponential law for the convergence time and reported that the exact sampler may diverge at a value of β lower than the critical value ($\approx$0.44). Childs et al.'s finding is consistent with the results reported in Table 4. Under the setting (0, 0.4), the exchange algorithm took an extremely long CPU time to produce an estimate of θ. Under the setting (0.5, 0.5), it failed to produce an estimate of θ even with more than 45 hours of CPU time on a 3.0 GHz personal computer. Finally, we note that due to the effect of α, it usually takes more CPU time for the exact sampler to generate a sampler under the setting $(0, \beta)$ than under the setting (α, β).

For a thorough exploration of the performance of the MCMH algorithms, we calculate the integrated autocorrelation time (IAT) of β samples generated by them. The results for the α samples are similar. IAT is often used to compare the efficiency of Monte Carlo algorithms. The shorter the IAT is, the larger number of independent samples the algorithm can generate per time unit and thus the more efficient the algorithm is. IAT can be estimated by

$$\Lambda = 1 + 2\sum_{k=1}^{L} c(k), \tag{3.2}$$

Table 4: Computational Results for the Simulated U.S. Cancer Mortality Data.

	MCMH-I				MCMH-III				Exchange Algorithm				MCMLE	
(α, β)	$\widehat{\alpha}$	$\widehat{\beta}$	IAT	CPU	$\widehat{\alpha}$	$\widehat{\beta}$	IAT	CPU	$\widehat{\alpha}$	$\widehat{\beta}$	IAT	CPU	$\widehat{\alpha}$	$\widehat{\beta}$
(0,0.1)	−.0037 (.0024)	.1006 (.0018)	6.6	11	−.0039 (.0024)	.1002 (.0018)	10.1	26	−.0037 (.0024)	.1004 (.0018)	16.4	33	−.0037 (.0024)	.1010 (.0018)
(0,0.2)	−.0024 (.0020)	.2007 (.0019)	6.5	10.5	−.0023 (.0020)	.2002 (.0018)	11.7	26	−.0025 (.0020)	.2009 (.0019)	23.4	79	−.0024 (.0020)	.2025 (.0019)
(0,0.3)	−.0009 (.0014)	.2971 (.0017)	7.4	10	−.0008 (.0014)	0.2977 (.0018)	12.2	26	−.0008 (.0014)	.2973 (.0018)	29.0	228	−.0010 (.0013)	.2986 (.0017)
(0,0.4)	.0003 (.0006)	.3985 (.0013)	9.2	9.2	.0002 (.0008)	.4056 (.0013)	15.7	26	.0002 (.0005)	.3982 (.0012)	79.0	5325	.0001[a] (.0005)	.3996[a] (.0012)
(0.1,0.1)	.1031 (.0025)	.0986 (.0022)	6.5	11	.1030 (.0026)	.0984 (.0022)	10.7	26	.1029 (.0026)	.0988 (.0022)	20.2	32	.1026 (.0025)	.0992 (.0022)
(0.3,0.3)	.3002 (.0093)	.3011 (.0042)	68.1	11	.3016 (.0097)	.3008 (.0043)	69.1	26	.3046 (.0092)	.2994 (.0042)	101.1	97	.2916 (.0094)	.3037 (.0042)
(0.5,0.5)	.5002 (.0223)	.5082 (.0091)	173.8	16.8	.5034 (.0245)	.5093 (.0095)	199.8	26	— —	— —	—	—	.5163 (.0328)	.4997 (.0117)

Notes: The numbers in parentheses denote the standard error of the estimates, which are evaluated based on 50 runs but with each for a different data set. IAT: integrated autocorrelation time; CPU: CPU time in seconds cost by a single run of the algorithm on a 3.0 GHz person computer.
[a]The values were calculated based on 43 data sets; MCMLE failed to converge for the other seven data sets.

where $c(k) = \text{Corr}(\beta_t, \beta_{t+k})$ is the autocorrelation coefficient of β_t and β_{t+k}, and L can be determined by the self-consistent windowing approach (Madras, 2000) as follows:

1. Determine the value of $L_1 = \min\{k : c(k) < 0\}$ and set $L = 2L_1$.
2. Calculate the current estimate Λ in equation 3.2.
3. If $L > 5\Lambda$, stop; otherwise, set $L = 5\Lambda$ and return to step 2.

To calculate Λ, for each setting of (α, β) in Table 4, MCMH-I was run once for one data set. The run has the same setting $m = 20$ as before except that it is much longer, consisting of 50,000 iterations. The resulting estimates of IAT were given in Table 4. We have tried different data sets, and the results were similar. For comparison, MCMH-III and the exchange algorithm were also run on the same data sets and each for 50,000 iterations; the results are reported in Table 4. Table 4 shows a common pattern for all the three algorithms. As expected, IAT increases as α or β increases. These results suggest that when α or β increases, to maintain the same level of Monte Carlo errors, MCMH may need a larger number of iterations or a larger value of m; for the exchange algorithm, one may have to increase the number of iterations.

For this example, the exchange algorithm has a longer autocorrelation time than the MCMH algorithms, and MCMH-III has a longer autocorrelation time than MCMH-I. This phenomenon is likely caused by the approximation error of the normalizing constant ratio. As illustrated by Figure 3 (the plots are similar for other settings of (α, β)), the samples generated by MCMH have larger variations and wider ranges than those generated by the exchange algorithm, and thus the autocorrelation time is shorter. The shorter autocorrelation time implies that MCMH can generate more independent samples within the same number of iterations, and this feature compensates for its inefficiency caused by large sample variations. As shown in Table 4, the overall results, the parameter estimates and standard errors calculated over 50 different data sets, suggest that MCMHs are very sound algorithms even when m is small. Note that the standard errors reported in Table 4 have mixed the Monte Carlo error due to simulations and the error due to different data sets. The highly consistent standard errors from different algorithms imply that the Monte Carlo errors from different algorithms are similar.

For a thorough comparison, the MCMLE method was also applied to this example. To resolve the difficulty in choosing the initial point θ^* in equation 1.3, a recursive procedure, originally suggested by Geyer and Thompson (1992), was used:

0. Initialize $\theta^{(0)}$ with the maximum pseudo-likelihood estimator, and set $k = 0$.
1. Simulate $m = 10{,}000$ samples from $f(x|\theta^{(k)})$ using the Gibbs sampler.
2. Find $\theta^{(k+1)} = \arg\max_\theta \log f_m(x|\theta^{(k)})$.

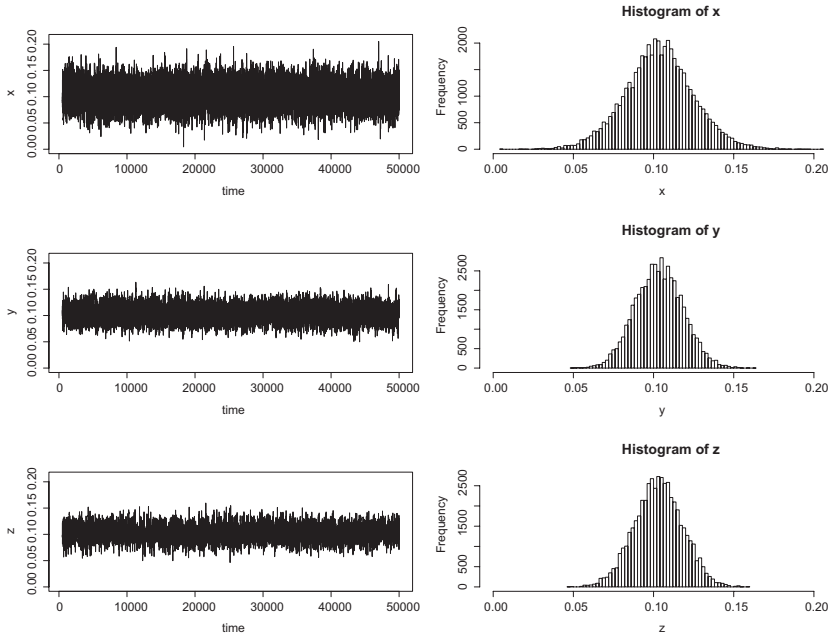

Figure 3: Plots of β samples generated by MCMH and the exchange algorithm for the case $(\alpha, \beta) = (0.1, 0.1)$. (Top) Time plot and histogram of β samples generated by MCMH-I. (Middle) Time plot and histogram of β samples generated by MCMH-III. (Bottom) Time plot and histogram of β samples generated by the exchange algorithm.

The algorithm iterates between steps 1 and 2 for 10 times such that a total of 100,000 auxiliary samples were generated in each run. This approximately matches with the CPU cost of the MCMH-III algorithm. The numerical results are shown in Table 4. MCMLE failed to converge for 7 data sets generated under the setting $\theta = (0, 0.4)$ due to difficulty in finding appropriate initial points. The overall performance of MCMLE on this example is good, but there is a clear pattern that its performance deteriorates as β increases. Comparing to MCMLE, the performance of the MCMH algorithms is less affected by the value of β. Geyer and Thompson (1992) also suggest increasing the value of m with the number of iterations. Since m has been set to a large value for this example, our implementation should be effective for assessing the performance of MCMLE.

4 Bayesian Analysis for Exponential Random Graph Models

Social network analysis has emerged as a key technique in modern sociology. The exponential family of random graphs is among the most widely

used, flexible models in social network analysis, which includes edge and dyadic independence models, Markov random graphs (Frank & Strauss, 1986), exponential random graphs (also known as p^* models; Snijders et al., 2006), and many other models. The model of particular interest is the exponential random graph model (ERGM), which allows one to include various network dependent structures in the analysis and thus improves goodness of fit for various social networks. (see Robins, Pattison, Kalish, & Lusher, 2007, for an overview of ERGMs.)

Consider a social network with n actors. The network can be specified in a matrix $\boldsymbol{X}$, where $X_{ij} = 1$ if there is a network tie from i to j and 0 otherwise. This matrix is known as the adjacency matrix. Note that the social network can be directed or nondirected. The likelihood function of the ERGM is given by

$$f(x|\theta) = \frac{1}{\kappa(\theta)} \exp\left\{\sum_{a \in A} \theta_a s_a(x)\right\}, \tag{4.1}$$

where $s_a(x)$ denotes an explanatory statistic, θ_a is the corresponding coefficient, A denotes the set of network statistics considered in the model, and $\kappa(\theta)$ is an intractable normalizing constant, which makes equation 4.1 a proper probability distribution.

In the literature, two methods are usually used for estimation of θ; the maximum pseudo-likelihood estimation (MPLE) method (Strauss & Ikeda, 1990) and the Monte Carlo maximum likelihood estimation (MCMLE) method (Snijders, 2002; Hunter & Handcock, 2006). The MPLE method analyzes ERGMs with a simplified, analytic form of the likelihood function under the assumption of dyadic independence. The properties of this method have been studied by many authors (Corander, Dahmström, & Dahmström, 1998; Wasserman & Robins, 2005; Lubbers & Snijders, 2007; van Duijn, Gile, & Handcock, 2009). MPLE is intrinsically highly dependent on the observed network. It usually works well for the networks with a low dependence structure but may produce substantially biased estimates for the networks with high dependency. The MCMLE method originates in Geyer and Thompson (1992), whose idea we briefly described in section 1. The main difficulty with this method is in finding an appropriate initial point. As Bartz, Blitzstein, and Liu (2008) pointed out, MCMLE often fails to converge in ERGMs, because the initial point (MPLE is usually used as the initial point) is often too far from the true MLE.

4.1 Exponential Random Graph Models. To define the ERGM explicitly, the explanatory statistics $s_a(x)$, $a \in A$, need to be specified. Since the number of possible specifications is large, only a few key statistics are considered here: the edge, degree distribution, and shared partnership

distribution. The edge, denoted by $e(x)$, counts the number of edges in the network. The other two statistics are defined below.

4.1.1 Degree Distribution. Let $D_i(x)$ denote the number of nodes in the network x whose degree, the number of edges incident to the node, equals i. For example, $D_{n-1}(x) = n$ when x is the complete graph and $D_0(x) = n$ when x is the empty graph. Note that $D_0(x), \ldots, D_{n-1}(x)$ satisfy the constraint $\sum_{i=0}^{n-1} D_i(x) = n$, and the number of edges in x can be expressed as

$$e(x) = \frac{1}{2} \sum_{i=1}^{n-1} iD_i(x).$$

The degree distribution statistic (Snijders et al., 2006; Hunter & Handcock, 2006; Hunter, 2007) is defined as

$$u(x|\tau) = e^{\tau} \sum_{i=1}^{n-2} \left\{1 - \left(1 - e^{-\tau}\right)^i\right\} D_i(x), \tag{4.2}$$

where the parameter τ specifies the decreasing rate of the weights put on the higher-order terms. This statistic is also called the geometrically weighted degree (GWD) statistic. Following Hunter, Goodreau and Handcock (2008), τ is fixed to 0.25 throughout this section. Fixing τ to be a constant is sensible, as $u(x|\tau)$ plays a role of explanatory variables for the ERGMs.

4.1.2 Shared Partnership. Following Hunter and Handcock (2006) and Hunter (2007), we define one type of shared partner statistics, the edgewise shared partner statistics, denoted by $EP_0(x)$, $\cdots$, $EP_{n-2}(x)$. The $EP_k(x)$ is the number of unordered pairs (i, j) such that $X_{ij} = 1$ and i and j have exactly k common neighbors. The geometrically weighted edgewise shared partnership (GWESP) statistic is defined as

$$v(x|\tau) = e^{\tau} \sum_{i=1}^{n-2} \left\{1 - \left(1 - e^{-\tau}\right)^i\right\} EP_i(x), \tag{4.3}$$

where the parameter τ specifies the decreasing rate of the weights put on the higher-order terms. Again, following Hunter et al. (2008), τ is fixed to 0.25.

Based on the statistics defined above, we consider three ERGMs with respective likelihood functions given by

$$f(\boldsymbol{x}|\theta) = \frac{1}{\kappa(\theta)} \exp\left\{\theta_1 e(x) + \theta_2 u(x|\tau)\right\} \quad \text{(Model 1)},$$

$$f(\boldsymbol{x}|\theta) = \frac{1}{\kappa(\theta)} \exp\left\{\theta_1 e(x) + \theta_2 v(x|\tau)\right\} \quad \text{(Model 2)},$$

$$f(\boldsymbol{x}|\theta) = \frac{1}{\kappa(\theta)} \exp\left\{\theta_1 e(x) + \theta_2 u(x|\tau) + \theta_3 v(x|\tau)\right\} \quad \text{(Model 3)}.$$

To conduct a Bayesian analysis for the models, the prior $\pi(\theta) = N_d(0, 10^2 I_d)$ was imposed on θ, where d is the dimension of θ and I_d is an identity matrix of size $d \times d$. Then MCMH can be applied to simulate from the posterior. The proposal distribution $Q(\cdot, \cdot)$ used here is a gaussian random walk proposal $N_d(\theta_t, s^2 I_d)$, and s is called the step size. In all simulations of this section, s was fixed to 0.2. Each auxiliary sample is generated through a cycle of Metropolis-within-Gibbs updates.

4.2 High School Student Friendship Network. The data were collected during the first wave (1994–1995) of the National Longitudinal Study of Adolescent Health(AddHealth) through a stratified sampling survey in the U.S. schools containing grades 7 through 12. To collect the data, the school administrator made a roster of all students and asked students to nominate five close male and female friends. Students were allowed to nominate their friends who were not in their school. The students could choose not to nominate if they did not have enough close male or female friends. The detailed description of the data can be found in Resnick et al. (1997), Udry and Bearman (1998), or online at http://www.cpc.unc.edu/projects/addhealth. The full data set contains 86 schools and 90,118 students. In this letter, we analyze only the subnetwork for school 10, which has 205 students, and consider only the undirected network for the case of mutual friendship.

MCMH-I was applied to this network with $m = 20$. For each model, MCMH-I was run five times independently. Each run started with a random point and consisted of 5000 iterations, where the first 1000 iterations were discarded for the burn-in process and the samples collected from the remaining iterations were used for estimation. The results are summarized in Table 5.

Since the exact sampler is not available for social networks, the MCMLE was also applied to this example for comparison. The software we used for MCMLE is an R package *ergm* by Hunter et al. (2008). MCMLE was also run five times for each model of this example. Each run consisted of 25 iterations with 6500 auxiliary networks generated at each iteration. In the *ergm* package, the auxiliary networks were simulated using the tie–no tie sampler (Morris, Handcock, & Hunter, 2008) with both the number of burn-in and the number of interval steps being set to 20,000. Under this setting, $1.3 \times 10^8 (= 20{,}000 \times 6{,}500)$ MH updates (each for one edge) are needed for generating 6500 networks at each iteration of MCMLE. The results, summarized in Table 5, indicate that MCMLE costs longer CPU

Table 5: Parameter Estimation for the AddHealth School 10 Network.

Method	Terms	Model 1	Model 2	Model 3
MCMH	Edge counts	$-3.922(7.0e-3)$	$-5.607(1.3e-2)$	$-5.507(3.7e-2)$
	GWD	$-1.545(1.6e-2)$		$-0.101(3.7e-2)$
	GWESP		$1.889(1.2e-2)$	$1.821(2.4e-2)$
	CPU(m)	33.6	33.5	60.1
MCMLE	Edge counts	$-3.977(5.3e-2)$	$-5.388(9.3e-3)$	$-5.170(1.5e-2)$
	GWD	$-1.297(4.3e-2)$		$-0.227(6.1e-3)$
	GWESP		$1.711(7.8e-3)$	$1.589(1.5e-2)$
	CPU(m)	45.1	48.9	70.8

Notes: The estimates were calculated by averaging over five independent runs with the standard Monte Carlo errors reported in the parentheses. CPU: CPU time (in minutes) cost by a single run on a 3.0 GHz Intel Core 2 Duo computer.

times than MCMH-I for this example. All computations for this example were done on a 3.0 GHz Intel Core 2 Duo computer.

To assess the accuracy of the MCMH estimates, the following procedure was proposed in a spirit similar to that of the parametric bootstrap method (Efron & Tibshirani, 1993), which calculated the root mean squared errors (RMSEs) of the estimates of $S_a(x)$'s. Since the statistics $\{S_a(x) : a \in A\}$ are sufficient for θ, if an estimate $\widehat{\theta}$ is accurate, then $S_a(x)$'s can be reverse-estimated by simulated networks from the distribution $f(x|\widehat{\theta})$. The procedure consists of three steps:

1. Given the estimate $\hat{\theta}$, simulate K networks, $x_1, \ldots, x_K$, independently using the Gibbs sampler.
2. Calculate the statistics $S_a(x)$, $a \in A$ for each of the simulated networks.
3. Calculate RMSE by the following equation,

$$RMSE(S_a) = \sqrt{\sum_{i=1}^{K} \left[S_a(x_i) - S_a(x)\right]^2 / K}, \quad a \in A, \tag{4.4}$$

where $S_a(x)$ is the corresponding statistic calculated from the network x.

In addition to RMSE, we calculate the absolute mean difference (AMD) for each statistic,

$$AMD(S_a) = \left| \frac{1}{K} \sum_{i=1}^{K} S_a(x_i) - S_a(x) \right|.$$

Table 6: RMSEs and AMDs of the MCMLE and MCMH Estimates for the ADDHealth School 10 Network.

		Model 1		Model 2		Model 3	
Method	Terms	RMSE	AMD	RMSE	AMD	RMSE	AMD
MCMH	Edge counts	32.449	2.672	26.998	2.252	22.993	10.821
	GWD	16.222	0.357			12.519	3.518
	GWESP			28.269	0.945	30.531	11.333
MCMLE	Edge counts	50.046	42.151	41.305	29.599	87.158	76.948
	GWD	26.964	24.497			27.609	25.277
	GWESP			33.180	14.568	70.470	56.189

With simple manipulations, it is easy to show that the following equalities hold at the MLE of θ:

$$E_\theta[S_a(X)] = S_a(x), \quad \forall\, a \in A, \tag{4.5}$$

where $E_\theta[\cdot]$ denotes the expectation with respect to the distribution $f(x|\theta)$ given in equation 4.1. Hence, AMD also provides a measure for the quality of the estimate of θ.

For each of the estimates shown in Table 5, the RMSEs and AMDs were calculated with $K = 1000$ and summarized in Table 6. The results indicate that MCMH-I produced much more accurate estimates than MCMLE for all three models. We note that Hunter et al. (2008) also applied MCMLE to models 1 and 2 for this network. Their estimate for model 2 is similar to ours, but their estimate for model 1 is not as close as ours. Hunter et al. (2008) reported the estimate of model 1 as $(-1.423, -1.305)$, for which the RMSE values are 4577.2 for the edge count and 90.011 for GWD. MCMH-I was also run with $m = 50$ for this network. The results were very similar.

Finally, we assessed the accuracy of the model estimates using the goodness-of-fit (GOF) plots (Hunter et al., 2008). The GOF plot shows the distribution (through box plots and confidence intervals) of three sets of statistics—the degree distribution, the edgewise shared partnership distribution, and the geodesic distance distribution—for the fitted model. If the statistics of the observed network, which are represented by a solid line in the GOF plots, fall into the confidence intervals of the fitted model, then the fitting is considered good. The closer the solid line is to the center of the box plots, the better the fitting is. Figure 4 compares the GOF plots for the two estimates of model 3. It indicates that MCMH-I provides a better fitting for the network than MCMLE. For the other two models, the GOF plots (omitted here) also indicate that MCMH-I works better than MCMLE for this example.

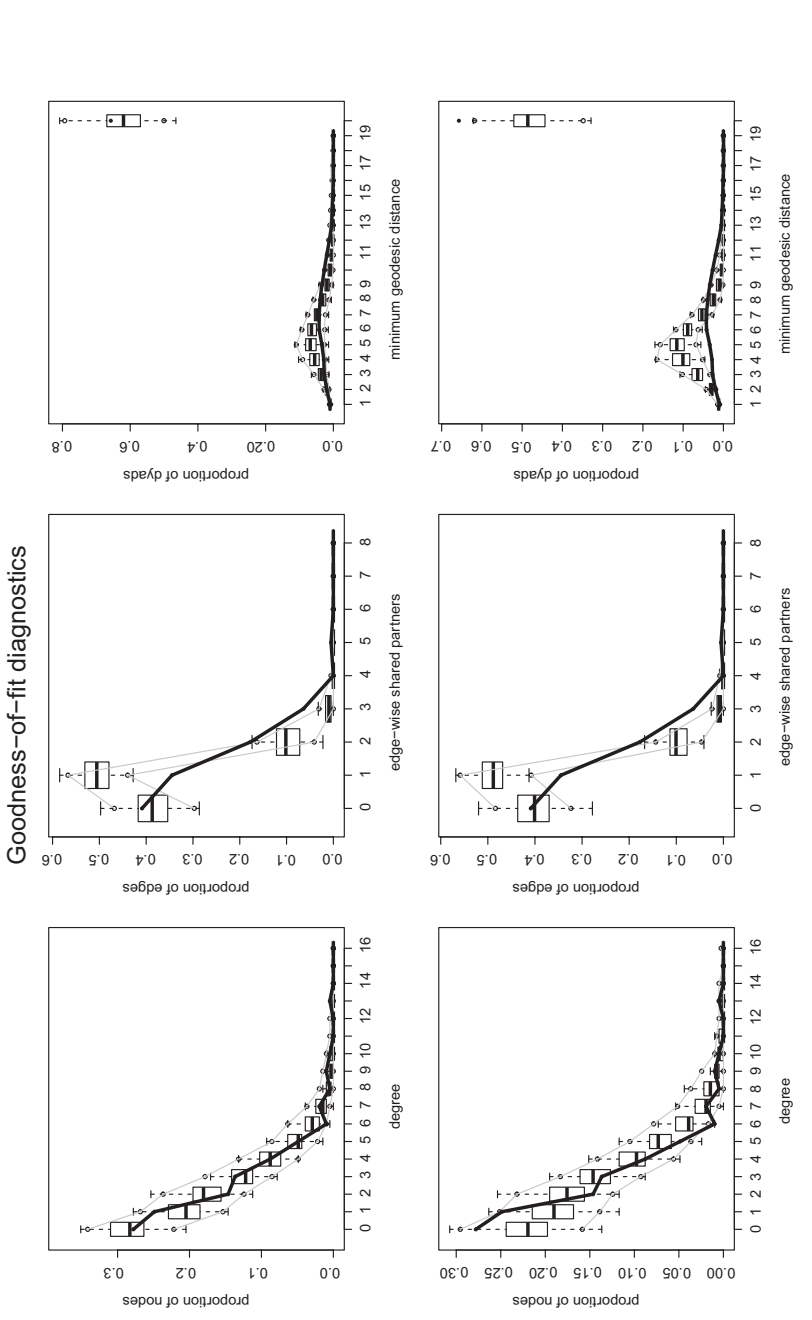

Figure 4: Goodness-of-fit plots for the high school student friendship network: (Top) MCMH-I estimate. (Bottom) MCMLE. The solid line shows the observed network statistics, and the box plots represent the distributions of simulated network statistics.

5 MCMH, GIMH, and Marginal Inference

In the literature, there is one algorithm, grouped independence MH (GIMH) (Beaumont, 2003), that is similar in spirit to the MCMH algorithm. GIMH is designed for marginal inference from a joint distribution.

Let $p(\theta, y)$ denote a joint distribution, let $p(\theta)$ denote the marginal distribution of θ, and let $p(y|\theta) = p(\theta, y)/p(\theta)$ denote the conditional distribution of y given θ. Suppose that we are interested in the marginal distribution $p(\theta)$. In Bayesian statistics, θ could represent a parameter of interest and y a set of missing data or latent variables. As implied by the Rao-Blackwell theorem (Bickel and Doksum, 2000), a basic principle in Monte Carlo computation is to carry out analytical computation as much as possible. Motivated by this principle, Beaumont (2003) proposed replacing $p(\theta)$ by its Monte Carlo estimate in simulations when the analytical form of $p(\theta)$ is not available. Let $\boldsymbol{y} = (y_1, \ldots, y_m)$ denote a set of independent and identically distributed (i.i.d.) samples drawn from a trial distribution $f_\theta(y)$. Note that $f_\theta(y)$ might not be equal to the conditional distribution $p(y|\theta)$. It follows from the standard theory of importance sampling that

$$\widetilde{p}(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{p(\theta, y_i)}{f_\theta(y_i)}, \tag{5.1}$$

forms an unbiased estimate of $p(\theta)$. In simulations, GIMH treats $\widetilde{p}(\theta)$ as a known target density, then simulates from it using the MH algorithm. Let θ_t denote the current draw of θ, and let $\boldsymbol{y}_t = (y_1^{(t)}, \ldots, y_m^{(t)})$ denote a set of i.i.d. auxiliary samples drawn from $f_\theta(y)$. One iteration of GIMH consists of the following steps:

Group Independence MH Algorithm

- Generate a new candidate point ϑ from a proposal distribution $Q(\theta_t, \vartheta)$.
- Draw m i.i.d. samples $\boldsymbol{y}' = (y_1', \ldots, y_m')$ from the trial distribution $f_\vartheta(y)$.
- Accept the proposal with probability

$$\min\left\{1, \frac{\widetilde{p}(\vartheta)}{\widetilde{p}(\theta_t)} \frac{Q(\vartheta, \theta_t)}{Q(\theta, \vartheta)}\right\}.$$

  If it is accepted, set $\theta_{t+1} = \vartheta$ and $\boldsymbol{y}_{t+1} = \boldsymbol{y}'$. Otherwise, set $\theta_{t+1} = \theta_t$ and $\boldsymbol{y}_{t+1} = \boldsymbol{y}_t$.

The convergence of the GIMH algorithm has been studied by Andrieu and Roberts (2009) under similar conditions to those assumed for MCMH in this letter. In the context of marginal inference, MCMH-I can be described

as follows. Let $\boldsymbol{y}_t = (y_1^{(t)}, \ldots, y_m^{(t)})$ denote a set of auxiliary samples drawn from the conditional distribution $p(y|\theta_t)$.

MCMH-I Algorithm (for marginal inference)

- Generate a new candidate point ϑ from a proposal distribution $Q(\theta_t, \vartheta)$.
- Accept the proposal with probability

$$\min\left\{1, \widetilde{R}(\theta_t, \vartheta)\frac{Q(\vartheta, \theta_t)}{Q(\vartheta, \theta_t)}\right\},$$

  where $\widetilde{R}(\theta_t, \vartheta) = \frac{1}{m}\sum_{i=1}^{m} p(\vartheta, y_i^{(t)})/p(\theta_t, y_i^{(t)})$ forms an unbiased estimate of the marginal density ratio $R(\theta_t, \vartheta) = \int p(\vartheta, y)dy / \int p(\theta_t, y)dy$. If it is accepted, set $\theta_{t+1} = \vartheta$; otherwise, set $\theta_{t+1} = \theta_t$.
- Set $\boldsymbol{y}_{t+1} = \boldsymbol{y}_t$ if a rejection occurs in the previous step. Otherwise, generate auxiliary samples $\boldsymbol{y}_{t+1} = (y_1^{(t+1)}, \ldots, y_m^{(t+1)})$ from the conditional distribution $p(y|\theta_{t+1})$. The auxiliary samples $y_1^{(t+1)}, \ldots, y_m^{(t+1)}$ can be generated using a MCMC simulation.

Taking a closer look at MCMH-I, we find that it is designed with a different rule from GIMH. First, GIMH estimates the marginal distributions, whereas MCMH-I directly estimates the ratio of marginal distributions. This leads to an important use of MCMH for simulating from distributions with intractable normalizing constants, the focus of this letter. Note that GIMH cannot be directly applied to this problem. Second, GIMH requires drawing samples from two distributions $f_\theta(\cdot)$ and $f_\vartheta(\cdot)$, while MCMH-I requires drawing samples from only a single distribution $p(\cdot|\theta)$. Thus, MCMH-I can be more efficient than GIMH for marginal inference. In addition, MCMH-I can recycle the auxiliary samples when a proposal is rejected, and this further improves its efficiency.

MCMH can potentially be applied to many statistical models for which marginal inference is of interest, such as generalized linear mixed models (see, e.g., McCulloch et al., 2008) and hidden Markov random field models (see, e.g., Rue & Held, 2005). MCMH can also be applied to Bayesian analysis for the missing data problems that are traditionally treated with the EM algorithm (Dempster, Laird, & Rubin, 1977) or the Monte Carlo EM algorithm (Wei & Tanner, 1990). Since the EM and Monte Carlo EM algorithms are local optimization algorithms, they tend to converge to suboptimal solutions. MCMH may perform better in this respect. Note that one may run MCMH under the framework of parallel tempering (Geyer, 1991) to help it escape from suboptimal solutions.

6 Conclusion

In this letter, we have proposed the new Monte Carlo Metropolis-Hastings algorithm for sampling from distributions with intractable normalizing

constants. The MCMH algorithm is a Monte Carlo version of the Metropolis-Hastings algorithm. At each iteration, it replaces the unknown normalizing constant ratio by a Monte Carlo estimate constructed based on auxiliary samples. Although the algorithm violates the detailed balance condition, it still converges, as we show in the letter, to the desired target distribution under mild conditions.

Unlike other auxiliary variable MCMC algorithms, such as the Møller and exchange algorithms, the MCMH algorithm avoids the requirement for perfect sampling and thus can be applied to many statistical models for which perfect sampling is not available or very expensive. Another advantage of the MCMH algorithm is that it can be easily run on a parallel architecture. The m auxiliary samples can be drawn from multiple short MCMC runs, with each running on a different node. For the problems for which simulating auxiliary samples is time-consuming, this will lead to a significant reduction of computational time.

As we discussed in section 5, the MCMH algorithm can also be applied to Bayesian inference for the random effect models and the missing data problems, which involve simulations from distributions with intractable integrals. Compared to the existing GIMH algorithm, the MCMH algorithm should be more efficient for these problems, as it recycles the auxiliary samples in simulations.

Appendix A: Proof of Theorems

Proof of Lemma 1. Since P defines an irreducible and aperiodic Markov chain, to show $\tilde{P}_m$ has the same property, it suffices to show that the accessible sets of P are included in those of $\tilde{P}_m$. More precisely, we show by induction that for any $k \in \mathbb{N}$, $\theta \in \Theta$, and $A \in \mathcal{B}(\Theta)$ such that $P^k(\theta, A) > 0$, then $\tilde{P}_m^k(\theta, A) > 0$. First, for any $\theta \in \Theta$ and $A \in \mathcal{B}(\Theta)$, it follows from equation 2.8 that

$$P(\theta, A) = \int_A \alpha(\theta, \vartheta) Q(\theta, d\vartheta) + \mathbb{I}(\theta \in A) \left[1 - \int_\Theta \alpha(\theta, \vartheta') Q(\theta, d\vartheta') \right].$$

Similarly, it follows from equations 2.4, 2.6, and 2.7 that

$$\begin{aligned}
\tilde{P}_m(\theta, A) &= \int_A \int_{\mathbb{Y}} (1 \wedge \gamma_m r(\theta, \vartheta)) f_\theta^m(d\boldsymbol{y}) Q(\theta, d\vartheta) + \mathbb{I}(\theta \in A) \rho(\theta) \\
&\geq \int_A \left[\int_{\mathbb{Y}} (1 \wedge \gamma_m) f_\theta^m(d\boldsymbol{y}) \right] (1 \wedge r(\theta, \vartheta)) Q(\theta, d\vartheta) + \mathbb{I}(\theta \in A) \rho(\theta) \\
&= \int_A \left[\int_{\mathbb{Y}} (1 \wedge \gamma_m) f_\theta^m(d\boldsymbol{y}) \right] \alpha(\theta, \vartheta) Q(\theta, d\vartheta) + \mathbb{I}(\theta \in A) \rho(\theta),
\end{aligned}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $a \wedge b = \min(a, b)$, $\alpha(\theta, \vartheta) = 1 \wedge r(\theta, \vartheta)$, and

$$r(\theta, \vartheta) = \frac{1}{R(\theta, \vartheta)} \frac{g(x, \vartheta)\pi(\vartheta)}{g(x, \theta)\pi(\theta)} \frac{Q(\vartheta, \theta)}{Q(\theta, \vartheta)}.$$

Therefore, for any set $A \in \mathcal{B}(\Theta)$, if $\theta \notin A$ and $P(\theta, A) > 0$, then $\int_A \alpha(\theta, \vartheta) Q(\theta, d\vartheta) > 0$ and thus $\tilde{P}_m(\theta, A) > 0$ by condition (A_2). If $\theta \in A$, regardless of the value of $P(\theta, A)$, we have $\tilde{P}_m(\theta, A) > 0$ by the assumption that $\rho(\theta) > 0$ for all $\theta \in \Theta$. The implication is that this true for $k = 1$.

Assume the induction assumption is true up to some $k = n \geq 1$. Now, for some $\theta \in \Theta$, let $A \in \mathcal{B}(\Theta)$ be such that $P^{n+1}(\theta, A) > 0$ and assume that

$$\int_\Theta \tilde{P}_m^n(\theta, d\vartheta) \tilde{P}_m(\vartheta, A) = 0,$$

which implies that $\tilde{P}_m(\vartheta, A) = 0$, $\tilde{P}_m^n(\theta, \cdot)$-a.s. and hence that $P(\vartheta, A) = 0$, $\tilde{P}_m^n(\theta, \cdot)$-a.s. from the induction assumption for $k = 1$. From this and the induction assumption for $k = n$, we deduce that $P(\vartheta, A) = 0$, $P^n(\theta, \cdot)$-a.s. (by contradiction), which contradicts the fact that $P^{n+1}(\theta, A) > 0$.

Proof of Lemma 2. Let

$$\begin{aligned} S &= P\psi(\theta) - \tilde{P}_m \psi(\theta) \\ &= \int_{\Theta \times \mathbb{Y}} \psi(\vartheta) \Big[1 \wedge r(\theta, \vartheta) - 1 \wedge \gamma_m r(\theta, \vartheta) \Big] Q(\theta, d\vartheta) f_\theta^m(d\boldsymbol{y}) \\ &\quad - \psi(\theta) \int_{\Theta \times \mathbb{Y}} \Big[1 \wedge r(\theta, \vartheta) - 1 \wedge \gamma_m r(\theta, \vartheta) \Big] Q(\theta, d\vartheta) f_\theta^m(d\boldsymbol{y}). \end{aligned}$$

We therefore focus on the quantity

$$\begin{aligned} S_0 &= \int_{\Theta \times \mathbb{Y}} \Big| 1 \wedge r(\theta, \vartheta) - 1 \wedge \gamma_m r(\theta, \vartheta) \Big| Q(\theta, d\vartheta) f_\theta^m(d\boldsymbol{y}) \\ &= \int_{\Theta \times \mathbb{Y}} \Big| 1 \wedge r(\theta, \vartheta) - 1 \wedge \gamma_m r(\theta, \vartheta) \Big| \mathbb{I}(\lambda_m > \epsilon) Q(\theta, d\vartheta) f_\theta^m(d\boldsymbol{y}) \\ &\quad + \int_{\Theta \times \mathbb{Y}} \Big| 1 \wedge r(\theta, \vartheta) - 1 \wedge \gamma_m r(\theta, \vartheta) \Big| \mathbb{I}(\lambda_m \leq \epsilon) Q(\theta, d\vartheta) f_\theta^m(d\boldsymbol{y}). \end{aligned}$$

Since, for any $(x, y) \in \mathbb{R}^2$,

$$|1 \wedge e^x - 1 \wedge e^y| = 1 \wedge |e^{0 \wedge x} - e^{0 \wedge y}| \leq 1 \wedge |x - y|,$$

we deduce that

$$S_0 \le Q(\theta, f_\theta^m(\mathbb{I}(\lambda_m > \epsilon))) + Q(\theta, f_\theta^m(1 \wedge \lambda_m \mathbb{I}(\lambda_m \le \epsilon))).$$

Consequently, we have

$$\begin{aligned}|S| &\le 2Q(\theta, f_\theta^m(\mathbb{I}(\lambda_m > \epsilon))) + 2Q(\theta, f_\theta^m(1 \wedge \lambda_m \mathbb{I}(\lambda_m \le \epsilon))) \\ &\le 2\epsilon + 2\epsilon = 4\epsilon.\end{aligned}$$

This completes the proof of lemma 2.

Proof of Theorem 1. For any $k \ge 1$ and any $\psi : \Theta \to [-1, 1]$, we have

$$\tilde{P}_m^k \psi(\theta_0) - \pi(\psi) = S_1(k) + S_2(k),$$

where $\pi(\psi) = \pi(\psi(\theta))$ for notational simplicity, and

$$S_1(k) = P^k \psi(\theta_0) - \pi(\psi), \qquad S_2(k) = \tilde{P}_m^k \psi(\theta_0) - P^k \psi(\theta_0).$$

For the term $S_2(k)$, we can further decompose it as follows. For any k_0 $(1 \le k_0 < k)$,

$$\begin{aligned}|S_2(k)| &\le |\tilde{P}_m^k \psi(\theta_0) - \tilde{P}_m^{k_0} \psi(\theta_0)| + |\tilde{P}_m^{k_0} \psi(\theta_0) - P^{k_0} \psi(\theta_0)| \\ &\quad + |P^{k_0} \psi(\theta_0) - P^k \psi(\theta_0)| \\ &= \left| \sum_{l=0}^{k_0 - 1} [P^l \tilde{P}_m^{k_0 - l} \psi(\theta_0) - P^{l+1} \tilde{P}_m^{k_0 - (l+1)} \psi(\theta_0)] \right| \\ &\quad + |\tilde{P}_m^k \psi(\theta_0) - \tilde{P}_m^{k_0} \psi(\theta_0)| + |P^k \psi(\theta_0) - P^{k_0} \psi(\theta_0)| \\ &= \left| \sum_{l=0}^{k_0 - 1} P^l (\tilde{P}_m - P) \tilde{P}_m^{k_0 - (l+1)} \psi(\theta_0) \right| + |\tilde{P}_m^k \psi(\theta_0) - \tilde{P}_m^{k_0} \psi(\theta_0)| \\ &\quad + |P^k \psi(\theta_0) - P^{k_0} \psi(\theta_0)|. \end{aligned} \tag{A.1}$$

For any $\epsilon > 0$, by lemma 2, there exists an $M(\epsilon, \theta_0)$ such that for any $m > M(\epsilon, \theta_0)$,

$$\begin{aligned}|S_2(k)| &\le 4k_0\epsilon + |\tilde{P}_m^k \psi(\theta_0) - \tilde{P}_m^{k_0} \psi(\theta_0)| + |P^k \psi(\theta_0) - P^{k_0} \psi(\theta_0)| \\ &= 4k_0\epsilon + S_3(m, k, k_0) + S_4(k, k_0),\end{aligned}$$

where Lemma 2 has been applied to equation A.1 k_0 times.

The magnitudes of $S_1(k)$, $S_4(k, k_0)$, and $S_3(m, k, k_0)$ can be controlled following from the convergence of the transition kernel P and lemma 1. For any $\epsilon > 0$, there exists $k_0 = k(\epsilon, \theta_0, m)$ such that for any $k > k_0$,

$$|S_1(k)| \le \epsilon, \quad S_3(m, k, k_0) \le \epsilon, \quad S_4(k, k_0) \le \epsilon.$$

Summarizing the results of $S_1(k)$ and $S_2(k)$, we conclude the proof by choosing $\epsilon = \varepsilon/(4k_0 + 3)$.

Proof of Theorem 2. This theorem can be proved as for theorem 1, as MCMH-I and MCMH-II share the same transition kernel.

Proof of Theorem 3. This theorem can be proved in the same way as for Theorem 1 except for some changes in notations.

Appendix B: Asymptotic Variances of Two Estimators of R

Suppose that we want to estimate the ratio $R = \kappa(\theta)/\kappa(\vartheta) = E_\vartheta[\frac{g(y,\theta)}{g(y,\vartheta)}]$, where $E_\vartheta[\cdot]$ denotes expectations with respect to $f(y|\vartheta) = g(y, \vartheta)/\kappa(\vartheta)$. In this section, we calculate the asymptotic variances of two estimators of R, which are given as follows:

$$\hat{R}_1 = \left[\frac{1}{n}\sum_{i=1}^{n}\frac{g(x_i, \vartheta)}{g(x_i, \theta)}\right]^{-1},$$

where $x_1, \ldots, x_n$ denote n i.i.d. samples drawn from $f(x|\theta)$, and

$$\hat{R}_2 = \frac{1}{n}\sum_{i=1}^{n}\frac{g(y_i, \theta)}{g(y_i, \vartheta)},$$

where $y_1, \ldots, y_n$ denote n i.i.d. samples drawn from $f(y|\vartheta)$.

If we define $Z_i = g(x_i, \theta)/g(x_i, \vartheta)$, then $\hat{R}_1$ can be viewed as a harmonic mean estimator of $E(Z_i)$. Under regularity conditions, it is easy to show

$$\sqrt{n}(\hat{R}_1 - R) \longrightarrow N\left(0, \frac{\mathrm{Var}(Z_i^{-1})}{(E(Z_i^{-1}))^4}\right),$$

as $n \to \infty$. Direct calculations yield $E(Z_i^{-1}) = 1/R$ and

$$\mathrm{Var}(Z_i^{-1}) = \frac{1}{R^2}E_\theta\left(\frac{f(x|\vartheta) - f(x|\theta)}{f(x|\theta)}\right)^2.$$

Thus, the asymptotic variance of $\hat{R}_1$ is

$$\sigma_1^2(\theta, \vartheta) = R^2 E_\theta \left(\frac{f(x|\vartheta) - f(x|\theta)}{f(x|\theta)} \right)^2 = R^2 \mathrm{Var}_\theta(W_x),$$

where $W_x = f(x|\vartheta)/f(x|\theta)$ and $\mathrm{Var}_\theta(\cdot)$ denotes the variance with respect to the distribution $f(x|\theta)$. Similarly, the asymptotic variance of $\hat{R}_2$ is

$$\sigma_2^2(\theta, \vartheta) = R^2 E_\vartheta \left(\frac{f(y|\theta) - f(y|\vartheta)}{f(y|\vartheta)} \right)^2 = R^2 \mathrm{Var}_\vartheta(W_y),$$

where $W_y = f(y|\theta)/f(y|\vartheta)$ and $\mathrm{Var}_\vartheta(\cdot)$ denotes the variance with respect to the distribution $f(y|\vartheta)$. Hence, the ordering of $\sigma_1^2(\theta, \vartheta)$ and $\sigma_2^2(\theta, \vartheta)$ depends on the values of θ and ϑ.

Acknowledgments

We thank the editor, associate editor, and two referees for their comments, which have led to significant improvement of this letter. F:L.'s research was partially supported by grants from the National Science Foundation (DMS-1007457 and DMS-1106494) and the award (KUS-C1-016-04) made by King Abdullah University of Science and Technology (KAUST).

References

Andrieu, C., & Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, *37*, 697–725.

Bartz, K., Blitzstein, J., & Liu, J. S. (2008). *Monte Carlo maximum likelihood for exponential random graph models: From snowballs to umbrella densities* (Tech. Rep.). Cambridge, MA: Department of Statistics, Harvard University.

Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, *164*, 1139–1160.

Besag, J. E. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. R. Statist. Soc. B*, *36*, 192–236.

Bickel, P. J., & Doksum, K. A. (2000). *Mathematical statistics: Basic ideas and selected topics* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Chen, M.-H., & Shao, Q.-M. (1997). On Monte Carlo methods for estimating ratios of normalizing constants. *Annals of Statistics*, *25*, 1563–1594.

Childs, A. M., Patterson, R. B., & MacKay, D. J. C. (2001). Exact sampling from nonattractive distributions using summary states. *Phys. Rev. E.*, *63*, 036113.

Corander, J., Dahmström, K., & Dahmström, P. (1998). *Maximum likelihood estimation for Markov graphs* (Research Rep. 8). Stockholm: Department of Statistics, University of Stockholm.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via EM algorithm. *J. R. Stat. Soc. B*, *39*, 1–38.

Efron, B., & Tibshirani, R. J. (1993), *An introduction to the bootstrap*. London: Chapman & Hall.

Frank, I., & Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, *81*, 832–842.

Gelman, A., & Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, *13*, 163–185.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Sciences*, *7*, 457–511.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In E. M. Keramigas (Ed.), *Computing science and statistics: Proceedings of the 23rd Symposium on the Interface* (pp. 156–163). Fairfax: Interface Foundation.

Geyer, C., & Thompson, E. (1992). Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society, Series B*, *54*, 657–699.

Green, P. J., & Richardson, S. (2002). Hidden Markov models and disease mapping. *Journal of the American Statistical Association*, *97*, 1055–1070.

Hunter, D. (2007). Curved exponential family models for social network. *Social Networks*, *29*, 216–230.

Hunter, D. R., Goodreau, S. M., & Handcock, M. S. (2008). Goodness of fit of social network models. *Journal of the American Statistical Association*, *103*, 248–258.

Hunter, D., & Handcock, M. (2006). Inference in curved exponential family models for network. *Journal of Computational and Graphical Statistics*, *15*, 565–583.

Hurn, M., Husby, O., & Rue, H. (2003). A tutorial on image analysis. *Lecture Notes in Statistics*, *173*, 87–141.

Liang, F. (2007). Continuous contour Monte Carlo for marginal density estimation with an application to a spatial statistical model. *J. Comput. Graph. Statist.*, *16*, 608–632.

Liang, F. (2010). A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computing and Simulation*, *80*, 1007–1022.

Lubbers, M., & Snijders, T. A. B. (2007). A comparison of various approaches to the exponential random graph model: A reanalysis of 104 student networks in school classes. *Social Networks*, *29*, 489–507.

Madras, N. (2000). *Lectures on Monte Carlo Methods*. Providence, RI: American Mathematical Society.

McCulloch, C. E., Searle, S. R., & Neuhaus, J. M. (2008). *Generalized, linear, and mixed models* (2nd ed.). Hoboken, NJ: Wiley.

Meng, X.-L., & Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: A theoretical exploration. *Statistica Sinica*, *6*, 831–860.

Møller, J., Pettitt, A. N., Reeves, R., & Berthelsen, K. K. (2006). An efficient Markov chain Monte Carlo method for distributions with intractable normalizing constants. *Biometrika*, *93*, 451–458.

Morris, M., Handcock, M. S., & Hunter, D. R. (2008). Specification of exponential-family random graph models: Terms and computational aspects. *Journal of Statistical Software*, *24* (4). http://www.jstatsoft.org/v24/i04

Murray, I., Ghahramani, Z., & MacKay, D. J. C. (2006). MCMC for doubly-intractable distributions. In *Proc. 22nd Annual Conference on Uncertainty in Artificial Intelligence*. San Francisco: Morgan Kaufmann.

Preisler, H. K. (1993). Modeling spatial patterns of trees attacked by bark-beetles. *Appl. Statist.*, *42*, 501–514.

Propp, J. G., & Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, *9*, 223–252.

Resnick, M. D., Bearman, P. S., Blum, R. W., Bauman, K. E., Harris, K. M., Jones, J., et al. (1997). Protecting adolescents from harm: Findings from the national longitudinal study on adolescent health. *Journal of the American Medical Association*, *278*, 823–832.

Riggan, W. B., Creason, J. P., Nelson, W. C., Manton, K. G., Woodbury, M. A., Stallard, E., et al. (1987). *U.S. cancer mortality rates and trends*, 1950–1979. Washington, DC: U.S. Government Printing Office.

Robins, G., Pattison, P., Kalish, Y., & Lusher, D. (2007). An introduction to exponential random graph (p^*) models for social networks. *Social Networks*, *29*, 173–191.

Rue, H., & Held, L. (2005), *Gaussian Markov random fields: Theory and applications*. London: Chapman & Hall/CRC.

Sherman, M., Apanasovich, T. V., & Carroll, R. J. (2006). On estimation in binary autologistic spatial models. *J. Statist. Comput. Simu.*, *76*, 167–179.

Snijders, T. A. B. (2002). Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure*, *3*, article 2.

Snijders, T. A. B., Pattison, P. E., Robins, G. L., & Handcock, M. S. (2006). New specifications for exponential random graph models. *Sociological Methodology*, *36*, 99–153.

Strauss, D., & Ikeda, M. (1990). Pseudo-likelihood estimation for social Network. *Journal of the American Statistical Association*, *82*, 204–212.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, *22*, 1701–1762.

Torrie, G. M., & Valleau, J. P. (1997). Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *Journal of Chemical Physics*, *23*, 187–199.

Udry, J. R., & Bearman, P. S. (1998). New methods for new research on adolescent sexual behavior. In R. Jessor (Ed.), *New perspectives on adolescent risk behavior* (pp. 241–269). Cambridge: Cambridge University Press.

van Duijn, M. A. J., Gile, K. J., & Handcock, M. S. (2009). A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Social Networks*, *31*, 52–62

Wasserman, S., & Robins, G. (2005). An introduction to random graphs, dependence graphs, and p^*. In P. J. Carrington, J. Scott, & S. Wasserman (Eds.), *Models and methods in social network analysis* (pp. 148–191). Cambridge: Cambridge University Press.

Wei, G., & Tanner, M. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithm. *Journal of the American Statistical Association*, *85*, 699–704.

Wu, H., & Huffer, F. W. (1997). Modeling the distribution of plant species using the autologistic regression model. *Ecological Statistics*, *4*, 49–64.

Received December 18, 2011; accepted January 29, 2013.