STATISTICAL INFERENCES FOR MODELS WITH INTRACTABLE NORMALIZING CONSTANTS

A Dissertation

by

ICK HOON JIN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2011

Major Subject: Statistics

STATISTICAL INFERENCES FOR MODELS WITH INTRACTABLE NORMALIZING CONSTANTS

A Dissertation

by

ICK HOON JIN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Faming Liang
Committee Members,	David B. Dahl
	Samiran Sinha
	Byung-Jun Yoon
Head of Department,	Simon Sheather

August 2011

Major Subject: Statistics

ABSTRACT

Statistical Inferences for Models with Intractable Normalizing Constants. (August 2011)

Ick Hoon Jin, B.A., Yonsei University; M.A., Yonsei University

Chair of Advisory Committee: Dr. Faming Liang

In this dissertation, we have proposed two new algorithms for statistical inference for models with intractable normalizing constants: the Monte Carlo Metropolis-Hastings algorithm and the Bayesian Stochastic Approximation Monte Carlo algorithm. 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. Although the algorithm violates the detailed balance condition, it still converges, as shown in the paper, to the desired target distribution under mild conditions. The BSAMC algorithm works by simulating from a sequence of approximated distributions using the SAMC algorithm. A strong law of large numbers has been established for BSAMC estimators under mild conditions. One significant advantage of our algorithms over the auxiliary variable MCMC methods is that they avoid the requirement for perfect samples, and thus it can be applied to many models for which perfect sampling is not available or very expensive. In addition, although the normalizing constant approximation is also involved in BSAMC, BSAMC can perform very robustly to initial guesses of parameters due to the powerful ability of SAMC in sample space exploration. BSAMC has also provided a general framework for approximated Bayesian inference for the models for which the likelihood function is intractable: sampling from a sequence of approximated distributions with their average converging to the target distribution. With these two illustrated algorithms,

we has demonstrated how the SAMCMC method can be applied to estimate the parameters of ERGMs, which is one of the typical examples of statistical models with intractable normalizing constants. We showed that the resulting estimate is consistent, asymptotically normal and asymptotically efficient. Compared to the MCMLE and SSA methods, a significant advantage of SAMCMC is that it overcomes the model degeneracy problem. The strength of SAMCMC comes from its varying truncation mechanism, which enables SAMCMC to avoid the model degeneracy problem through re-initialization. MCMLE and SSA do not possess the re-initialization mechanism, and tend to converge to a solution near the starting point, so they often fail for the models which suffer from the model degeneracy problem.

To Youn Sil

ACKNOWLEDGMENTS

I am grateful to my dissertation advisor, Prof. Faming Liang and committee members for their interaction and support during my graduate study. This dissertation is dedicated to Youn Sil and my family for their endless encouragement, support and love.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In statistical applications, 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 [82], the Potts model used in image analysis [42], the autonormal model used in agriculture experiments [9], and the exponential random graph model used in social network study [75], among others.

Suppose we have a dataset X generated from a statistical model with the likelihood function

$$f(x|\theta) = \frac{1}{\kappa(\theta)} \exp\{-U(x,\theta)\}, \quad x \in \mathcal{X},\ \theta \in \Theta, \tag{1.1}$$

where θ is the parameter, and $\kappa(\theta)$ is the normalizing constant which 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

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

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

The MH algorithm cannot be applied to simulate from $\pi(\theta|x)$, because the acceptance probability would involve an unknown ratio $\kappa(\theta)/\kappa(\theta')$, where θ' denotes the

---

The journal model is *IEEE Transactions on Automatic Control*.

proposed value. To circumvent this difficulty, various approximation methods to the likelihood function or the normalizing constant function have been proposed in the literature. [9] proposed to approximate the likelihood function by a pseudo-likelihood function which is tractable. The method is easy to use, but it typically performs less well for the models for which neighboring dependence is strong. [32] 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 MCMC simulation. Then

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

approaches to $\log f(x|\theta)$ as $m \to \infty$. The estimator $\hat{\theta} = \arg\max_\theta \log f_m(x|\theta)$ is called the MCMLE of θ. If $\theta^{(0)}$ lies in the attraction region of true MLE, the method usually produces a good estimate of θ. Otherwise, the method may converge to a suboptimal solution or fail to converge. To alleviate this difficulty, [32] recommended an iterative approach, which drew new samples at the current estimate of θ and then re-estimate:

(a) Initialize with a point $\theta^{(0)}$, usually taking to be the maximum pseudo-likelihood estimator. Set $t = 0$.

(b) Simulate m auxiliary samples from $f(\boldsymbol{x}|\theta^{(t)})$ using MCMC.

(c) Find $\theta^{(t+1)} = \arg\max_\theta \log f_m(\boldsymbol{x}|\theta)$.

(d) Stop if a specified number of iterations has been reached, or some other termination criterion has reached. Otherwise, go back to step (b).

Even with this iterative approach, non-convergence is still quite common if $\theta^{(0)}$ is far from the true MLE. [46] proposed an alternative Monte Carlo approach to approxi-

mate $\kappa(\theta)$, where $\kappa(\theta)$ is viewed as a marginal density function of the unnormalized distribution $g(x,\theta) = \exp\{-U(x,\theta)\}$ and estimated using an adaptive kernel smoothing approach with Monte Carlo samples.

Toward Bayesian analysis for the model (1.1), a significant step was made by [55], who propose to augment the distribution $f(x|\theta)$ by an auxiliary variable such that the normalizing constant ratio $\kappa(\theta)/\kappa(\theta')$ can be canceled in simulations. Soon, this algorithm was improved by [57], who, based on the idea of parallel tempering [31], proposed the following algorithm—the exchange algorithm:

Exchange Algorithm

- Propose a candidate point θ' from a proposal distribution denoted by $q(\theta'|\theta, x)$.
- Generate an auxiliary variable $y \sim f(y|\theta')$ using a perfect sampler [61].
- Accept θ' with probability $\min\{1, r(\theta, \theta'|x)\}$, where

$$r(\theta, \theta'|x) = \frac{\pi(\theta')f(x|\theta')f(y|\theta)q(\theta|\theta', x)}{\pi(\theta)f(x|\theta)f(y|\theta')q(\theta'|\theta, x)}.$$

Since a swapping operation between (θ, x) and (θ', y) is involved, the algorithm is called the exchanged 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 it avoids an initial estimation step (for θ) that required by the Møller algorithm. See [55] for the role that an initial estimate of θ plays in their algorithm. [57] reported that 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.

In Chapter II, we introduce the exponential random graph models (ERGMs) which is one of the well-known statistical models with intractable normalizing constants. In Chapter III, we describe the Monte Carlo Metropolis-Hasting algorithm, which replaces the unknown normalizing constant ratio $\kappa(\theta)/\kappa(\theta')$ by a Monte Carlo estimate to handle intractable normalizing constants problems. In Chapter IV, we illustrate the Bayesian Stochastic Approximation Monte Carlo algorithm, which works by simulating from a sequence of approximated distributions using the stochastic approximation Monte Carlo algorithm [50]. for tickling intractable normalizing constants problems. In Chapter V, we propose to use the stochastic approximation MCMC (SAMCMC) algorithm to find the maximum likelihood estimator for ERGMs. We conclude our statistical methods for models with intractable normalizing constants in Chapter VI.

CHAPTER II

EXPONENTIAL RANDOM GRAPH MODELS

A. Introduction

The social network is a social structure made of actors (individuals, organizations, etc.) which are interconnected by certain relationship, such as friendship, common interest, financial exchange, etc. The network can be represented in a graph with a node for each actor and an edge for each relation between a pair of actors. This graph representation can provide insight into organizational structures, social behavior patterns, and a variety of other social phenomena. Recently, social network analysis has been applied to many other fields, such as biology [71], political science [23], etc. etc.

Many statistical models have been proposed in the literature for social network analysis, including the dyadic independence model, the Markov random graph model [27], the exponential random graph model [75], among others. The model of particular interest is the exponential random graph model (ERGM), which allows to include various network dependent structures in the analysis and thus generally improves goodness of fit of social networks. See [68] for an overview of ERGMs.

Consider a social network with n actors. The network can be specified in an $n \times n$-matrix $\mathbf{Y} = (Y_{ij})$, where $Y_{ij} = 1$ if there is an edge between node i and node j and 0 otherwise. This matrix is also known as the adjacency matrix. Note that the social network can be either directed or non-directed. The likelihood function of the

ERGM is given by

$$f(\boldsymbol{y}|\theta) = \frac{1}{\kappa(\theta)} \exp\{\sum_{i \in A} \theta_i S_i(\mathbf{y})\}, \tag{2.1}$$

where $S_i(\boldsymbol{y})$ denotes a statistic, θ_i is the corresponding parameter, A specifies the set of statistics considered in the model, and $\kappa(\theta)$ is the normalizing constant which makes (2.1) a proper probability distribution. An exact calculation of $\kappa(\theta)$ is impossible for all but the smallest networks, as it involves a sum over all possible networks. In the rest of this paper, we will let $\boldsymbol{S}(\boldsymbol{y}) = (S_1(\boldsymbol{y}), \ldots, S_d(\boldsymbol{y}))$ denote the vector of d statistics considered in the model, and let $\theta = (\theta_1, \ldots, \theta_d)$ denote the vector of d parameters of the model.

Parameter estimation for ERGMs suffers from two difficulties. The first difficulty is due to the intractability of $\kappa(\theta)$, and the second is the so-called model degeneracy problem. They will be discussed in sequel as follows.

B. Parameter Estimation Methods

Because $\kappa(\theta)$ in (2.1) is intractable, estimation of θ has put a great challenge on the current statistical methods. Several methods have been proposed in the literature, including the maximum pseudo-likelihood estimation (MPLE) method [76], Monte Carlo maximum likelihood estimation (MCMLE) method ([32], [40]), stochastic approximation (SA) method [74], among others.

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 has been studied by many authors, see e.g., [22], [25], [51] and [80]. MPLE is intrinsically highly dependent on the observed network. It usually works well for the networks with low dependence structure, but may produce substantially biased estimates for the networks with high dependency.

The MCMLE method originates in [32], whose basic idea is to approximate the normalizing constant $\kappa(\theta)$ using Monte Carlo samples. It is known that the performance of this method depends on the choice of an initial guess. If the initial guess is near the MLE, it can produce a good estimate of θ. Otherwise, it may converge to a local optimal solution or even fail to converge. To alleviate this difficulty, [32] recommended an iterative approach, which drew new samples at the current estimate of θ and then re-estimate. Even with this iterative procedure, as pointed out by [7], non-convergence is still quite common for ERGMs.

With some simple manipulations, it is easy to show that maximizing the likelihood function (2.1) is equivalent to solve the system of equations

$$E_\theta\left(\boldsymbol{S}(\boldsymbol{Y})\right) = \boldsymbol{S}(\boldsymbol{y}_{obs}), \tag{2.2}$$

where the expectation is taken with respect to the distribution $f(\boldsymbol{y}|\theta)$ as specified in (2.1). The rationale underlying this reformulation is the exponential family theory ([6]; [14]), which says that the MLE of (2.1), if existing, is the unique vector $\hat{\theta}$ such that (2.2) holds. [74] applies the stochastic approximation algorithm [64] to solve (2.2) for θ. In this paper, we call this method the SSA method. One iteration of SSA consists of two main steps:

(a) (Independence network generation) Generate an independent sample $\boldsymbol{y}^{(k+1)}$ from the distribution $f(\boldsymbol{y}|\theta^{(k)})$: Starting with a random graph in which each arc variable Y_{ij} is determined independently with a probability 0.5 for the values 0 and 1; and then updating the random graph using the Gibbs sampler [30] or the MH algorithm [36] and [54].

(b) (Estimate updating) Set

$$\theta^{(k+1)} = \theta^{(k)} - a_k D^{-1}\big(U(\boldsymbol{y}_{k+1}, \bar{\boldsymbol{y}}_{k+1}) - \boldsymbol{S}(\boldsymbol{y}_{obs})\big), \tag{2.3}$$

where $\{a_k\}$ denotes a positive sequence converging to 0, D denotes a pre-estimated covariance matrix of $\boldsymbol{S}(\boldsymbol{Y})$ at the initial estimate $\theta^{(1)}$, $\bar{\boldsymbol{y}}_{k+1} = 1 - \boldsymbol{y}_{k+1}$ denotes the complementary network of $\boldsymbol{y}_{k+1}$ (with each cell of the adjacency matrix of $\boldsymbol{y}_{k+1}$ being switched from 0 to 1 and vice versa),

$$U(\boldsymbol{y}_{k+1}, \bar{\boldsymbol{y}}_{k+1}) = P(\bar{\boldsymbol{y}}_{k+1}|\boldsymbol{y}_{k+1})\boldsymbol{S}(\bar{\boldsymbol{y}}_{k+1}) + (1 - P(\bar{\boldsymbol{y}}_{k+1}|\boldsymbol{y}_{k+1}))\boldsymbol{S}(\boldsymbol{y}_{k+1}),$$

and $P(\bar{\boldsymbol{y}}_{k+1}|\boldsymbol{y}_{k+1})$ denotes the MH acceptance probability of the transition from $\boldsymbol{y}_{k+1}$ to $\bar{\boldsymbol{y}}_{k+1}$.

A major shortcoming of SSA is its inefficiency in generating independent network samples. The number of updating steps for generating each sample $\boldsymbol{y}_{k+1}$ is in the order of $100n^2$, where n denotes the total number of nodes included in the network. This is very time consuming when n is large.

C. Model Degeneracy

The model degeneracy problem [34] refers to the phenomenon that for some configurations of θ, the model (2.1) produces networks that are either full (every tie exists) or empty (no ties exist) with probability close to one. For example, the models with basic Markovian statistics (e.g., the number of triangles) often suffer from the model degeneracy problem. When one edge is added to or removed from the network, the values of the basic Markovian statistics can change a lot while the values of other statistics do not change proportionally, so the dyadic dependence effects amplify quickly and the model tend to be degenerated. When the observed network is fitted by such a model, the MCMLE and SSA method may produce a degenerated estimate of θ (i.e., the estimate falls in a degeneracy region) if the starting value is in or close to a degeneracy region. In this case, the resulting model will not provide

a good fitting to the network. The reason why MCMLE and SSA often fail for the model degeneracy problem is due to their local convergence property, i.e., they tend to converge to a local optimal solution near the starting point.

As pointed out by [35], the model degeneracy problem can also be viewed as a model mis-specification problem. A solution to avoid this problem is to specify a model whose parameter space contains no or less degeneracy regions. However, this is often more difficult than usual. For a linear model, the mis-specification can be diagnosed by comparing observed to predicted values; but for ERGMs, if the model is mis-specified, the analyst can be left with little information to help guide the re-specification of the model.

D. Network Statistics

Recall the likelihood function given in (2.1). To define ERGMs, it is necessary to specify the sufficient statistics $\boldsymbol{S}(\boldsymbol{y})$ explicitly. Since a large number of specifications are available for ERGMs, we consider only several commonly used statistics in this article, including basic Markovian statistics [27], the degree distribution, the edge-wise shared partnership distribution [75], and nodal covariates. The basic Markovian statistics, which consist of edge counts, k2-star, k3-star, and triangle counts, describe the basic structure of social networks. The degree distribution and the edgewise shared partnership distribution describe the higher order transitivity of social networks. Nodal covariates introduce actor attributes into ERGMs. See [68] and [69] for overviews of ERGMs.

1. Basic Markovian Statistics

The edge counts, denoted by $S_1(\boldsymbol{y})$, is the count of edges contained in the social network $\boldsymbol{y}$. If one node connects to other two nodes, it is called 2-star. In the same manner, if one node connects to other three nodes, it is called 3-star. The counts of 2-stars and 3-star are called k2-star and k3-star, and denoted by $S_2(\boldsymbol{y})$ and $S_3(\boldsymbol{y})$, respectively. If node 'a' connects to node 'b', node 'b' connects to node 'c', and node 'c' connects to node 'a' simultaneously, then the nodes 'a', 'b' and 'c' form a triangle. The count of triangles is denoted by $T(\boldsymbol{y})$. Mathematically, the statistics $S_k(\boldsymbol{y})$ $(k = 2, 3)$ and $T(\boldsymbol{y})$ can be calculated by

$$S_1(\boldsymbol{y}) = \sum_{1\le i<j\le G} y_{ij}; \quad S_k(\boldsymbol{y}) = \sum_{1\le i\le G} \binom{y_{i+}}{k}, \quad k = 2, 3; \quad T(\boldsymbol{y}) = \sum_{1\le i<j<h\le G} y_{ij}y_{ih}y_{jh}, \tag{2.4}$$

where y_{i+} denotes the degree of node i.

2. Degree

Let $D_i(\boldsymbol{y})$ denote the number of nodes whose degree, the number of edges incident to the node, equals i. The statistics $D_0(\boldsymbol{y}), \ldots, D_{G-1}(\boldsymbol{y})$ satisfy the constraint $\sum_{i=0}^{G-1} D_i(\boldsymbol{y}) = G$, and the edge count statistic can be re-expressed as $S_1(\mathbf{y}) = \frac{1}{2}\sum_{i=1}^{G-1} iD_i(\mathbf{y})$.

The geometrically weighted degree statistic ([38], [40], and [75]) is defined by

$$u(\boldsymbol{y}|\tau) = e^{\tau}\sum_{i=1}^{G-2}\left\{1 - \left(1 - e^{-\tau}\right)^i\right\}D_i(\boldsymbol{y}), \tag{2.5}$$

where the additional parameter τ specifies the decreasing rate of weights put on the higher order terms. Following [39], we fix τ to be a constant throughout this paper. Although treating τ as an unknown parameter can potentially improve the model fitting, the distribution (2.1) will no longer satisfy the form of exponential family.

Rather it belongs to a curved exponential family.

3. Shared Partnership

Let $EP_k(\boldsymbol{y})$ denote the number of unordered pairs (i, j) for which i and j have exactly k common neighbors and $Y_{ij} = 1$. Let $DP_k(\boldsymbol{y})$ denote the number of unordered pairs (i, j) for which i and j have exactly k common neighbors regardless the value of Y_{ij}. In the literature, $EP_k(\boldsymbol{y})$ is called the edge-wise shared partnership statistic and $DP_k(\boldsymbol{y})$ the dyad-wise shared partnership statistic. They satisfy the constraint $\sum_{k=0}^{G-2} EP_k(\boldsymbol{y}) = S_1(\boldsymbol{y})$ and $\sum_{k=0}^{G-2} DP_k(\boldsymbol{y}) = \binom{n}{2}$. The geometrically weighted edgewise shared partnership (GWESP) statistic and geometrically weighted dyadwise shared partnership (GWDSP) statistic ([38], [40], and [75]) are defined by

$$v(\boldsymbol{y}|\tau) = e^{\tau} \sum_{i=1}^{G-2} \left\{ 1 - \left(1 - e^{-\tau}\right)^i \right\} EP_i(\boldsymbol{y}), \tag{2.6}$$

$$w(\boldsymbol{y}|\tau) = e^{\tau} \sum_{i=1}^{G-2} \left\{ 1 - \left(1 - e^{-\tau}\right)^i \right\} DP_i(\boldsymbol{y}), \tag{2.7}$$

where the parameter τ specifies the decreasing rate of weights put on the higher order terms. As for the GWD statistic, τ is fixed to a constant throughout this paper.

4. Nodal covariates

Nodal covariates represent specific features of a node. Let X_i denote a covariate of node i. All nodal covariates can be expressed as a dyadic independence statistic of the form

$$\sum \sum_{i<j} y_{ij} h(X_i, X_j) \tag{2.8}$$

for a suitably chosen function $h(X_i, X_j)$ [39]. In this paper, we consider a few types of nodal covariates, including the main factor effect, nodal factor effect, homophily

factor effect and absolute difference factor effect. The latter three are usually used for categorical factors and their statistics take values of 0 or natural numbers.

The main factor effect directly adds covariates of nodes i and j; that is,

$$h(X_i, X_j) = X_i + X_j. \tag{2.9}$$

For each edge, the nodal factor effect gives the node a score according to the counts of endpoints which have the specified factor level. It is defined by

$$h(X_i, X_j) = \begin{cases} 2, & \text{if both } i \text{ and } j \text{ have the specified factor level,} \\ 1, & \text{if exactly one of } i, j \text{ has the specified factor level,} \\ 0, & \text{if neither } i \text{ nor } j \text{ has the specified factor level.} \end{cases} \tag{2.10}$$

Since the sum of nodal factor effects for all levels are equal to twice the edge counts of the network, one level must be excluded in nodal factor effects to remove the linear dependency.

The homophily factor effect gives each edge a score of 0 or 1, depending on whether or not the two endpoints have the same factor level. There are two types of homophily factor effects: uniform homophily factor effect and differential homophily factor effect. The former is defined by

$$h(X_i, X_j) = \begin{cases} 1, & \text{if } i \text{ and } j \text{ have the same factor level,} \\ 0, & \text{otherwise,} \end{cases} \tag{2.11}$$

and the latter is defined by

$$h(X_i, X_j) = \begin{cases} 1, & \text{if } i \text{ and } j \text{ have the specified factor level,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.12}$$

For ordinal factors, one may expect that the nodes with smaller absolute dif-

ferences in covariates tend to connect by edges. To incorporate this effect into the models, we add a new statistic, the so-called absolute difference factor effect, into the model. This effect is defined by

$$h(X_i, X_j) = \begin{cases} 1 & \text{if } |X_i - X_j| = C \text{ for some nonzero constant } C, \\ 0 & \text{otherwise.} \end{cases} \tag{2.13}$$

If $C = 0$, it would introduce a linear dependence with the homophily factor effect.

5. Summary

In summary, the network statistics can be generally classified into two groups: dyadic dependent and dyadic independent [35], where a dyad refers to a pair of nodes. Dyadic independence means there are no direct dependence among dyads; that is, the state of a dyad is independent of the state of other dyads. The edge counts and nodal covariate terms are dyadic independent statistics. Dyadic dependence means the state of one dyad stochastically depends on the state of other dyad. An example is "the friend of my friend is also my friend"— edges in dyad (i, j) and (j, k) increase the probability of relation in dyad (i, k). The k-star, triangle, degree and share partnership statistics are dyadic dependent statistics. The dyadic dependent statistics tend to cause the model degeneracy problem.

CHAPTER III

THE MONTE CARLO METROPOLIS-HASTINGS ALGORITHM

A. Introduction

In this chapter, we propose a new algorithm, the so-called 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 $\kappa(\theta)/\kappa(\theta')$ 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 not available or very expensive.

The remainder of this chapter is organized as follows. In Section B, we describe the MCMH algorithm. In Section C, we test the MCMH algorithm on social network models. In Section D, we discuss the relation between MCMH and the group independence MH algorithm introduced by [8], and the potential applications of MCMH in marginal inference.

B. The Monte Carlo Metropolis-Hastings Algorithm

1. The Algorithm

Consider the problem of sampling from the distribution (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 a MCMC algorithm or an automated rejection sampling algorithm [13]. The MCMH algorithm works by iterating between the following steps:

Monte Carlo MH Algorithm I

1. Draw ϑ from some proposal distribution $Q(\theta_t, \vartheta)$.

2. 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 $g(y, \theta) = \exp\{-U(y, \theta)\}$, and $\boldsymbol{y}_t = (y_1^{(t)}, \ldots, y_m^{(t)})$ denotes the collection of the auxiliary samples.

3. Calculate the Monte Carlo MH ratio

$$\tilde{r}_m(\theta_t, \boldsymbol{y}_t, \vartheta) = \frac{1}{\widehat{R}_m(\theta_t, \boldsymbol{y}_t, \vartheta)} \frac{g(x, \vartheta)\pi(\vartheta)}{g(x, \theta_t)\pi(\theta_t)} \frac{Q(\vartheta, \theta_t)}{Q(\theta_t, \vartheta)},$$

   where $\pi(\theta)$ denotes the prior distribution imposed on θ.

4. 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.

5. If the proposal is rejected in step 4, 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 a MCMC algorithm or an automated rejection sampling algorithm.

Since the algorithm involves a Monte Carlo step to estimate the unknown normalizing constant ratio, it is termed as "Monte Carlo MH". Clearly, the samples $\{(\theta_t, \boldsymbol{y}_t)\}$ forms a Markov chain whose transition kernel is given by

$$\begin{aligned}\tilde{P}_m(\theta, \boldsymbol{y}; d\vartheta, d\boldsymbol{z}) &= \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta) Q(\theta, d\vartheta) f_\vartheta^m(d\boldsymbol{z}) \\ &\quad + \delta_{\theta,\boldsymbol{y}}(d\vartheta, d\boldsymbol{z})\big[1 - \int_{\Theta\times\mathbb{Y}} \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta') Q(\theta, d\vartheta') f_{\vartheta'}^m(d\boldsymbol{z}')\big] \\ &= \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta) Q(\theta, d\vartheta) f_\vartheta^m(d\boldsymbol{z}) \\ &\quad + \delta_{\theta,\boldsymbol{y}}(d\vartheta, d\boldsymbol{z})\big[1 - \int_{\Theta} \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta') Q(\theta, d\vartheta')\big],\end{aligned} \tag{3.1}$$

where $f_\theta^m(\boldsymbol{y}) = f(y_1, \ldots, y_m|\theta)$ denotes the joint density of $y_1, \ldots, y_m$.

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 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

$$\begin{aligned}\tilde{P}_m(\theta_t, d\vartheta) &= \int_{\mathbb{Y}}\int_{\mathbb{Y}} \tilde{P}_m(\theta_t, \boldsymbol{y}_t; d\vartheta, d\boldsymbol{z}) f_{\theta_t}^m(d\boldsymbol{y}_t) \\ &= \int_{\mathbb{Y}} \tilde{\alpha}(\theta_t, \boldsymbol{y}_t, \vartheta) Q(\theta_t, d\vartheta) f_{\theta_t}^m(d\boldsymbol{y}_t) \\ &\quad + \delta_{\theta_t}(d\vartheta)\big[1 - \int_{\Theta\times\mathbb{Y}} \tilde{\alpha}(\theta_t, \boldsymbol{y}_t, \vartheta') Q(\theta_t, d\vartheta') f_{\theta_t}^m(d\boldsymbol{y}_t)\big].\end{aligned} \tag{3.2}$$

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 non-adaptive Markov chain. However, in Theorem B.2 we still show that $\{\theta_t\}$ has the same stationary distribution as the time homogeneous Markov chain under the framework of adaptive Markov chain (see e.g., [66]). Note that the independence of $\tilde{P}_m(\theta_t, d\vartheta)$ on past states is not generally true for all marginal Markov chains. It is true for MCMH-I as for which $\boldsymbol{y}_t$ is generated from $f_{\theta_t}(\boldsymbol{y})$, which implies that $\boldsymbol{y}_t$ is independent of $\theta_0, \ldots, \theta_{t-1}$ conditional on θ_t.

The MCMH-I algorithm requires the auxiliary samples to be drawn at equilib-

rium, if a MCMC algorithm is used for generating the auxiliary samples. To ensure this requirement to be satisfied, we propose to choose the initial auxiliary sample at each iteration through an importance resampling procedure; that is, set $y_0^{(t+1)} = y_i^{(t)}$ with a probability proportional to the importance weight

$$w_i = g(y_i^{(t)}, \theta_{t+1})/g(y_i^{(t)}, \theta_t). \tag{3.3}$$

As long as $y_0^{(t+1)}$ follows correctly from the conditional distribution $f(y|\theta_{t+1})$, this procedure ensures that all samples, $\boldsymbol{y}_{t+1}$, $\boldsymbol{y}_{t+2}$, $\boldsymbol{y}_{t+3}$, ..., drawn in the followed iterations will follow correctly from the respective conditional distributions, provided that θ does not change drastically at each iteration. Note that the resampling procedure may introduce a (probably very slight) dependence on the previous samples. In practice, we may ignore this dependence, especially when m is large.

Regarding the choice of m, we note that m may not necessarily be very large in practice. In our experience, a value between 20 and 50 may be good for most problems. It seems that the random errors introduced by the Monte Carlo estimate of $\kappa(\theta_t)/\kappa(\vartheta)$ can be smoothed out by path averaging over iterations. This is particularly true for parameter estimation.

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 be described as follows:

Monte Carlo MH Algorithm II

1. Draw ϑ from some proposal distribution $Q(\theta_t, \vartheta)$.

2. Draw auxiliary samples $\boldsymbol{y}_t = (y_1^{(t)}, \ldots, y_m^{(t)})$ from $f(y|\theta_t)$ using a MCMC algorithm or an automated rejection algorithm.

3. Estimate the normalizing constant ratio $R(\theta_t, \vartheta) = \kappa(\vartheta)/\kappa(\theta_t)$ by

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

4. Calculate the Monte Carlo MH ratio

$$\tilde{r}_m(\theta_t, \boldsymbol{y}_t, \vartheta) = \frac{1}{\widehat{R}_m(\theta_t, \boldsymbol{y}_t, \vartheta)} \frac{g(x, \vartheta)\pi(\vartheta)}{g(x, \theta_t)\pi(\theta_t)} \frac{Q(\vartheta, \theta_t)}{Q(\theta_t, \vartheta)}.$$

5. 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

$$\begin{aligned} \tilde{P}_m(\theta, d\vartheta) = &\int_{\mathbb{Y}} \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta) Q(\theta, d\vartheta) f_\theta^m(d\boldsymbol{y}) \\ &+ \delta_\theta(d\vartheta)\big[1 - \int_{\Theta\times\mathbb{Y}} \tilde{\alpha}(\theta, \boldsymbol{y}, \vartheta') Q(\theta, d\vartheta') f_\theta^m(d\boldsymbol{y})\big], \end{aligned} \tag{3.4}$$

which is identical to the marginal transition kernel (3.2) except for notations. Hence, the two algorithms will have the same convergence rate for $\{\theta_t\}$.

Intuitively, one may expect that MCMH-I converges slowly than MCMH-II, as the former recycles the auxiliary samples when rejection occurs and thus the successive samples generated by it may have significantly higher correlation than those generated by the latter. 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

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

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

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

In addition to $f(y|\theta_t)$ or $f(y|\vartheta)$, the auxiliary samples can also 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 ([18], [78]) 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 [53] and path sampling [28], are also applicable to MCMH with an appropriate strategy for generating auxiliary samples.

2. Convergence

In this subsection, we first prove the ergodicity of MCMH-II; that is, showing

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

where k denotes the number of iterations, $\pi(\cdot|x)$ denotes the target distribution defined in (1.2), and $\|\cdot\|$ denotes the total variation norm as specified in [77]. Then, based on the theory of adaptive Markov chain [4], we show that MCMH-I has the same stationary distribution as MCMH-II. The main results are presented below, whose proofs can be found in Appendix A. Extension of these results to other variants of MCMH is straightforward.

Define

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

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

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

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

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

$$P(\theta, \vartheta) = \alpha(\theta, \vartheta) Q(\theta, \vartheta) + \delta_\theta(d\vartheta)\big[1 - \int_\Theta \alpha(\theta, \vartheta) Q(\theta, \vartheta) d\vartheta\big],$$

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 $\pi P = \pi$, and for any $\theta_0 \in \Theta$, $\lim_{k \to \infty} \|P^k(\theta_0, \cdot) - \pi(\cdot|x)\| = 0$.

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

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

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

The 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 [77]. The condition (A_2) assumes that the distributions $f(y|\theta)$ and $f(y|\vartheta)$ have a reasonable overlap such that $\widehat{R}$ forms a reasonable estimator of R. The 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 $m > M$,

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

It essentially requires that $\widehat{R}(\theta, \boldsymbol{y}, \vartheta)$ is a consistent estimator of $R(\theta, \boldsymbol{y}, \vartheta)$ and the stepsize of the proposal $Q(\theta, \vartheta)$ is reasonably small, i.e., ϑ lies in a small neighborhood of θ.

Lemma B.1 states that the marginal kernel $\tilde{P}_m$ has a stationary distribution. It is proved in a similar way to Theorem 1 of [3]. The relation between this work and [8] and [3] will be discussed in Section 5.

Lemma B.1 *Assume* (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 B.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 B.2 *Assume* (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* $m > M(\theta)$,

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

Theorem B.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 B.1 *Assume the 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* $k > K(\varepsilon, \theta_0, m)$

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

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

Theorem B.2 *Assume the 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.*

Theorem B.1 and Theorem B.2 imply, by standard MCMC theory (see, e.g., [77]), 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 B.1 for a fixed value of m.

C. An Example: Exponential Random Graph Models

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

$$\begin{aligned}
f(\boldsymbol{x}|\theta) &= \frac{1}{\kappa(\theta)} \exp\left\{\theta_1 e(x) + \theta_2 u(x|\tau)\right\}, \quad \text{(Model 1)}, \\
f(\boldsymbol{x}|\theta) &= \frac{1}{\kappa(\theta)} \exp\left\{\theta_1 e(x) + \theta_2 v(x|\tau)\right\}, \quad \text{(Model 2)}, \\
f(\boldsymbol{x}|\theta) &= \frac{1}{\kappa(\theta)} \exp\left\{\theta_1 e(x) + \theta_2 u(x|\tau) + \theta_3 v(x|\tau)\right\}, \quad \text{(Model 3)}.
\end{aligned}$$

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.

1. High School Student Friendship Network

The data was collected during the first wave (1994-1995) of 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 may choose not to nominate if they did not have enough number of close male or female friends. The detailed description of the data can be found in [62], [79], or at http://www.cpc.unc.edu/projects/addhealth. The full dataset contains 86 schools and 90,118 students. In this paper, only the subnetwork for school 10, which has 205 students, is analyzed. Also, only the undirected network for the case of mutual friendship are considered.

MCMH-I was applied to this network with $m = 20$. For each model, MCMH-I was run 5 times independently. Each run started with a random point and consisted of 5,000 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 were summarized in Table I.

For comparison, the MCMLE was also applied to this example. The software we used for MCMLE is an R package *ergm* by [41]. MCMLE was also run 5 times for each model of this example. Each run consisted of 25 iterations with 6,500 aux-

iliary networks being generated at each iteration. In the *ergm* package, the auxiliary networks was simulated using the tie-no-tie sampler with both the number of burnin and the number of interval steps being set to 20,000. Under this setting, a total of $1.3\times10^8(=20,000\times6,500)$ MH updates (each for one edge) are needed for generating 6500 networks at each iteration of MCMLE. The results are summarized in Table I. It indicates that MCMLE costs longer CPU times than MCMH-I for this example. All computations for this example were done on a 3.0GHz Intel Core 2 Duo computer.

Table I. Parameter estimation for the AddHealth school 10 network. The estimates were calculated by averaging over 5 independent runs with the standard deviations reported in the parentheses. CPU: the CPU time (in minutes) cost by a single run on a 3.0GHz Intel Core 2 Duo computer.

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

To assess accuracy of the MCMH estimates, the following procedure was proposed in a similar spirit to the parametric bootstrap method [26], 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 reversely estimated by simulated networks from the distribution $f(x|\widehat{\theta})$. The procedure consists of three steps:

(a) Given the estimate $\hat{\theta}$, simulate K networks, $x_1, \ldots, x_K$, independently using the Gibbs sampler.

(b) Calculate the statistics $S_a(x)$, $a \in A$ for each of the simulated networks.

(c) Calculate RMSE by following equation.

$$RMSE(S_a) = \sqrt{\sum_{i=1}^{K} \left[S_a(x_i) - S_a(x)\right]^2 / K}, \quad a \in A, \tag{3.5}$$

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

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

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

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

$$E_\theta[S_a(X)] = S_a(x), \quad \forall\ a \in A, \tag{3.6}$$

where $E_\theta[\cdot]$ denotes the expectation with respect to the distribution $f(x|\theta)$ given in (2.1). Hence, AMD also provides a measure for the quality of the estimate of θ.

For each of the estimates shown in Table I, the RMSEs and AMDs were calculated with $K = 1000$ and summarized in Table II. The results indicate that MCMH-I produced much more accurate estimates than MCMLE for all the three models. We note that [39] have also applied MCMLE to model 1 and model 2 for this network. Their estimate for model 2 is similar to ours, but their estimate for model 1 is much worse than ours. [39] 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.

Table II. 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

Finally, we assessed accuracy of the model estimates using the goodness-of-fit (GOF) plots [39]. 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, falls 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 1 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 other two models, GOF plots (omitted here) also indicate that MCMH-I works better than MCMLE for this example.

D. MCMH, GIMH and Marginal Inference

In the literature, there is one algorithm, namely, grouped independence MH (GIMH) [8], which is similar in spirit to the MCMH algorithm. GIMH is designed for marginal

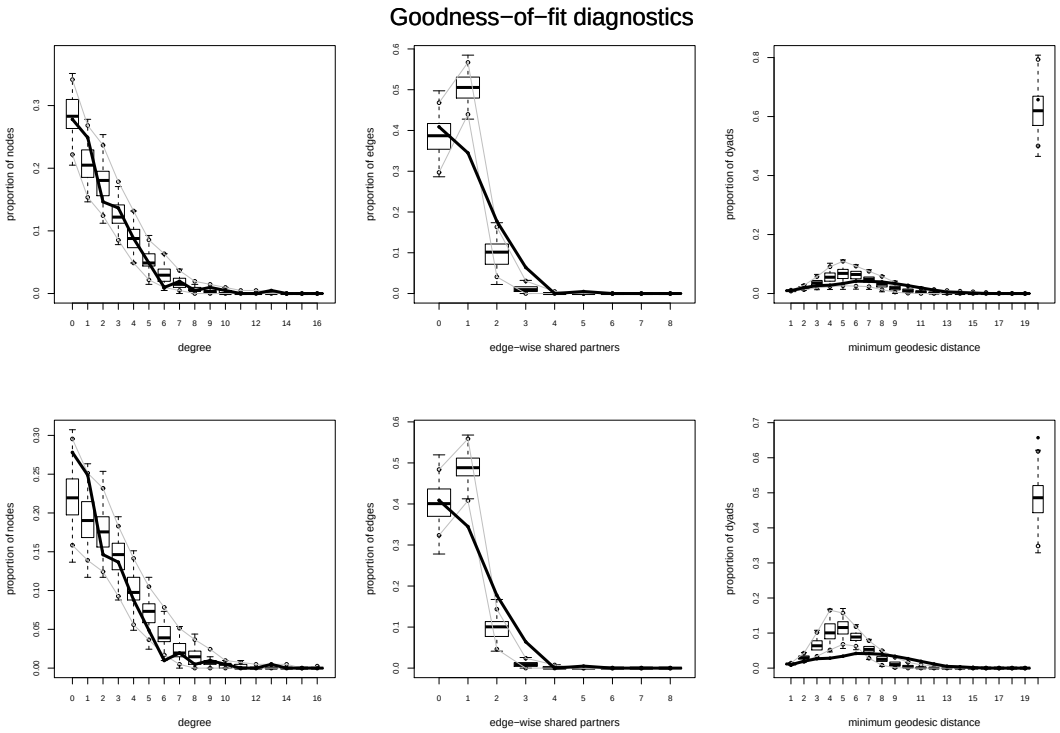

Fig. 1. Goodness-of-fit(GOF) plots for the high school student friendship network. Row 1: MCMH-I estimate; Row 2: MCMLE. The solid line shows the observed network statistics, and the box-plots represent the distributions of simulated network statistics.

inference from a joint distribution.

Let $\pi(\theta, y)$ denote a joint distribution. Suppose that one is interested in the marginal distribution $\pi(\theta)$. For example, 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 [11], a basic principle in Monte Carlo computation is to carry out analytical computation as much as possible. Motivated by this principle, [8] proposed to replace $\pi(\theta)$ by its Monte Carlo estimate in simulations when the analytical form of $\pi(\theta)$ is not available. Let $\boldsymbol{y} = (y_1, \ldots, y_m)$ denote a set of independently identically distributed (iid) samples drawn from a trial distribution $q_\theta(y)$. It follows from the standard theory of importance sampling that

$$\widetilde{\pi}(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{\pi(\theta, y_i)}{q_\theta(y_i)}, \tag{3.7}$$

forms an unbiased estimate of $\pi(\theta)$. In simulations, GIMH treats $\widetilde{\pi}(\theta)$ as a known target density, then simulate from it using the Metropolis-Hastings 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 iid auxiliary samples drawn from $q_\theta(y)$. One iteration of GIMH consists of the following steps:

Group Independence MH Algorithm

- Generate a new candidate point θ' from a proposal distribution $T(\theta'|\theta_t)$.
- Draw m iid samples $\boldsymbol{y}' = (y'_1, \ldots, y'_m)$ from the trial distribution $q_{\theta'}(y)$.
- Accept the proposal with probability

$$\min\left\{1, \frac{\widetilde{\pi}(\theta')}{\widetilde{\pi}(\theta_t)} \frac{T(\theta_t|\theta')}{T(\theta'|\theta_t)}\right\}.$$

  If it is accepted, set $\theta_{t+1} = \theta'$ 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 [3] under similar conditions to those assumed for MCMH in this paper. In the context of marginal inference, MCMH-I can be described as follows.

MCMH-I algorithm (for marginal Inference)

- Generate a new candidate point θ' from a proposal distribution $T(\theta'|\theta_t)$.
- Accept the proposal with probability

$$\min\left\{1, \widetilde{R}(\theta_t, \theta')\frac{T(\theta_t|\theta')}{T(\theta'|\theta_t)}\right\},$$

  where $\widetilde{R}(\theta_t, \theta') = \frac{1}{m}\sum_{i=1}^{m} \pi(\theta', y_i^{(t)})/\pi(\theta_t, y_i^{(t)})$ forms an unbiased estimate of the marginal density ratio $R(\theta_t, \theta') = \int \pi(\theta', y)dy / \int \pi(\theta_t, y)dy$. If it is accepted, set $\theta_{t+1} = \theta'$; 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 $\pi(y|\theta_{t+1})$. The auxiliary samples $y_1^{(t+1)}, \ldots, y_m^{(t+1)}$ can be generated via a MCMC simulation.

Taking a closer look at MCMH-I, we can find that it is designed in a different rule from GIMH. Firstly, one estimates the marginal distributions in GIMH; whereas, one directly estimates the ratio of marginal distributions in MCMH-I. This leads to an important use of MCMH for simulating from distributions with intractable normalizing constants, which is the focus of this paper. Note that GIMH cannot be directly used to this problem. Secondly, GIMH requires to draw samples in iterations from two distributions $q_\theta(\cdot)$ and $q_{\theta'}(\cdot)$, while MCMH-I requires only to draw samples from a single distribution $\pi(\cdot|\theta)$. 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. From the theoretical perspective, we analyze the convergence of the marginal chain resulted from MCMH-I under the framework of adaptive Markov chains, while GIMH is analyzed in [3] under the framework of time homogeneous Markov chains.

MCMH can potentially be applied to many statistical models for which marginal inference is our main interest, such as generalized linear mixed models (see, e.g., [52]) and hidden Markov random field models [70]. MCMH can also be applied to Bayesian analysis for the missing data problems that are traditionally treated with the EM algorithm [24] or the Monte Carlo EM algorithm [81]. 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 [31] to help it escape from suboptimal solutions.

CHAPTER IV

BAYESIAN STOCHASTIC APPROXIMATION MONTE CARLO ALGORITHM

A. Introduction

In this chapter, we propose a new algorithm, the so-called Bayesian Stochastic Approximation Monte Carlo (BSAMC) algorithm, for tackling the intractable normalizing constant problem. BSAMC works by simulating from a sequence of approximated distributions, which are denoted by $\pi_t(\theta|\boldsymbol{z})$ and obtained using the stochastic approximation Monte Carlo (SAMC) algorithm [50]. Let θ_t denote a sample simulated from $\pi_t(\theta|\boldsymbol{z})$. Under mild conditions, we show that for any bounded measurable function $\varphi(\theta)$, $\sum_{t=1}^{n} \varphi(\theta_t)/n$ converges almost surely to the posterior mean of $\varphi(\theta)$ as n goes to infinity. One significant advantage of BSAMC over the auxiliary variable MCMC methods is that it avoids the requirement for perfect samples, and thus can be applied to many models for which the auxiliary variable MCMC methods are not applicable. BSAMC is general; it can be applied to any models whose normalizing constant is intractable. Comparing to Monte Carlo MLE, BSAMC is very robust to the choice of θ_0 due to the powerful ability of SAMC in sample space exploration. Finally, we note that although BSAMC works based on SAMC, SAMC itself cannot be directly applied to sample from the posterior $\pi(\theta|\boldsymbol{z})$. Hence, BSAMC represents an extension of SAMC for Bayesian analysis for the models with intractable normalizing constants. BSAMC also provides a general framework for approximated Bayesian analysis through simulating from a sequence of approximated distributions with their average converging to the target posterior distribution.

The remainder of this chapter is organized as follows. In Section B, we describe the BSAMC algorithm and explore its theoretical property. In Section C, we apply BSAMC to Ising models along with a comparison with the MCMLE algorithm. The numerical results show that BSAMC can perform very robustly to the initial guess of θ. In Section D, we apply BSAMC algorithm to autologistic and autonormal models.

B. Bayesian Stochastic Approximation Monte Carlo Algorithm

1. The BSAMC Algorithm

To approximate the normalizing constant $\kappa(\theta)$, the MCMLE method proposed by [32] adopts an importance sampling method with the trial distribution $f(x|\theta_0)$. It is obvious that, when θ_0 is far from the true value of θ, $f(x|\theta_0)$ may approximate $f(x|\theta)$ poorly, and the resulting estimate of $\kappa(\theta)$ may be biased. To resolve this difficulty, we choose the following mixture distribution as the trial distribution:

$$g(x, \theta_0) = \frac{1}{k} \sum_{i=1}^{k} \frac{p(x, \theta_0)}{\xi^{(i)}} I(x \in E_i), \tag{4.1}$$

where $E_1, \ldots, E_k$ forms a partition of the sample space $\mathcal{X}$, and $\xi^{(i)} = \int_{E_i} p(x, \theta_0) dx$. Let $\xi = (\xi^{(1)}, \ldots, \xi^{(k)})$. Without loss of generality, we assume that the sample space has been partitioned according to the energy function $-\log p(x, \theta_0)$ as follows: $E_1 = \{x : -\log p(x, \theta_0) < h_1\}$, $E_2 = \{x : h_1 \leq -\log p(x, \theta_0) < h_2\}$, ..., $E_k = \{x : -\log p(x, \theta_0) \geq h_{k-1}\}$, where $h_1 < h_2 < \ldots < h_{k-1}$ are some pre-fixed numbers. It is easy to see that sampling of $g(x, \theta_0)$ will lead to an equal sampling from each of the subregions $E_1, \ldots, E_k$, and the normalizing constant $\kappa(\theta)$ can thus be well approximated even when θ_0 is far from the true value of θ. Clearly, the success of the approximation amounts on the estimation of the quantities $\xi^{(1)}, \ldots, \xi^{(k)}$ which are unknown *a priori*. Thanks to the SAMC algorithm, it provides consistent estimates

of these quantities in an iterative way. Let $\xi_t^{(i)}$ denote the estimate of $\xi^{(i)}$ at iteration t, let $\xi_t = (\xi_t^{(1)}, \ldots, \xi_t^{(k)})$, and let $x_t^{(1)}, \ldots, x_t^{(m)}$ denote the samples simulated from the working trial distribution

$$g_{\xi_t}(x, \theta_0) = \frac{1}{Z_t} \sum_{i=1}^{k} \frac{p(x, \theta_0)}{\xi_t^{(i)}} I(x \in E_i), \tag{4.2}$$

where Z_t is the normalizing constant of $g_{\xi_t}(x, \theta_0)$. Then $\log \pi(\theta|\boldsymbol{z})$ can be approximated by

$$\log \pi_{\xi_t}(\theta|\boldsymbol{z}) = \log \pi(\theta) + \log p(\boldsymbol{z}, \theta) - \log(Z_t) - \log \left(\frac{1}{m} \sum_{i=1}^{m} p(x_t^{(i)}, \theta) / g_{\xi_t}(x_t^{(i)}, \theta_0) \right). \tag{4.3}$$

It is clear that as $m \to \infty$, $\log \pi_{\xi_t}(\theta|\boldsymbol{z})$ approaches to $\log \pi(\theta|\boldsymbol{z})$.

BSAMC Algorithm

(a) (*Auxiliary sample generating*) Simulate samples $x_t^{(1)}, \ldots, x_t^{(m)}$ from the working trial distribution $g_{\xi_{t-1}}(x, \theta_0)$ using the MH algorithm. Denote the set of auxiliary samples by $\boldsymbol{x}_t = (x_t^{(1)}, \ldots, x_t^{(m)})$.

(b) (*Estimate updating*) Update the estimates ξ_{t-1} by setting

$$\log(\xi_t) = \log(\xi_{t-1}) + \gamma_t H_{\xi_{t-1}}(\boldsymbol{x}_t), \tag{4.4}$$

where $H_{\xi_{t-1}}(\boldsymbol{x}_t)$ is a k-vector with the i-th component given by $\sum_{j=1}^{m} I(x_t^{(j)} \in E_i)/m - 1/k$, $I(\cdot)$ is the indicator function, and $\{\gamma_t\}$ is a pre-specified gain factor sequence. How to choose the sequence $\{\gamma_t\}$ will be discussed later.

(c) (*Posterior sample generating*) Draw sample $\theta_t^{(1)}, \ldots, \theta_t^{(s)}$ from the approximated posterior $\pi_{\xi_t}(\theta|\boldsymbol{z})$ (as specified in (4.3)) using the MH algorithm.

Let $(\theta_1^{(1)}, \ldots, \theta_1^{(s)})$, ..., $(\theta_n^{(1)}, \ldots, \theta_n^{(s)})$ denote the samples of θ generated in n iterations of BSAMC. Then, for any bounded measurable function $\varphi(\theta)$, its posterior

mean $\pi(\varphi) = \int \varphi(\theta)\pi(\theta|\boldsymbol{z})d\theta$ can be estimated by

$$\widehat{\pi_n(\varphi)} = \frac{1}{(n-n_0)s} \sum_{t=n_0+1}^{n} \sum_{i=1}^{s} \varphi(\theta_t^{(i)}), \tag{4.5}$$

where n_0 denotes the burn-in time of the simulation. In Section 2, we show that under mild conditions, $\widehat{\pi_n(\varphi)}$ converges almost surely to $\pi(\varphi)$ when both n and m become large.

The merit of this algorithm is the use of SAMC for learning of the trial distribution $g(x, \theta_0)$. As discussed in [50], SAMC possesses a self-adjusting mechanism: If a component $\xi^{(i)}$ is underestimated (overestimated) in the current iteration, then the subregion E_i will tend to be oversampled (undersampled) in the next iteration and the current estimate of $\xi^{(i)}$ will thus be counter-adjusted by the quantity $\gamma_t(e_t^{(i)} - 1/k)$ as prescribed in step (b). This mechanism enables the simulation to converge very quickly with samples being drawn equally from different subregions of the sample space even when θ_0 is far from the true value of θ. In general, the performance of BSAMC can be very robust to the choice of θ_0.

For an effective implementation of BSAMC, several issues need to be considered:

- *Choice of θ_0:* Like the MCMLE method, θ_0 can be chosen using another estimator of θ, which is easy to calculate, such as the maximum pseudo likelihood estimator (MPLE) [9] or the double MH estimator [48]. In our study, we set θ_0 to the MPLE of θ.

- *Sample space partition:* As discussed in [50], the sample space should be partitioned such that the simulation conducted in step (a) should have a reasonable acceptance rate. Since, within the same subregion, the SAMC simulation of $g_{\xi_t}(x, \theta_0)$ is reduced to the MH simulation of $f(x|\theta_0)$, the cutting values $h_1, \ldots, h_{k-1}$ are required to satisfy the constraint $\max_i(h_i - h_{i-1}) \leq 2$. In prac-

tice, we often set the subregions to have an equal energy bandwidth; that is, setting $h_i = h_{i-1} + \delta$ with δ taking a value between 0 and 2. The value of h_1 can be chosen to be reasonably small, and that of h_{k-1} can be reasonably large, such that both the energy regions of the true distribution $f(x|\theta)$ and of the initial distribution $f(x|\theta_0)$ can be covered.

- *Choice of m:* In practice, BSAMC is often run in two stages, although, in theory, this is not necessary. In stage I, a small value of m is often used. The goal of this stage is to approximate ξ_i's, so step (c) can be omitted in this stage. In stage II, a large value of m is often used. The goal of this stage is draw samples of θ. This two-stage implementation strategy often improves the efficiency of the algorithm. In terms of MCMC simulations, stage I corresponds to the burn-in steps in (4.5).

- *Choice of $\{\gamma_t\}$:* As shown in [47], to ensure the convergence of ξ_t, $\{\gamma_t\}$ should be chosen as a positive, nondecreasing sequence satisfying the conditions

$$\text{(a) } \bar{\lim}_{t\to\infty}|\gamma_t^{-1} - \gamma_{t+1}^{-1}| < \infty, \quad \text{(b) } \sum_{t=1}^{\infty}\gamma_t = \infty, \quad \text{and} \quad \text{(c) } \sum_{t=1}^{\infty}\gamma_t^{\eta} < \infty, \tag{4.6}$$

for some $\eta > 1$. In this paper, we choose

$$\gamma_t = \frac{t_0}{\max(t_0, t)}, \quad t = 1, 2, \cdots \tag{4.7}$$

for some specified value $t_0 > 1$. As discussed in [50], a large value of t_0 will force the sampler to reach all subregions quickly, even in the presence of multiple local energy minima. Therefore, t_0 should be set to a large value for a complex problem. In practice, the choice of t_0 should be associated with the choice of N, the total number of iterations of the run. The appropriateness of their choices can be diagnosed by checking the convergence of multiple runs (starting with

different points) through an examination for the variation of $\hat{\xi}$ or $\hat{\boldsymbol{f}}$, where $\hat{\xi}$ and $\hat{\boldsymbol{f}}$ denote, respectively, the estimate of ξ and the sampling frequencies of the subregions obtained at the end of a run. A rough examination for $\hat{\xi}$ is to see visually whether $\hat{\xi}$'s produced in multiple runs follow the same pattern. Existence of different patterns implies that the gain factor is still large at the end of the runs or some parts of the sample space are not yet visited in all runs. This is similar to $\hat{\boldsymbol{f}}$. If the choices of t_0 and N are appropriate, each nonempty subregion should be sampled roughly equally at end of each run. If the runs are diagnosed as non-converged, BSAMC should be re-run with a large value of N, a larger value of t_0, or both.

2. Convergence

For the reason of mathematical simplicity, we assume that Ξ, the parameter space of ξ, is compact. Therefore, the sequence $\{\xi_t\}$ can be kept in a compact set. Extension of our results to the case that $\Xi = \mathbb{R}^k$ is trivial with the technique of varying truncations studied in [2] and [17], which ensures, almost surely, that the sequence $\{\xi_t\}$ can be kept in a compact set.

To establish the convergence of the BSAMC estimator (4.5), we first prove Theorem B.1, which concerns the convergence of ξ_t and the convergence of the sample average of $\rho(\boldsymbol{x}_t)$, where ρ denotes a bounded measurable function. Note that Theorem B.1 concerns only steps (a) and (b) of BSAMC. If step (c) is ignored, BSAMC is reduced to the multiple SAMC algorithm studied in [47], where "multiple" means that multiple samples are allowed to be generated from the working density $g_{\xi_t}(x|\theta_0)$ at each iteration. Including step (c) enables BSAMC to be used for Bayesian inference for the models with intractable normalizing constants, and this is also the main methodology contribution of this paper. Rigorous theory has been established

for the methodology development. BSAMC also provides a general framework for approximated Bayesian analysis through sampling from a sequence of approximated distributions with their averages converging to the target posterior distribution.

With a slight abuse of notations, we let $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(m)})$ denote m MCMC samples drawn from $g(x|\theta_0)$, and let $g(\boldsymbol{x}|\theta_0)$ denote the joint density/mass function of $\boldsymbol{x}$. Then Theorem B.1 can be stated as follows:

Theorem B.1 *Consider the BSAMC algorithm. If the condition (4.6) and the drift condition (given in Appendix) hold and Ξ is compact, then for any integer $m \geq 1$,*

(i)

$$\xi_t^{(i)} \to c\xi_i, \quad a.s. \quad as\ t \to \infty, \tag{4.8}$$

where c is an arbitrary positive constant and it can be determined by imposing a constraint on $\xi_t^{(i)}$'s, e.g., $\sum_{i=1}^k \xi_t^{(i)}$ is equal to a fixed constant.

(ii) For any bounded measurable function $\rho(\cdot)$,

$$\frac{1}{n}\sum_{t=1}^{n} \rho(\boldsymbol{x}_t) \to \int_{\mathcal{X}} \rho(\boldsymbol{x}) g(\boldsymbol{x}|\theta_0) d\boldsymbol{x}, \quad a.s. \quad as\ n \to \infty.$$

PROOF: The proof of part (i) can be found in [47]. As aforementioned, ignoring step (c), BSAMC is reduced to the multiple SAMC algorithm described in [47].

To prove part (ii), we first consider a general measurable function $W_\xi(\boldsymbol{y})$ (possibly depending on ξ) and the Poisson equation:

$$u_\xi(\boldsymbol{y}) - P_\xi u_\xi(\boldsymbol{y}) = W_\xi(\boldsymbol{y}) - w_\xi,$$

where P_ξ denotes the joint Markov transition kernel as defined in Appendix, $\boldsymbol{y}$ denotes a sample generated by P_ξ, $P_\xi u_\xi(\boldsymbol{y}) = \int_{\mathcal{X}^m} u_\xi(\boldsymbol{y}') P_\xi(\boldsymbol{y}, \boldsymbol{y}') d\boldsymbol{y}'$, $w_\xi = \int W_\xi(\boldsymbol{y}) f_\xi(\boldsymbol{y}) d\boldsymbol{y}$, and $f_\xi(\boldsymbol{y})$ denotes the stationary distribution of P_ξ. It follows from [2] (Proposition

6.1) that if $W_\xi(\boldsymbol{y})$ is bounded, then there exists a constant C such that for any $\xi, \xi' \in \Xi$,

$$\begin{aligned} &\sup_{\boldsymbol{y}\in\mathcal{X}^m} [\|u_\xi(\boldsymbol{y})\| + \|P_\xi u_\xi(\boldsymbol{y})\|] < C, \\ &\sup_{\boldsymbol{y}\in\mathcal{X}^m} [\|u_\xi(\boldsymbol{y}) - u_{\xi'}(\boldsymbol{y})\| + \|P_\xi u_\xi(\boldsymbol{y}) - P_{\xi'} u_{\xi'}(\boldsymbol{y})\|] < C\|\xi - \xi'\|. \end{aligned} \tag{4.9}$$

Let $\epsilon_0 = \epsilon_0' = 0$, and

$$\begin{aligned} \epsilon_t &= \gamma_t[u_{\xi_{t-1}}(\boldsymbol{x}_t) - P_{\xi_{t-1}} u_{\xi_{t-1}}(\boldsymbol{x}_{t-1})], \\ \epsilon_t' &= \gamma_t[P_{\xi_t} u_{\xi_t}(\boldsymbol{x}_t) - P_{\xi_{t-1}} u_{\xi_{t-1}}(\boldsymbol{x}_t)] + (\gamma_{t+1} - \gamma_t) P_{\xi_t} u_{\xi_t}(\boldsymbol{x}_t), \\ \epsilon_t'' &= -\gamma_{t+1} P_{\xi_t} u_{\xi_t}(\boldsymbol{x}_t). \end{aligned}$$

With the Poisson equation, it is easy to verify that

$$\begin{aligned} \gamma_t[W_{\xi_{t-1}}(\boldsymbol{x}_t) - w_{\xi_{t-1}}] &= \epsilon_t + \epsilon_t' + \epsilon_t'' - \epsilon_{t-1}'', \\ \sum_{t=1}^n \gamma_t[W_{\xi_{t-1}}(\boldsymbol{x}_t) - w_{\xi_{t-1}}] &= \sum_{t=1}^n \epsilon_t + \sum_{t=1}^n \epsilon_t' + \epsilon_n'' - \epsilon_0''. \end{aligned}$$

It follows from (4.6) and (4.9) that $\sum_{t=1}^\infty \|\epsilon_t\|^2 < \infty$. Similarly, there exists a constant C' such that

$$\sum_{t=1}^\infty \|\epsilon_t'\| \le C' + C\sum_{t=1}^\infty \gamma_t \|\xi_t - \xi_{t-1}\| = C' + C\sum_{t=1}^\infty \gamma_t\gamma_{t-1}\|H_{\xi_{t-1}}(\boldsymbol{x}_{t-1})\| < \infty,$$

where the last inequality follows from (4.6) and the boundedness of $H_\xi(\boldsymbol{x})$. For BSAMC, we have $\|H_\xi(\boldsymbol{x})\| \le 1$.

Let $\mathcal{F}_t = \{\xi_0, \boldsymbol{x}_0; \xi_1, \boldsymbol{x}_1; \ldots, \xi_t, \boldsymbol{x}_t\}$ denote a filtration. Then $\{\epsilon_t\}$ forms a martingale difference sequence adapted to $\{\mathcal{F}_t\}_{t\ge 0}$. Since $\sum_{t=1}^\infty \|\epsilon_t\|^2 < \infty$, by the martingale convergence theorem, $\sum_{t=1}^n \epsilon_t$ converges almost surely. Then, following from (4.9) and

the convergence of $\sum_{t=1}^{\infty} \|\epsilon_t'\|$, we have

$$\sum_{t=1}^{n} \gamma_t [W_{\xi_{t-1}}(\boldsymbol{x}_t) - w_{\xi_{t-1}}] < \infty, \quad a.s. \tag{4.10}$$

Applying Kronecker's Lemma to (4.10) with $\gamma_t = 1/t$ and $W_{\xi_{t-1}}(\boldsymbol{x}_t) = \rho(\boldsymbol{x}_t)$, we obtain

$$\frac{1}{n} \sum_{t=1}^{n} \left[\rho(\boldsymbol{x}_t) - \int_{\mathcal{X}} \rho(\boldsymbol{x}) g_{\xi_t}(\boldsymbol{x}|\theta_0) d\boldsymbol{x} \right] \to 0, \quad a.s. \tag{4.11}$$

By the convergence of ξ_t established in part (i), which implies that $\boldsymbol{x}_t$ will converge in distribution to a random variable distributed according to $g(\boldsymbol{x}|\theta_0)$, and the boundedness of $\rho(\boldsymbol{x})$, we have

$$\int_{\mathcal{X}} \rho(\boldsymbol{x}) g_{\xi_t}(\boldsymbol{x}|\theta_0) d\boldsymbol{x} \to \int_{\mathcal{X}} \rho(\boldsymbol{x}) g(\boldsymbol{x}|\theta_0) d\boldsymbol{x}, \quad \text{as } t \to \infty,$$

which, together with (4.11), concludes the proof of part (ii). □

The drift condition is classical in the literature of Markov chain. It implies the existence of a stationary distribution and uniform ergodicity of the Markov chain. However, it is usually difficult to verify. For example, for the random walk MH kernel, complicated conditions are needed to control the tail behavior of the target distribution [2]. For mathematical simplicity, one may assume that the sample space $\mathcal{X}$ is compact. For example, one may restrict $\mathcal{X}$ to the set $\{x : f(x|\theta_0) \geq \epsilon_0\}$ for a sufficiently small number ϵ_0. In addition, one may assume that the proposal distribution $q(\cdot, \cdot)$ satisfies the local positive condition:

For every $x \in \mathcal{X}$, there exists $\epsilon_1 > 0$ and $\epsilon_2 > 0$ such that

$$\|x - y\| \leq \epsilon_1 \Longrightarrow q(x, y) \geq \epsilon_2,$$

where $\|x - y\|$ denotes a certain distance measure between x and y.

When the sample space $\mathcal{X}$ is compact and the proposal distribution satisfies the

local positive condition, it is easy to verify that BSAMC satisfies the drift condition. Refer to [47] for the details, where the drift condition is verified for multiple SAMC.

The next theorem concerns the convergence of the log-posterior distribution $\log \pi_{\xi_t}(\theta|\boldsymbol{z})$.

Theorem B.2 *Consider the BSAMC algorithm. If the parameter space Θ is compact and the conditions of Theorem B.1 hold, then for any $\theta \in \Theta$,*

$$\frac{1}{n}\sum_{t=1}^{n} \log \pi_{\xi_t}(\theta|\boldsymbol{z}) \to \log \pi(\theta|\boldsymbol{z}), \quad a.s.$$

as $m \to \infty$ and $n \to \infty$.

PROOF: Let $R_t(\theta) = \kappa(\theta)/Z_t$, where Z_t denotes the normalizing constant of $g_{\xi_t}(x|\theta_0)$. To define notations, we rewrite $g_{\xi_t}(x|\theta_0)$ as

$$g_{\xi_t}(x|\theta_0) = \frac{1}{Z_t}\widehat{\psi}_t(x) = \frac{1}{Z_t}\sum_{i=1}^{k} \frac{p(x,\theta_0)}{\xi_t^{(i)}} I(x \in E_i),$$

and let

$$\widehat{R}_t(\theta) = \frac{1}{m}\sum_{j=1}^{m} \frac{p(x_t^{(j)},\theta)}{\widehat{\psi}_t(x_t^{(j)})}.$$

It is easy to show that for any θ, $\widehat{R}_t(\theta)$ forms an unbiased and consistent estimator of $R_t(\theta)$ [20]. Following the standard theory of Markov chain Monte Carlo (see, e.g., [77]), we have

$$\sqrt{m}(\widehat{R}_t(\theta) - R_t(\theta)) \to N(0, \sigma_t^2(\theta)),$$

for some positive constant $\sigma_t(\theta)$, depending on θ and t, as $m \to \infty$. See [65] for an explicit form of $\sigma_t(\theta)$. By the Delta method,

$$\sqrt{m}\left(\log\left(\widehat{R}_t(\theta)\right) - \log(R_t(\theta))\right) \to N\big(0, \sigma_t^2(\theta)/R_t^2(\theta)\big). \tag{4.12}$$

By part (i) of Theorem B.1, we have, as $t \to \infty$,

$$R_t(\theta) \to R(\theta), \quad \text{and} \quad \sigma_t^2(\theta) \to \sigma^2(\theta), \quad a.s., \tag{4.13}$$

where $R(\theta) = \kappa(\theta)/Z$ and $Z = 1/k$ denotes the normalizing constant of the distribution $g(x|\theta_0)$.

With $\widehat{R}_t(\theta)$, we have

$$\log \pi_{\xi_t}(\theta|\boldsymbol{z}) = \log \pi(\theta) + \log p(\boldsymbol{z}, \theta) - \log(Z_t) - \log(\widehat{R}_t(\theta)).$$

By (4.12), (4.13) and part (ii) of Theorem B.1, we have

$$\frac{1}{n}\sum_{t=1}^{n} \log \pi_{\xi_t}(\theta|\boldsymbol{z}) \to \log \pi(\theta|\boldsymbol{z}), \quad a.s.$$

for any $\theta \in \Theta$ as $m \to \infty$ and $n \to \infty$. This completes the proof of the theorem. □

Theorem B.3 concerns the convergence of the BSAMC estimator (4.5). For simplicity, we considered only the case $s = 1$. Extension to the case $s > 1$ is trivial. To prove this theorem, we first introduce a definition of *strongly residually Cesàro α-integrable* and a lemma of strong law of large numbers, which are both taken from [16].

Definition B.1 *A sequence of random variables, $\{X_n, n \geq 1\}$, is said to be strongly residually Cesàro α-integrable (SRCI(α), in short) if there exists an $\alpha \in (0, \infty)$ such that the following two conditions hold:*

$$(i)\ \sup_{n\geq 1} \frac{1}{n}\sum_{i=1}^{n} E[|X_i|] < \infty, \quad \textit{and}, \quad (ii)\ \sum_{n=1}^{\infty} \frac{1}{n} E[(|X_i| - n^{\alpha})I(|X_i| > n^{\alpha})] < \infty. \tag{4.14}$$

Lemma B.1 *Let $\{X_n\}$ be a ϕ-mixing sequence of random variables and suppose that there exist constants C and γ with $0 < \gamma < 1$, such that, $\phi_n \leq C\gamma^n$ $\forall n$. If the*

sequence $\{X_n\}$ *satisfies the condition* $SRCI(\alpha)$ *for some* $\alpha \in (0,1)$*, then*

$$\frac{1}{n}(S_n - E[S_n]) \to 0, \quad a.s. \tag{4.15}$$

as $n \to \infty$*, where* $S_n = \sum_{i=1}^{n} X_i$*.*

Theorem B.3 *Consider the BSAMC algorithm. Assume that the parameter space* Θ *is compact and the conditions of Theorem B.1 hold. Let* θ_t *denote a sample drawn from* $\pi_{\xi_t}(\theta|\boldsymbol{z})$ *in the BSAMC algorithm. Then, for any bounded measurable function* φ*,*

$$\frac{1}{n}\sum_{t=1}^{n} \varphi(\theta_t) \to \pi(\varphi), \quad a.s., \tag{4.16}$$

as $m \to \infty$ *and* $n \to \infty$*, where* $\pi(\varphi) = \int_\Theta \varphi(\theta)\pi(\theta|\boldsymbol{z})d\theta$ *denotes the posterior mean of* $\varphi(\theta)$*.*

PROOF: To show this theorem, we first show that $\{\varphi(\theta_t)\}$ is strongly residually Cesàro α-integrable for some $\alpha \in (0,1)$, say $\alpha = 1/2$. This is obvious, as $\varphi(\theta)$ is bounded.

From the BSAMC algorithm, it is easy to see that $\{\theta_t\}$ forms a Markovian sequence. Since Θ is compact, the Markov chain induced by the MH algorithm for simulating from $\pi_{\xi_t}(\theta|\boldsymbol{z})$ is uniformly ergodic, and thus $\{\theta_t\}$ forms a ϕ-mixing sequence and there exist constants C and η with $0 < \eta < 1$ such that $\phi_n \leq C\eta^n$ holds, where C and η are given by

$$C = \sup_{t \geq 1} C_t, \quad \eta = \sup_{t \geq 1} \eta_t,$$

where C_t and η_t are determined by the MH algorithm for simulating from $\pi_{\xi_t}(\theta|\boldsymbol{z})$. Since Ξ is compact, $0 < \eta < 1$ holds. Then, it follows from Lemma B.1 that

$$\frac{1}{n}\left[\sum_{t=1}^{n} \varphi(\theta_t) - \sum_{t=1}^{n} E\varphi(\theta_t)\right] \to 0, \quad a.s., \quad \text{as } n \to \infty. \tag{4.17}$$

Now we consider the quantity $\sum_{t=1}^{n} E\varphi(\theta_t)/n$. Let

$$\epsilon_t(\theta) = \log \pi_{\xi_t}(\theta|\boldsymbol{z}) - \log \pi(\theta|\boldsymbol{z}).$$

A direct calculation yields

$$\begin{aligned}
\frac{1}{n}\sum_{t=1}^{n} E\varphi(\theta_t) &= \frac{1}{n}\sum_{t=1}^{n} \int \varphi(\theta)\pi_{\xi_t}(\theta|\boldsymbol{z})d\theta \\
&= \frac{1}{n}\sum_{t=1}^{n} \int \varphi(\theta) \exp\left\{\log \pi(\theta|\boldsymbol{z}) + \epsilon_t(\theta)\right\} d\theta \\
&= \frac{1}{n}\sum_{t=1}^{n} \int \varphi(\theta)\pi(\theta|\boldsymbol{z}) \left[1 + \epsilon_t(\theta) + O(\epsilon_t^2(\theta)\right] d\theta \\
&= \int \varphi(\theta)\pi(\theta|\boldsymbol{z}) + \int \varphi(\theta)\pi(\theta|\boldsymbol{z}) \left[\frac{1}{n}\sum_{t=1}^{n} \epsilon_t(\theta)\right] d\theta \\
&\quad + \int \varphi(\theta)\pi(\theta|\boldsymbol{z}) \left[\frac{1}{n}\sum_{t=1}^{n} O(\epsilon_t^2(\theta))\right] d\theta \\
&= \pi(\varphi) + (I) + (II).
\end{aligned} \tag{4.18}$$

Theorem B.2 implies that $\sum_{t=1}^{n} \epsilon_t(\theta)/n \to 0$ almost surely as $n \to \infty$ and $m \to \infty$. Since $\varphi(\theta)$ is bounded and Θ is compact, the integrand $\varphi(\theta)\pi(\theta|\boldsymbol{z}) \left[\frac{1}{n}\sum_{t=1}^{n} \epsilon_t(\theta)\right]$ is uniformly bounded on Θ. It follows from the bounded convergence theorem ([12], p.214) that $(I) \to 0$ as $m \to \infty$ and $n \to \infty$.

Since Θ is compact, it follows from (4.17) (with $\varphi(\theta) = \epsilon^2(\theta)$), (4.12), (4.13) that

$$\frac{1}{n}\sum_{t=1}^{n} O(\epsilon_t^2(\theta)) \to O(\frac{1}{m}), \quad \text{as } n \to \infty,$$

which implies that $(II) \to 0$ as $m \to \infty$ and $n \to \infty$. We conclude the proof by summarizing (4.17) and (4.18). □

In Theorems B.2 and B.3, Θ is restricted to a compact set. Since Θ can be set to a huge set, say, $[-10^{100}, 10^{100}]^{\dim(\theta)}$, which, as a practical matter, is equivalent to setting $\Theta = \mathbb{R}^{\dim(\theta)}$. Hence, the compactness assumption of Θ does not significantly affect

applications of the BSAMC algorithm. From the theoretical perspective, relaxing Θ to $\mathbb{R}^{\dim(\theta)}$ is of great interest. However, this may require some extra theory on the law of large numbers of adaptive MCMC, as BSAMC falls into the class of adaptive MCMC algorithms whose target distributions $g_{\xi_t}(x|\theta_0)$ (for auxiliary samples $x_t^{(1)}, \ldots, x_t^{(m)}$) and $\pi_{\xi_t}(\theta|\boldsymbol{z})$ (for samples $\theta_t^{(1)}, \ldots, \theta_t^{(s)}$) change from iteration to iteration.

C. The Ising Model

In this section, we illustrate the use of BSAMC using the Ising model along with a comparison with the MCMLE method. Consider an Ising model defined on an $N \times N$ lattice, whose likelihood function can be written as

$$f(\boldsymbol{x}|\theta) = \frac{1}{Z(\theta)} \exp\{\theta U(\boldsymbol{x})\}, \tag{4.19}$$

where the negative energy function

$$U(\boldsymbol{x}) = \sum_{i=1}^{N} \sum_{j=1}^{N-1} x_{i,j} x_{i,j+1} + \sum_{i=1}^{N-1} \sum_{j=1}^{N} x_{i,j} x_{i+1,j}, \tag{4.20}$$

and $x_{i,j} \in \{-1, 1\}$. For this model, we impose a uniform prior on $\Theta = [0, 1]$. Then the posterior can be written as

$$\pi(\theta|\mathbf{x}) = \frac{\exp\{\theta U(\boldsymbol{x})\}}{\kappa(\theta)}. \tag{4.21}$$

In simulations, we set $N = 32$ and simulated 50 independent datasets of $\boldsymbol{x}$ for each value of $\theta = 0.2$, 0.3 and 0.4 using the perfect sampler [21]. To show that BSAMC is robust to the initial guess θ_0, we apply BSAMC to the $\theta = 0.3$ datasets with initial guesses, θ_0 =0.275, 0.3, 0.325, 0.35, and 0.375, where 0.35 and 0.375 are far from the true value 0.3. We set $\delta = 0.8$ and partitioned the sample spaces according to δ. For the runs with θ_0 =0.275, 0.3 and 0.325, the sample space was

partitioned into $k = 252$ subregions according to the energy function with $h_1 = -300$ and $h_{k-1} = -100$. For the runs with θ_0 =0.35, 0.375, we set $k = 252$, $h_1 = -320$ and $h_{k-1} = -120$. BSAMC was run in two stages. Stage I consisted of $200k$ iterations where k is the number of subregions, for which we set $m = 10$ and $s = 0$. Recall s denotes the samples generated from $\pi_{\xi_t}(\theta|\boldsymbol{x})$. Stage II consisted of 50 iterations, for which we set $m = 20k$ and $s = 100$. To draw samples from $\pi_{\xi_t}(\theta|\boldsymbol{x})$, we adopted a Gaussian random walk proposal $N(\theta_t^{(i-1)}, a^2 I)$ with $a = 0.05$. The CPU time for a single run is 69.1s on a 3.0 GHz personal computer (all computations reported in this paper were done on this computer).

These settings are made according to the energy range of $f(\boldsymbol{x}|\theta_0)$ with end-point extensions for accommodating the difference of energy ranges of $f(\boldsymbol{x}|\theta_0)$ and $f(\boldsymbol{x}|\theta)$. Choosing h_1 and h_{k-1} to cover the main energy range of $f(\boldsymbol{x}|\theta_0)$ is also important, as this facilitates the mixing of the simulations. The values of t_0 were set to 20000 for the all cases of θ_0. As discussed in Section 2.1, a large value of t_0 should be used for a system which is hard to mix. The numerical results were summarized in Table III. The results indicate that BSAMC works well for all initial guesses, even for the guesses 0.35 and 0.375 which are really far from the true value.

For comparison, the MCMLE (the iterative version) was also applied to these datasets with the same initial guesses $\theta_0 = 0.275$, 0.3, 0.325, 0.35 and 0.375. Each run consisted of 500 iterations, and 5000 samples were generated at each iteration. This setting matches the setting of BSAMC for stage II simulations; the same numbers of auxiliary samples are used in both algorithms. Simulation of auxiliary samples is the major part of CPU cost for both algorithms. MCMLE works well for the runs with $\theta_0 = 0.275$, 0.3, and 0.325, but it often fails to converge for the runs with $\theta_0 = 0.35$ and 0.375. When $\theta_0 = 0.375$, it failed to converge in 49 out of 50 runs! The results were summarized in Table III. This experiment shows that BSAMC is much more

robust to initial guesses than MCMLE.

Table III. Parameter estimation for the Ising model with true $\theta = 0.3$. The estimates were calculated by averaging over 50 data sets, with the standard deviations given in parentheses. * Calculated based on 47 datasets; MCMLE failed to converge for 3 datasets. ** Calculated based on 23 datasets; MCMLE failed to converge for 27 datasets. *** Calculated based on one dataset; MCMLE failed to converge for 49 datasets.

	$\theta = 0.3$	
Initial Guess	BSAMC	MCMLE
$\theta_0 = 0.275$	0.2978 (2.0e-3)	0.2979 (2.0e-3)
$\theta_0 = 0.3$	0.2982 (2.0e-3)	0.2980 (2.0e-3)
$\theta_0 = 0.325$	0.2978 (2.0e-3)	0.2994 (1.8e-3)*
$\theta_0 = 0.35$	0.2978 (1.9e-3)	0.3070 (2.1e-3)**
$\theta_0 = 0.375$	0.2983 (2.0e-3)	0.3183 (N/A)***

In the above simulations of BSAMC, the number of iterations in stage I and the value of m have been set to a function of k. Their values can be much reduced when θ_0 is near the MLE. This can be seen in the next experiment. In this experiment, BSAMC was applied to all datasets generated with θ=0.2, 0.3 and 0.4. For each dataset, θ_0 was set to the MPLE of θ. We set $\delta = 0.8$ and divide the number of subregions according to δ. Each run consisted of two stages. In stage I, we set $s = 0$ and $m = 10$, and set the number of iterations as $200k$ according to the number of subregions for the runs with $\theta = 0.2$, 0.3 and 0.4, respectively. In stage II, we set $s = 100$, set the number of iterations to 50, and set $m = 200k$ for the runs with $\theta = 0.2$, 0.3, and 0.4, respectively. We set $t_0 = 2000$. The other settings, such as sample space partitioning and the number of subregions k, were given in Table IV.

The results were summarized in Table IV. They indicate that BSAMC works well for all datasets.

Table IV. Parameter estimation for the Ising model with θ_0 chosen as the MPLE of θ. ^{a} the value of h_1; ^{b} the value of h_{k-1}. Time: the CPU time cost by a single run of BSAMC. The estimates (standard deviations given in the parentheses) are calculated by averaging over 50 datasets.

	Parameter Estimates	Setting	Time (3.0 GHz CPU)
$\theta = 0.2$	0.1997 (2.2e-3)	(-150^a, -30^b), $k = 75$	53.5s
$\theta = 0.3$	0.2979 (1.9e-3)	(-300^a, -100^b), $k = 250$	68.4s
$\theta = 0.4$	0.3993 (1.6e-3)	(-550^a, -270^b), $k = 350$	103.6s

D. Spatial Models with an Intractable Normalizing Constant

1. Autologistic Model

The autologistic model [9] has been widely used for analysis of spatial lattice data (see, e.g. [60], [73], and [82]). Let $\boldsymbol{x} = \{x_i : i \in D\}$ denote the binary response data, where $x_i \in \{-1, 1\}$ is called a spin and D is the set of indices of spins. Let $|D|$ denote the total number of spins in D, and let $N(i)$ denote the set of nearest neighbors of spin i. The likelihood function of the autologistic model is

$$f(\boldsymbol{x}|\alpha,\beta) = \frac{1}{Z(\alpha,\beta)} \exp\left\{\alpha \sum_{i\in D} x_i + \frac{\beta}{2}\sum_{i\in D} x_i \left(\sum_{j\in N(i)} x_j\right)\right\}, (\alpha,\beta) \in \Theta, \quad (4.22)$$

where α represents the overall proportion of $x_i = +1$ and β represents the intensity of interaction between x_i and its neighbor $N(i)$. The normalizing constant is defined

by

$$Z(\alpha,\beta)=\sum_{\text{for all possible } \boldsymbol{x}} \exp\left\{\alpha\sum_{i\in D}x_i+\frac{\beta}{2}\sum_{i\in D}x_i\left(\sum_{j\in N(i)}x_j\right)\right\}, \tag{4.23}$$

Since an exact calculation of $Z(\alpha,\beta)$ requires summary of over all $2^{|D|}$ possible configurations of $\boldsymbol{x}$, it is impossible to be calculated exactly even for a moderate system.

To conduct a Bayesian analysis for the autologistic model,we assume an uniform prior on

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

For the autologistic model, we set the initial guess $(\alpha^{(0)},\beta^{(0)})$ to the MPLE of (α,β), and draw $(\alpha_i^{(t)},\beta_i^{(t)})$ using the MH algorithm with a Gaussian random walk proposal $N((\alpha_{i-1}^{(t)},\beta_{i-1}^{(t)}),a^2I)$, where we set $a=0.03$ for the U.S. cancer mortality data studied below.

U.S. Cancer Mortality Data. United States cancer mortality maps have been collected by [63] for investigating the possible relation of cancer with unusual demographics, environmental, industrial characteristics, or employment patterns. Figure 2 shows the mortality map of liver and gallbladder (including bile ducts) cancers for white males during the decade 1950-1959, which indicates some apparent geographic clusterings. Refer to [73] for more descriptions of the data. Following [73], we modeled the data by a spatial autologistic model. The total number of spins is $|D|=2293$. Since the boundary points have less neighbor than the interior points, we assume a free boundary condition, which is natural for such an irregular shape lattice.

To conduct a Bayesian analysis for the auotologistic model, we assume an uniform prior for

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

In BSAMC simulations, we set the initial guess $(\alpha^{(0)},\beta^{(0)})$ to the MPLE of (α,β),

Fig. 2. US cancer mortality data. The mortality map of liver and gallbladder cancers (including bile ducts) for white males during the decade 1950-1959. Black squares denote counties of high cancer mortality rate, and white squares denote counties of low cancer mortality rate.

and simulate $(\alpha_t^{(i)}, \beta_t^{(i)})$'s at each iteration using the MH algorithm with a Gaussian random walk proposal $N((\alpha_{i-1}^{(t)}, \beta_{i-1}^{(t)}), a^2 I)$, where the step size $a = 0.03$. BSAMC was run for the data five times. Each run consisted of two stages. Stage I consisted of 200k iterations, for which we set $m = 10$ and $s = 0$. Stage II consisted of 100 iterations, for which we set $m = 20k$ and $s = 100$. We collected every 10th sample at each iteration. We set $\delta = 0.8$ and the sample space was partitioned into 452 subregions according to δ where $h_1 = -650$ and $h_{451} = -350$. The gain factor was set in (4.7) with $t_0 = 2000$. The CPU time cost by a single run is 6.2m. The resulting estimates of (α, β) were summarized in Table V.

For comparison, the exchange algorithm was applied to this example. As aforementioned, the exchange 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

Table V. Parameter estimation for the autologistic model. The MCMLE is from [73] and the MPLE is from [46].

	BSAMC	Exchange	MCMLE	MPLE
α	-0.3006 (4.5e-4)	-0.3015 (1.3e-3)	-0.304 (7.4e-4)	-0.3205
β	0.1232 (2.6e-4)	0.1229 (7.6e-4)	0.117 (1.3e-3)	0.1115

assessing whether the results produced by MCMH are correct. The perfect sampler used here is the summary state algorithm [21], which is known to be suitable for high dimensional binary spaces. The exchange algorithm was run 5 times for this example. 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 estimation of θ. The overall acceptance rate was 0.2, which indicates that the algorithm has been implemented efficiently. The numerical results were summarized in Table V. For a thorough comparison, we also include in the table two estimation results from the literature, the MCMLE from [73] and the MPLE from [46]. The comparison indicates that BSAMC is valid; it can produce almost identical results with the exchange method for this example.

Figure 3 shows histograms, trace, and autocorrelation plots of the last sample θ_t drawn at each iteration of stage II of a BSAMC run. It indicates the BSAMC performs quite stationarily for this example. The autocorrelation plots imply that the samples generated in different iterations are approximately independent.

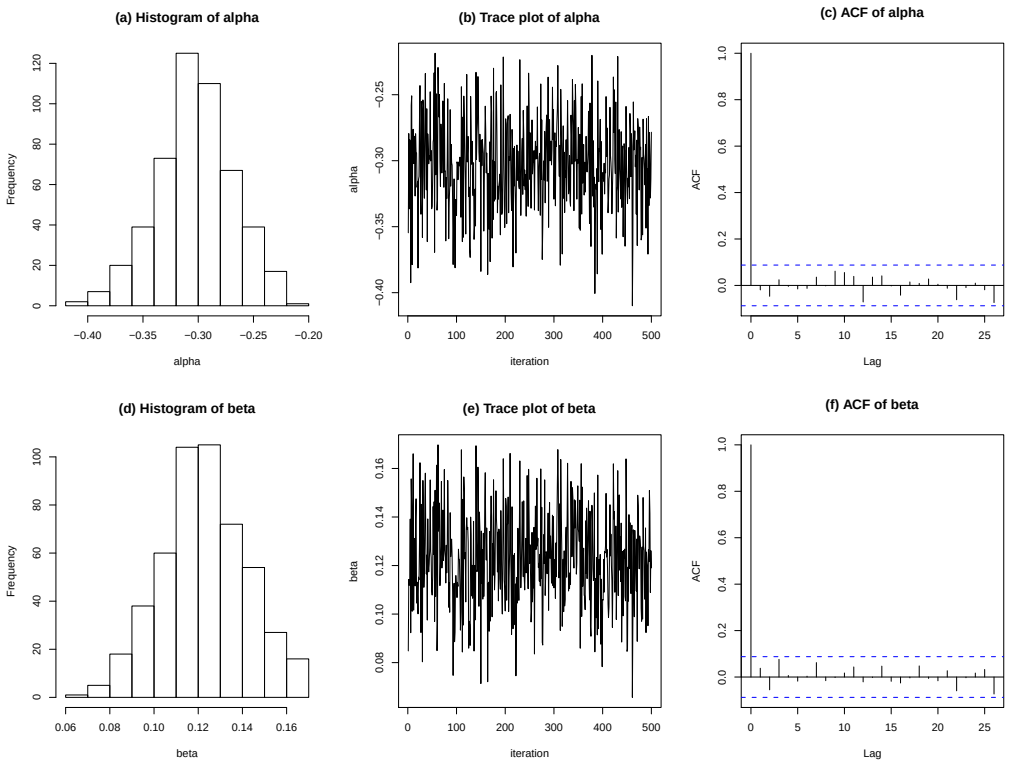

Fig. 3. Histogram, trace and autocorrelation plots of BSAMC samples for the autologistic model. (a)–(c) for the samples of α and (d)–(f) for the samples of β.

2. Autonormal Model

Consider a second-order zero-mean Gaussian Markov random field $\boldsymbol{X} = (X_{i,j})$ defined on $M \times N$ lattice [9], whose conditional density is given by

$$\begin{aligned} & f(x_{i,j} \mid \beta, \sigma^2, x_{u,v}; (u,v) \neq (i,j)) \\ & \quad = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{1}{2\sigma^2}(x_{i,j} - \beta_h \sum_{n_h(i,j)} x_{u,v} - \beta_v \sum_{n_v(i,j)} x_{u,v} - \beta_d \sum_{n_d(i,j)} x_{u,v})^2 \right\}, \end{aligned} \tag{4.24}$$

where $\beta = (\beta_h, \beta_v, \beta_d)$ and σ^2 are parameters, $n_h(i,j) = \{(i,j-1), (i,j+1)\}$, $n_v(i,j) = \{(i-1,j), (i+1,j)\}$ and $n_d(i,j) = \{(i-1,j-1), (i+1,j+1), (i-1,j+1), (i+1,j-1)\}$ are neighbors of (i,j). The model is stationary when $|\beta_h| + |\beta_v| + 2|\beta_d| < 0.5$ [5]. The joint likelihood function of the model is given by

$$f(\boldsymbol{x}|\beta, \sigma^2) = (2\pi\sigma^2)^{-MN/2} |B|^{1/2} \exp\left\{ -\frac{1}{2\sigma^2} \boldsymbol{x}' B \boldsymbol{x} \right\},$$

where $|B|$ is an $(MN \times MN)$-dimensional matrix and $|B|$ is intractable except for some special cases [10].

For a Bayesian analysis, we assume the prior as

$$\pi(\beta) \propto I(|\beta_h| + |\beta_v| + 2|\beta_d| < 0.5), \qquad \pi(\sigma^2) \propto \frac{1}{\sigma^2}, \tag{4.25}$$

where $I(\cdot)$ is the indicator function. Under the free boundary condition, the posterior distribution is

$$\begin{aligned} \pi(\beta, \sigma^2|\mathbf{x}) & \propto (2\pi\sigma^2)^{-MN/2-1} |B|^{1/2} \\ & \quad \times \exp\left\{ -\frac{MN}{2\sigma^2}(S_x - 2\beta_h X_h - 2\beta_v X_v - 2\beta_d X_d) \right\} I(|\beta_h| + |\beta_v| + 2|\beta_d| < 0.5), \end{aligned} \tag{4.26}$$

where

$$
\begin{aligned}
S_x &= \frac{1}{MN}\sum_{i=1}^{M}\sum_{i=1}^{N} x_{i,j}^2, \\
X_h &= \frac{1}{MN}\sum_{i=1}^{M}\sum_{i=1}^{N-1} x_{i,j}^2 x_{i,j+1}, \\
X_v &= \frac{1}{MN}\sum_{i=1}^{M-1}\sum_{i=1}^{N} x_{i,j}^2 x_{i+1,j}, \\
X_d &= \frac{1}{MN}\left(\sum_{i=1}^{M-1}\sum_{i=1}^{N-1} x_{i,j}^2 x_{i+1,j+1} + \sum_{i=1}^{M-1}\sum_{i=2}^{N} x_{i,j}^2 x_{i+1,j-1}\right).
\end{aligned}
$$

Although σ^2 can be integrated out, we do not suggest to do so, as this facilitate the sampling of $x_{i,j}$'s in our comparison studies. Also, to facilitate the sampling of σ^2, we reparameterize σ^2 by $\tau = \log(\sigma^2)$ in simulations. In step (a), a single cycle of the Metropolis-within-Gibbs update [56] was used for drawing samples of $\boldsymbol{X}$. In step (c), $(\beta_i^{(t)}, \tau_i^{(t)})$, the current state of Markov chain, is updated by a MH step with a Gaussian random walk proposal $N((\beta_{i-1}^{(t)}, \tau_{i-1}^{(t)})', a^2 I_4)$, where $a = 0.02$ for the wheat yield example studied below. In BSAMC, we treat $|B|$ as intractable.

Wheat Yield Data. This data, shown in Figure 4(a), was collected on a 20×25 rectangular lattice (Tables 6.1. [1]). This data has been analyzed by a number of authors, e.g., [9], [37], [33] and [48]. Following the previous authors, we subtracted the mean from the data and then fitted the data by the autonormal model. In our analysis, the free boundary condition is assumed. This is natural, as the lattice is often irregular for the real data.

BSAMC was applied to this example with 5 independent runs. Each run consisted of two stages. Stage I consisted of 200k iterations, and stage II consisted of 100 iterations where every 10th sample was collected at each iteration. In stage I, we set $m = 10$ and $s = 0$; and in stage II, we set $m = 20k$ and $s = 100$. The sample space

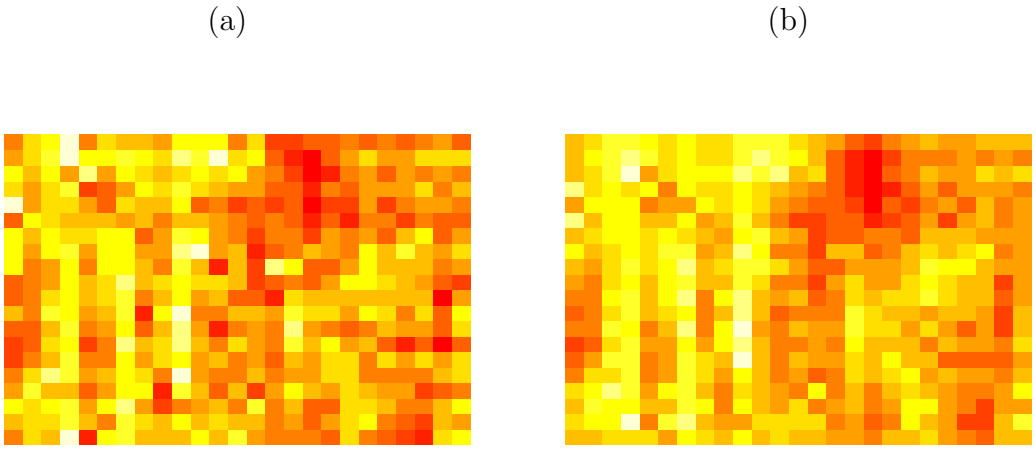

Fig. 4. Image of the wheat yield data. (a) Image of real wheat yield data (b) Image of fitted wheat yield data using BSAMC: black squares denote high yield area, and white squares denote low yield area.

was partitioned into 452 subregions according to the energy function of the model and $\delta = 0.8$, with $h_1 = -650$ and $h_{451} = -350$. The gain factor sequence was set in (4.7) with $t_0 = 2000$. The CPU time of a single run is about 2.5m in a 3.0 GHz personal computer. The numerical results were summarized in Table VI. For comparison, we also gave in Table VI the true Bayesian estimates and the double MH estimates which are reported by [48]. The former was obtained by directly simulating from (4.26) with an analytical expression of $|B|$ [5], and the latter was obtained by the double MH algorithm. Like MPLE, the double MH estimate only works approximately, lacking a theoretical justification for its consistency. The comparison indicates that BSAMC is valid; it can produce almost identical results with the true Bayesian method for this example.

Figure 5 shows the histogram, trace and autocorrelation plots of the samples of $(\beta_1, \beta_2, \beta_3, \sigma)$ generated by BSAMC in a run. It indicates that BSAMC performs

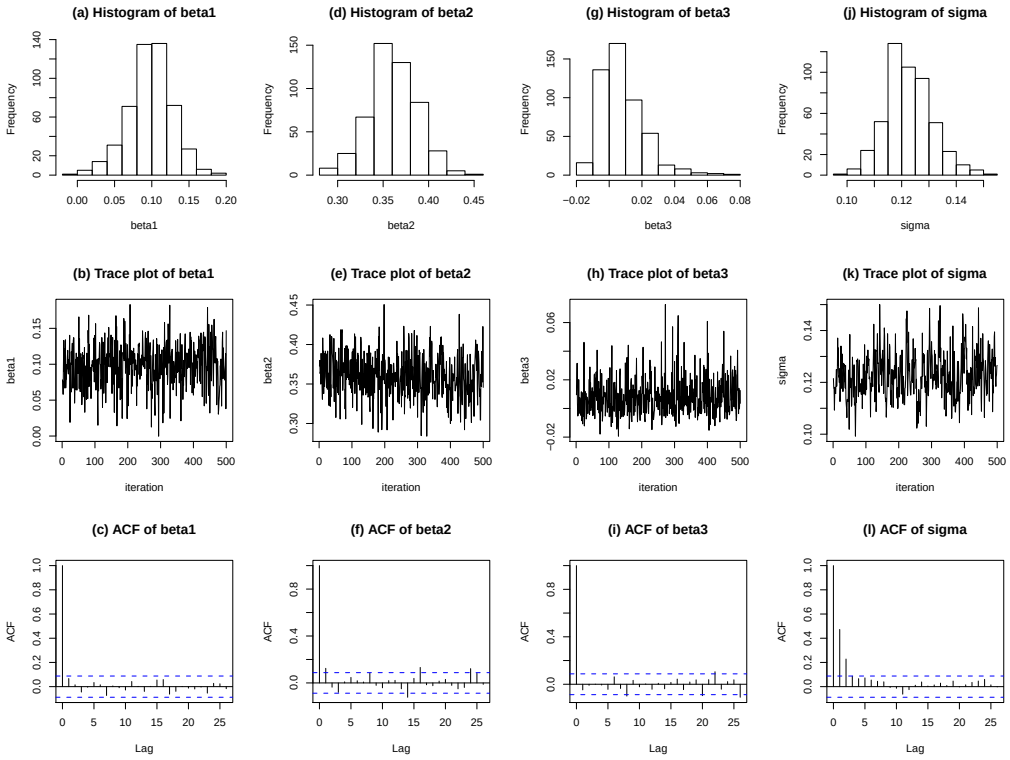

Fig. 5. Histogram, trace and autocorrelation plots of BSAMC samples for the autonormal model. (a - c) for β_1, (d - f) for β_2, (g - i) for β_3, and (j - l) for σ.

Table VI. Estimation of the autonormal model for the wheat yield data. The results of true Bayes and double MH are from [48].

Method	β_1	β_2	β_3	σ
BSAMC	0.096 (2e-3)	0.362 (1e-3)	0.007 (5e-4)	0.123 (2e-4)
True Bayes	0.102 (4e-4)	0.355 (3e-4)	0.006 (2e-4)	0.123 (2e-4)
double MH	0.099 (6e-4)	0.351 (5e-4)	0.006 (3e-4)	0.126 (3e-4)
MPLE	0.140	0.340	-0.010	0.122

quite stably for this example. The autocorrelation plots imply that the samples of $(\beta_1, \beta_2, \beta_3)$ obtained in different iterations are approximately independent, while the samples of σ have a short autocorrelation.

CHAPTER V

FITTING ERGMS USING VARYING TRUNCATION STOCHASTIC APPROXIMATION MCMC ALGORITHM

A. Introduction

In this paper, we propose to use the stochastic approximation MCMC (SAMCMC) algorithm [49] to find the MLE for ERGMs. Like the SSA algorithm, SAMCMC is rooted in the stochastic approximation algorithm. But it is fundamentally different from SSA in two aspects. Firstly, it avoids the requirement for independent network samples. In SAMCMC, $\boldsymbol{y}_{k+1}$ can be generated via a short MH run starting with $\boldsymbol{y}_k$. This generally improves efficiency of the simulation. Secondly, SAMCMC works under the framework of varying truncation stochastic approximation algorithms ([2], [17]). The varying truncation mechanism enables SAMCMC to overcome the model degeneracy problem. In degeneracy regions, SAMCMC tend to produce large updates in the parameters due to generation of complete or empty networks. This will trigger the varying truncation mechanism to force the simulation to be re-initialized. The re-initialization enables SAMCMC to move out of degeneracy regions. Under mild conditions, we show that the resulting estimator is consistent, asymptotically normal, and asymptotically efficient. The SAMCMC method is illustrated using a variety of networks, including the Florentine business network, Kapferer's tailor shop network, Lazega's lawyer network, and Zachary's Karate network. The numerical results indicate that SAMCMC can significantly outperform MCMLE and SSA. For the ERGMs which consist of basic Markovian statistics, the MCMLE and SSA methods often fail

due to the model degeneracy, while SAMCMC still works well. For the ERGMs which do not suffer from the model degeneracy, SAMCMC can work as well as or better than the MCMLE and SSA methods.

The remainder of this paper is organized as follows. In Section B, we describe the SAMCMC algorithm and study its theoretical property. In Section C, we apply SAMCMC to a variety of social network examples along with comparisons with the MCMLE and SSA methods. In Section D, we apply SAMCMC to a large network study, the high school student friendship network.

B. Stochastic Approximation MCMC with Trajectory Averaging

The subject of stochastic approximation was founded by [64]. After five decades of continual development, it has developed into an important area in systems control and optimization, and it has also served as a prototype for development of recursive algorithms for on-line estimation and control of stochastic systems. [18] proposed a varying truncation version of the stochastic approximation algorithm, which removes the growth rate restriction imposed on the mean field function and weakens the conditions imposed on noise in showing the convergence of the algorithm. [2] proved the convergence of the varying truncation stochastic approximation algorithm for a wide class of mean field functions with Markov state-dependent noise. Quite recently, [49] showed that the trajectory averaging technique used in traditional stochastic approximation algorithms can also be applied to the varying truncation stochastic approximation MCMC (SAMCMC) algorithm. In this section, we first give a brief review for the varying truncation SAMCMC algorithm, and then give the details how the algorithm can be applied to ERGMs.

1. Varying Truncation Stochastic Approximation MCMC Algorithm

Suppose that we want to solve the integration equation

$$\int_{\mathcal{X}} H(\boldsymbol{y}, \theta) f(\boldsymbol{y}|\theta) d\boldsymbol{y} = 0, \tag{5.1}$$

where $\mathcal{X}$ denotes the sample space of the distribution $f(\boldsymbol{y}|\theta)$. This equation can be solved using the varying truncation stochastic approximation MCMC algorithm as follows.

Let $\{\mathcal{K}_s, s \geq 0\}$ denote a sequence of compact sets of Θ such that

$$\bigcup_{s \geq 0} \mathcal{K}_s = \Theta, \quad \text{and} \quad \mathcal{K}_s \subset \text{int}(\mathcal{K}_{s+1}), \tag{5.2}$$

where $\text{int}(A)$ denotes the interior of set A. Let $\mathcal{X}_0$ be a subset of $\mathcal{X}$, and let $\mathcal{T} : \mathcal{X} \times \Theta \to \mathcal{X}_0 \times \mathcal{K}_0$ be a measurable function which maps a point $(\boldsymbol{y}, \theta)$ in $\mathcal{X} \times \Theta$ to a random point in the initial region $\mathcal{X}_0 \times \mathcal{K}_0$; that is, $\mathcal{T}$ defines a re-initialization mechanism, re-initializing the simulation in the set $\mathcal{X}_0 \times \mathcal{K}_0$. Other types of re-initialization mechanism is also possible, but needs a little different theory.

Let $\{a_k\}$ and $\{b_k\}$ be two positive sequences satisfying the condition (A_4) (given in Appendix A). Let σ_k denote the number of truncations performed until iteration k, and $\sigma_0 = 0$. The stochastic approximation MCMC algorithm starts with a random choice of $(\boldsymbol{y}_0, \theta^{(0)})$ in the space $\mathcal{X}_0 \times \mathcal{K}_0$, and then iterates between the following steps:

Varying Truncation SAMCMC algorithm

(a) Draw a sample $\boldsymbol{y}_{k+1}$ with a Markov transition kernel, which admits $f(\boldsymbol{y}|\theta^{(k)})$ as the invariant distribution.

(b) Set $\theta^{(k+\frac{1}{2})} = \theta^{(k)} + a_k H(\boldsymbol{y}_{k+1}, \theta^{(k)})$

(c) If $||\theta^{(k+\frac{1}{2})} - \theta^{(k)}|| \leq b_k$ and $\theta^{(k+\frac{1}{2})} \in \mathcal{K}_{\sigma_k}$, where $\|z\|$ denote the Euclidean norm

of the vector z, then set $\sigma_{k+1} = \sigma_k$ and $(\boldsymbol{y}_{k+1}, \theta^{(k+1)}) = (\boldsymbol{y}_{k+1}, \theta^{(k+\frac{1}{2})})$; otherwise set $\sigma_{k+1} = \sigma_k + 1$ and $(\boldsymbol{y}_{k+1}, \theta^{(k+1)}) = \mathcal{T}(\boldsymbol{y}_k, \theta^{(k)})$.

The SAMCMC algorithm is an adaptive algorithm, because it is re-initialized with a smaller initial value of the gain factor a_k, and a larger truncation set $\mathcal{K}_{\sigma+1}$ when the current parameter estimates are outside the active truncation set or when the difference between two successive estimates is greater than a time-threshold value b_k. This varying truncation mechanism enables the algorithm to select an appropriate gain factor sequence and a starting point automatically. Note that, as shown in [2], the number of re-initializations is almost surely finite for every $(\boldsymbol{y}_0, \theta^{(0)}) \in \mathcal{X}_0 \times \mathcal{K}_0$. Under the conditions (A_1)–(A_4) given in Appendix A, [49] showed that the trajectory averaging estimator of θ, i.e., $\sum_{k=1}^{n} \theta^{(k)}/n$, is asymptotically efficient. Refer to Theorem .1 of Appendix A for the details. The self-adaptivity of the SAMCMC algorithm plays a crucial role for establishing asymptotic efficiency of the trajectory averaging estimator.

2. Varying Truncation SAMCMC for ERGMs

To apply the varying truncation SAMCMC algorithm to ERGMs, we set the sequences $\{a_k\}$ and $\{b_k\}$ as follows:

(C_1) Set

$$a_k = C_a \left(\frac{k_0}{\max(k_0, k)} \right)^{\eta}, \quad b_k = C_b \left(\frac{t_0}{\max(k_0, k)} \right)^{\xi}, \tag{5.3}$$

for some constants $k_0 > 1$, $\eta \in (1/2, 1)$, $\xi \in (1/2, \eta)$, $C_a > 0$, and $C_b > 0$.

How to choose the values of the parameters C_a, C_b, k_0, η and ξ will be discussed at the end of this section.

Let $\{\mathcal{K}_s, s \geq 0\}$ denote a sequence of compact sets of Θ, which satisfy the following condition:

(C_2) $\{\mathcal{K}_s, s \geq 0\}$ satisfies (5.2) and there exist constants l_0 and l_1 such that $l_0 > l_1$, $\mathcal{K}_0 \subset \{\theta \in \Theta : l(\theta|\boldsymbol{y}_{obs}) > l_0\}$, and $\{\theta \in \Theta : l(\theta|\boldsymbol{y}_{obs}) \geq l_1\}$ is compact, where $l(\theta|\boldsymbol{y}_{obs})$ denotes the log-likelihood function of the model under consideration.

In this paper, we set $\mathcal{K}_s$ to be a product rectangle; that is, $\mathcal{K}_s = \mathcal{K}_{s,1} \times \mathcal{K}_{s,2} \times \cdots \times \mathcal{K}_{s,d}$, where $\mathcal{K}_{s,i}$ corresponds to the parameter θ_i in (2.1) and is of the form $[-d_i(s+1), d_i(s+1)]$. How to choose d_i's will be discussed at the end of this section. By the continuity of $l(\theta|\boldsymbol{y}_{obs})$, it is easy to see that (C_2) is satisfied. For ERGMs, the sample space $\mathcal{X}$ is finite, we set $\mathcal{X}_0 = \mathcal{X}$; that is, each run starts or is re-initialized with a random configuration of the network.

In summary, one iteration of the algorithm consists of the following steps:

Varying truncation SAMCMC for ERGMs

(a) Draw an auxiliary social network $\boldsymbol{y}_{k+1}$ from the distribution $f(\boldsymbol{y}|\theta^{(k)})$ using the Gibbs sampler, which starts with the network $\boldsymbol{y}_k$ and iterates for m sweeps.

(b) Set $\theta^{(k+\frac{1}{2})} = \theta^{(k)} + a_k \left(S(\boldsymbol{y}_{k+1}) - S(\boldsymbol{y}_{obs})\right)$.

(c) If $||\theta^{(k+\frac{1}{2})} - \theta^{(k)}|| \leq b_k$ and $\theta^{(k+\frac{1}{2})} \in \mathcal{K}_{\sigma_k}$, where $\|z\|$ denote the Euclidean norm of the vector z, then set $\sigma_{k+1} = \sigma_k$ and $(\boldsymbol{y}_{k+1}, \theta^{(k+1)}) = (\boldsymbol{y}_{k+1}, \theta^{(k+\frac{1}{2})})$; otherwise set $\sigma_{k+1} = \sigma_k + 1$ and $(\boldsymbol{y}_{k+1}, \theta^{(k+1)}) = \mathcal{T}(\boldsymbol{y}_k, \theta^{(k)})$.

For this algorithm, θ can be estimated by the trajectory averaging estimator

$$\bar{\theta}_n = \sum_{k=1}^{n} \theta^{(k)}/n. \tag{5.4}$$

In practice, to reduce the variation of the estimate, we often use

$$\bar{\theta}(n_0, n) = \frac{1}{n - n_0} \sum_{k=n_0+1}^{n} \theta^{(k)}, \tag{5.5}$$

to estimate θ, where n_0 denotes the number of burn-in iterations and it is usually set to a value at which the last truncation occurs.

Theorem B.1 concerns the convergence and asymptotic efficiency of $\bar{\theta}_n$, whose proof can be found in Appendix B. To make the theory more general (the auxiliary network samples can be generated using the MH algorithm), we further assume the following condition for the proposal distribution used in step (a):

(C_3) (Local positive) For every $\boldsymbol{y} \in \mathcal{X}$, there exist positive ϵ_1 and ϵ_2 such that

$$\|\boldsymbol{z} - \boldsymbol{y}\| \leq \epsilon_1 \Longrightarrow q(\boldsymbol{z}|\boldsymbol{y}) \geq \epsilon_2, \tag{5.6}$$

where $q(\boldsymbol{z}|\boldsymbol{y})$ denotes the proposal distribution conditioned on the current sample $\boldsymbol{y}$.

It is easy to see that the Gibbs sampler used in step (a) satisfies the local positive condition by noting that the Gibbs sampler is special case of the MH algorithm and that only a single arc variable is updated at each updating step.

Theorem B.1 *Assume that the conditions* (C_1), (C_2) *and* (C_3) *hold. Then, as* $n \to \infty$, *we have*

1. *(Convergence)* $\theta^{(n)} \to \theta^*$ *almost surely, where* θ^* *denotes a solution of equation (2.2).*
2. *(Asymptotic Normality)*

$$\sqrt{n}(\bar{\theta}_n - \theta^*) \longrightarrow N(\mathbf{0}, \Gamma),$$

where Γ *is a negative definite matrix independent of the sequences* $\{a_k\}$ *and* $\{b_k\}$. *See [49] for an explicit form of* Γ.

3. *(Asymptotic Efficiency)* $\bar{\theta}_n$ *is asymptotically efficient; that is,* Γ *is the smallest*

possible limit covariance matrix that an estimator based on a stochastic approximation algorithm can achieve.

Although both SSA and SAMCMC originate in the stochastic approximation algorithm, they are different in several aspects. Firstly, SAMCMC avoids the requirement of independent samples. As described above, $\boldsymbol{y}_{k+1}$ can be generated by the Gibbs sampler starting with $\boldsymbol{y}_k$. Note that Theorem B.1 holds for any value of $m \geq 1$. In one numerical example, we show that the value of m does not significantly affect the convergence of the trajectory averaging estimator. Secondly, SAMCMC includes the varying truncation step, which enables SAMCMC to overcome the model degeneracy problem. When the model degeneracy occurs, the sampled network $\boldsymbol{y}_k$ tends to be complete or empty. In this case, $\|\boldsymbol{S}(\boldsymbol{y}_k) - \boldsymbol{S}(\boldsymbol{y}_{obs})\|$ tends to have a large value, and thus re-initialization will be triggered. This enables SAMCMC to move out of the degeneracy region. Thirdly, SAMCMC avoids estimation for the covariance matrix D used in (2.3). This greatly simplifies computation. As shown in Theorem B.1, the asymptotic efficiency of the SAMCMC estimate can be obtained by averaging over its trajectory.

The SAMCMC algorithm consists of several free parameters, including the sequences $\{a_k\}$ and $\{b_k\}$, the compact sets $\{\mathcal{K}_s, s \geq 0\}$, and the number (m) of Gibbs sweeps used at each iteration. As shown in Theorem B.1, the choice of the sequences $\{a_k\}$ and $\{b_k\}$ will not affect the efficiency of the algorithm as long as the condition (C_1) is satisfied. This gives us much freedom for choosing the two sequences. In this paper, we fix $k_0 = 100$, $\eta = 0.65$, $\xi = (0.5 + \eta)/2$, $C_b = 1000$ and leave C_a as a free parameter to be adjusted for different examples. Since C_a determines the learning rate of θ_k, it is reasonable to set its value according to the variation of $S(\boldsymbol{y}_k)$. If the variation is large, a small value may be set for C_a, say, $C_a = 0.001$. Otherwise, a little

larger value, say, $C_a = 0.01$ may be set for C_a. This ensures the update of θ_t not to be very large at a single iteration.

The choice of the compact sets $\{\mathcal{K}_s, s \geq 0\}$ is quite critical to the performance of SAMCMC. If the sets, especially $\mathcal{K}_0$, are not chosen appropriately, a lot of truncations will occur in simulations, and this will delay the convergence of the simulation. We suggest to choose $\mathcal{K}_0$ to be around the MPLE [76], and then to enlarge $\mathcal{K}_s$ ($s \geq 1$) gradually. For simplicity, we set in this paper $\mathcal{K}_{s,1} = [-4(s+1), 4(s+1)]$ and $\mathcal{K}_{s,2} = \cdots = \mathcal{K}_{s,d} = [-2(s+1), 2(s+1)]$. The reason why $\mathcal{K}_{s,1}$ is separated from others is that $S_1(\boldsymbol{y})$ always denotes the edge count in our examples, whose coefficient θ_1 has usually a value around -3 and is greater than other coefficients in magnitudes.

On the number of Gibbs sweeps used at each iteration, we note that a small value of m will result in a smooth trajectory due to the strong dependency between the samples generated in successive iterations, and a high value of m will result in a relatively rough trajectory. However, the value of m will not significantly affect the convergence of SAMCMC, see e.g., Figure 8. An excessively large value of m may cause some waste of CPU times. For computational simplicity, we set $m = 1$ for all examples of this paper.

C. Numerical Examples

To illustrate the performance of SAMCMC, we consider in this section four examples, including the Florentine business network, Kapferer's tailor shop network, Lazega's lawyer network, and Karate network, which are shown in Figure 6. For the first two networks, we consider some models with basic Markovian statistics, which are known as the main reason for model degeneracy. Using these networks, we show that SAMCMC can potentially avoid the model degeneracy problem. While the MCMLE

and SSA methods fail to produce any reasonable estimates for these networks due to model degeneracy. Using the last two networks, we show that SAMCMC can work as well as or better than the MCMLE and SSA methods for the models which do not suffer from the degeneracy problem.

SAMCMC was run five times independently for each example. Each run consisted of 200,000 iterations and the estimates produced in the last 150,000 iterations were averaged to get the final estimate. In simulations, we set $C_a = 0.001$ for Kapferer's tailor shop network and $C_a = 0.01$ for others. As explained previously, this is due to that $\boldsymbol{S}(\boldsymbol{y})$ has a large variation for Kapferer's tailor shop network. All other parameters were set to their default values as given at the end of previous section.

For comparison, MCMLE and SSA, which both have been implemented in the `ergm` package [39], were also applied to these examples. MCMLE was run 5 times for each example. Each run consisted of 25 iterations with 50,000 auxiliary networks being generated at each iteration. SSA was also run 5 times for each example, each run consisting of 10 iterations. Other parameters (for both MCMLE and SSA) were set to their default values as suggested in the `ergm` package. As shown below, both MCMLE and SSA cost longer CPU times than MCMLE in all examples. All computations for the three algorithms were done on a 3.0GHz Intel Core 2 Duo computer.

The goodness-of-fit (GOF) plot [39] was used as the tool for assessing the performance of the three algorithms. 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. It is clear that if the statistics of the observed network, which are represented by a solid line in the GOF plots, falls 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.

1. Florentine Business Network

This network (shown in Figure 6) was collected by [59] from historic documents, which represents a set of business ties, such as loans, credits and joint partnership, among Renaissance Florentine families. The network consists of 16 families who were locked in a struggle for political control of the city of Florence around 1430. Two factions were dominant in this struggle: one revolved around the infamous Medicis, and the other around the powerful Strozzis.

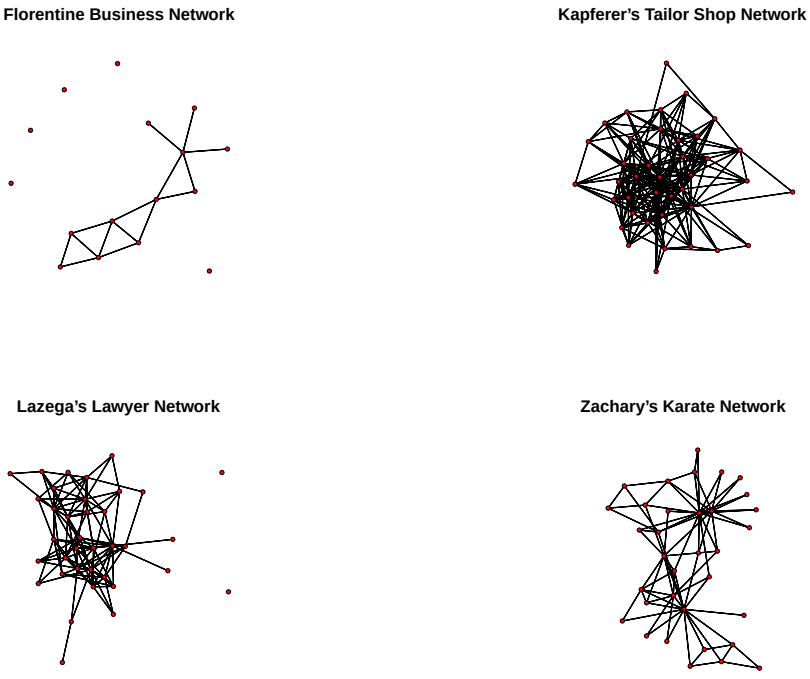

Fig. 6. Social network examples. (a) Florentine business network; (b) Kapferer's tailor shop network; (c) Lazega's lawyer network; (d) Karate network.

We analyzed this network using an ERGM with the edge and 2-star counts. The likelihood function of this model is given by

$$f(\boldsymbol{y}|\theta) = \frac{1}{\kappa(\theta)} \exp\left\{\theta_1 S_1(\boldsymbol{y}) + \theta_2 S_2(\boldsymbol{y})\right\}, \tag{5.7}$$

where $S_1(\boldsymbol{y})$ is the edge count and $S_2(\boldsymbol{y})$ is the 2-star count. The estimates obtained by SAMCMC, MCMLE and SSA are summarized in Table VII, and the corresponding GOF plots are shown in Figure 7. The GOF plots indicate that SAMCMC provides a much better fitting than MCMLE and SSA for this network. Both the MCMLE and SSA estimates fall into a degeneracy region of the model (5.7), where complete networks tend to be generated. However, SAMCMC avoided the model degeneracy problem and produced an estimate for which the simulated networks match well with the observed network. It is also remarkable that SAMCMC is computationally very efficient, which costs only 3.2s for a single run. Both MCMLE and SSA are much more time consuming than SAMCMC.

Table VII. Parameter estimates the Florentine business network. The standard deviations given in the parentheses.

	Edge Count(θ_1)	K2-Star(θ_2)	CPU
SAMCMC	-2.733 (4.2×10^{-4})	0.198 (9.0×10^{-5})	3.2s
MCMLE	-3.191 (2.6×10^{-1})	0.412 (1.2×10^{-1})	78.1s
SSA	-2.842 (7.7×10^{-3})	0.283 (3.9×10^{-2})	370.4s

For this example, we also assessed the effect of m, the number of Gibbs sweeps used for generating auxiliary networks at each iteration, on the performance of SAMCMC. Figure 8 shows the trajectories of θ produced by SAMCMC in three runs with $m = 1$, 5 and 10, where the sample frequencies have been adjusted such that the same number of estimates are collected within the same CPU time in each run. We collected estimates at every 100^{th}, 20^{th} and 10^{th} iterations in the runs with $m = 1$, 5 and 10, respectively. The trajectories show some fluctuations at the early stage of the simulations, which are caused by varying truncations. Figure 8 suggests that $\theta^{(n)}$ can

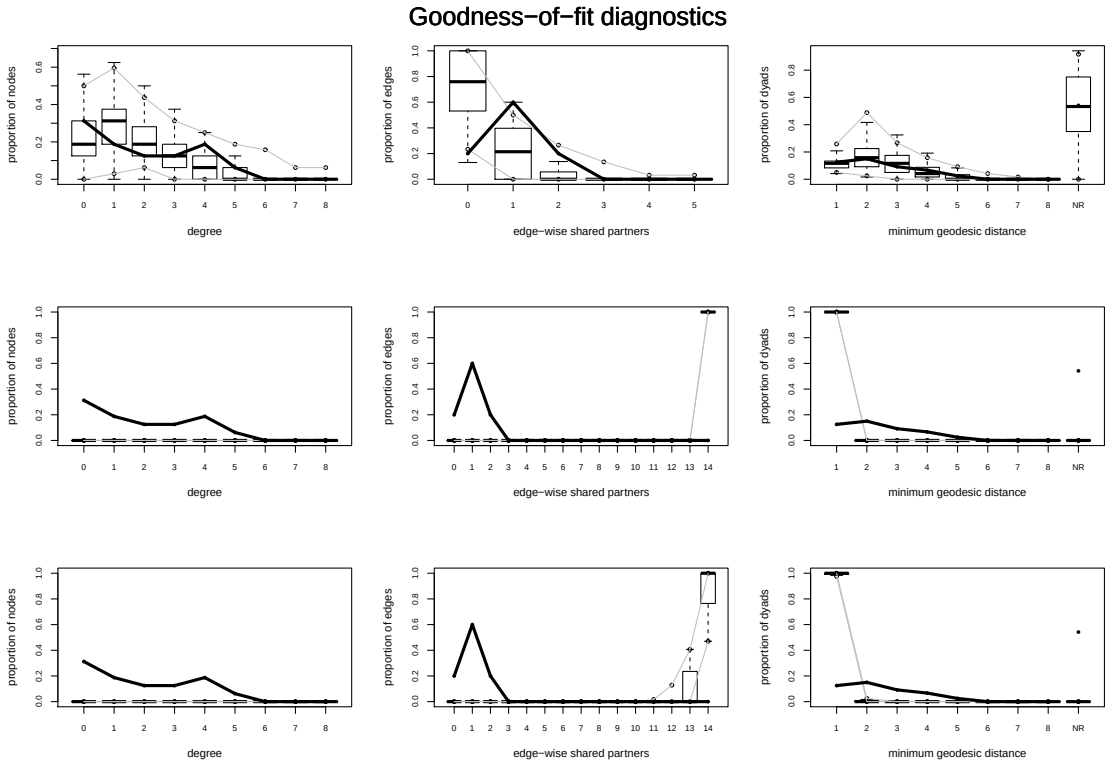

Fig. 7. Goodness-of-fit(GOF) plots for the Florentine business network. Row 1: SAM-CMC; Row 2: MCMLE; Row 3: SSA. The solid line shows the observed network statistics, and the box-plots represent the distributions of simulated network statistics.

converge to the same value in all three runs, and the value of m does not significantly affect the convergence of SAMCMC. When m is small, the networks generated in successive iterations are highly correlated, so the trajectory looks smooth. In contrast, when m is large, the trajectory looks a little rough, as the networks generated in successive iterations are less correlated.

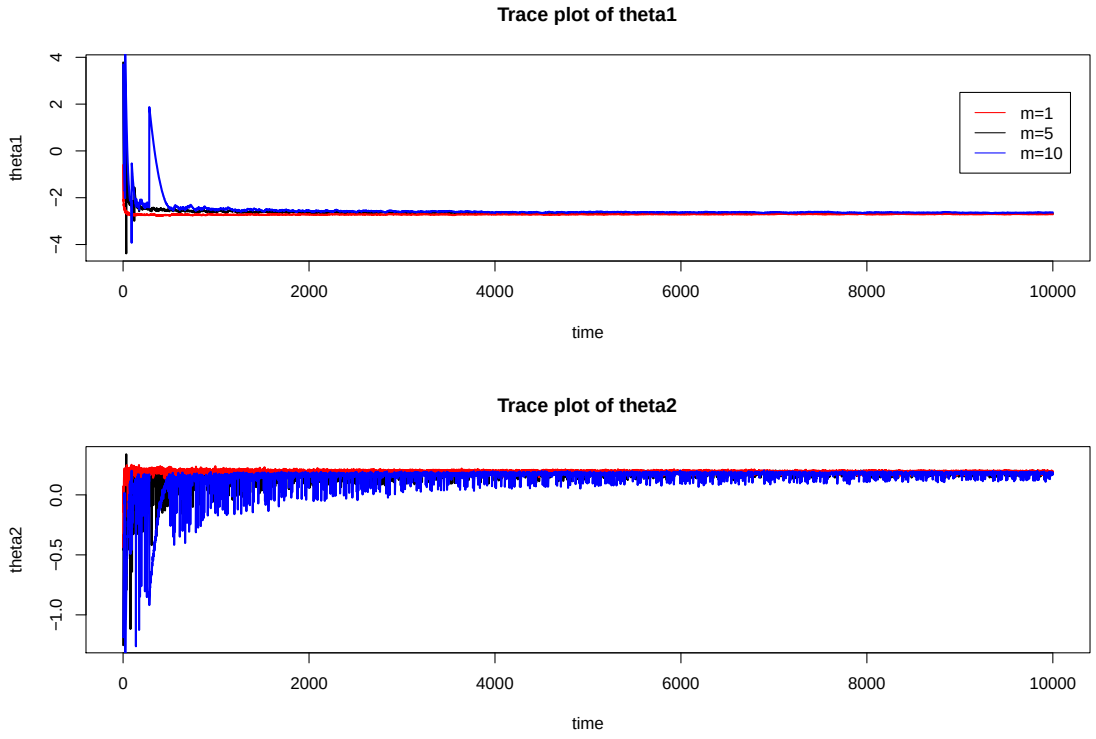

Fig. 8. Trajectories of θ produced by SAMCMC for the Florentine business network with different values of m. $m = 1$ (red), $m = 5$ (black) and $m = 10$ (blue).

2. Kapferer's Tailor Shop Network

[43] collected interactions in a tailor shop in Zambia (then Northern Rhodesia) over a period of ten months, with the focus on changing patterns of alliance among workers during extended negotiations for higher wages. There are two different types of interactions, the "instrumental" (work- and assistance-related) interaction and the

"sociational" (friendship, socioemotional) interaction, which are recorded at two different times (seven months apart). This dataset is particularly interesting because an abortive strike occurred after the first time of observations, and a successful strike took place after the second time of observations. In this paper, we analyze the "sociational" network recorded at the second time of observations. It consists of 39 nodes and is shown in Figure 6.

For this network, we consider an ERGM model with the likelihood function given by

$$f(\boldsymbol{y}|\theta) = \frac{1}{\kappa(\theta)} \exp\left\{\theta_1 S_1(\boldsymbol{y}) + \theta_2 S_2(\boldsymbol{y}) + \theta_3 v(\boldsymbol{y}|\tau)\right\}, \tag{5.8}$$

where $S_1(\boldsymbol{y})$ denotes the edge count, $S_2(\boldsymbol{y})$ denotes the 2-star count, and $v(\boldsymbol{y}|\tau)$ denotes the geometrically weighted edgewise shared distribution defined with $\tau = \log 2$. Table VIII summarizes the estimates obtained by SAMCMC, MCMLE and SSA, and Figure 9 shows the GOF plots of the respective estimates. The GOF plots indicate that SAMCMC produces a much better fitting than MCMLE and SSA for this network. The MCMLE and SSA estimates fall into a degeneracy region of the model (5.8) as shown in Figure 9. However, SAMCMC avoids the model degeneracy problem for this network, and produced an estimate for which the simulated networks match well with the observed network. In addition, SAMCMC costs much less CPU time than MCMLE and SSA for this example.

Kapferer's network has been used as a benchmark example in the literature for testing whether a method can avoid the model degeneracy problem. It is known that this network has two degeneracy regions, which are around $(-20, 0, 17)$ and $(-350, 0, 350)$, respectively. To apply SAMCMC to this benchmark example, we conducted two experiments. In the first experiment, we set $\theta^{(0)} = (-20, 0, 17)$, $\mathcal{K}_{s,1} = \mathcal{K}_{s,3} = [-4(s+5), 4(s+5)]$, and $\mathcal{K}_{s,2} = [-(s+1), (s+1)]$, which covers the known

Table VIII. Estimates of θ produced by SAMCMC, MCMLE and SSA for the Kapferer's tailor shop network. Standard deviations are shown in the parentheses. The MCMLE estimates are calculated based on 4 runs only, as it failed to produce an estimate in one run.

Methods	Edge Count(θ_1)	K2-star(θ_2)	GWESP	Time
SAMCMC	-4.056 (5.7×10^{-3})	0.038 (2.4×10^{-4})	0.962 (1.0×10^{-3})	4.3m
MCMLE	-4.256 (1.8×10^{-5})	0.089 (1.2×10^{-3})	0.542 (3.4×10^{-5})	204.8m
SSA	-3.927 (3.8×10^{-3})	0.068 (8.4×10^{-5})	0.571 (5.7×10^{-4})	99.9m

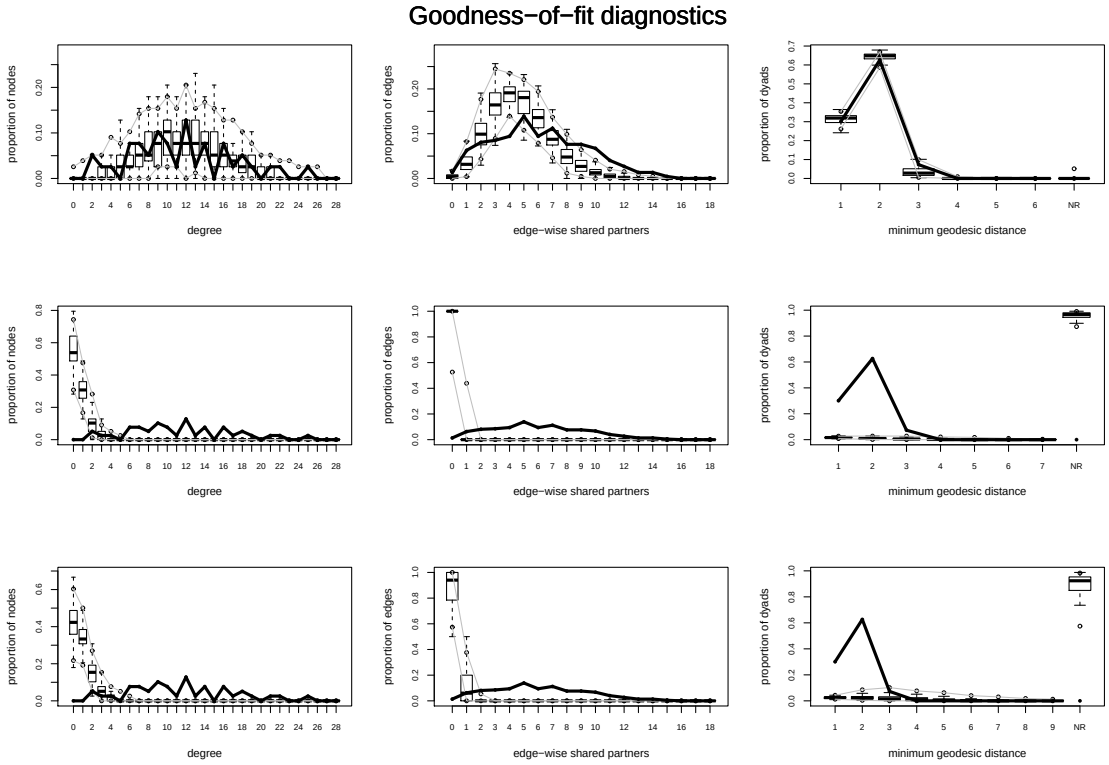

Fig. 9. Goodness-of-fit(GOF) plots for Kapferer's tailor shop network. Row 1: SAMCMC; Row 2: MCMLE; Row 3: SSA.

degeneracy point $(-20, 0, 17)$. The other parameters were the same as those used in previous runs. SAMCMC was run 5 times. Each run consisted of 300,000 iterations, and the estimates produced in the last 100,000 iterations were averaged for final estimation. The resulting estimates are summarized in Table IX, which are very close to the previous estimates shown in Table VIII.

Table IX. Estimates of θ produced by SAMCMC for Kapferer's tailor Shop network with different starting points.

Starting Points	Edge Count(θ_1)	K2-star(θ_2)	GWESP(θ_3)
$(-20, 0, 17)$	-4.015 (4.4×10^{-3})	0.040 (6.9×10^{-5})	0.921 (2.5×10^{-3})
$(-350, 0, 350)$	-3.886 (3.9×10^{-4})	0.028 (3.9×10^{-5})	0.954 (3.5×10^{-4})

In the second experiment, we set $\theta^{(0)} = (-350, 0, 350)$, $\mathcal{K}_{s,1} = \mathcal{K}_{s,3} = [-4(s + 90), 4(s + 90)]$, and $\mathcal{K}_{s,2} = [-(s + 1), (s + 1)]$, which covers the known degeneracy point $(-350, 0, 350)$. The other parameters were the same as those used in previous runs. Each run consisted of 1,100,000 iterations, and the estimates produced in the last 100,000 iterations were used for final estimation. The resulting estimates are also summarized in Table IX, which are slightly different from the previous ones. The reason can be explained as follows. We checked the details of the five runs. The numbers of truncations in these runs are 112, 101, 62, 113, and 94, respectively; and the last re-initialization points are (337.8, -38.9, 95.6), (226.2, -8.6, -92.7), (191.4, -5.3, 4.6), (2.0, 27.9, -27.3), (342.2, -13.6, -108.6), respectively. Since these re-initialization points are very far from the putative solution, it needs a little longer time for the simulation to converge. Figure 10 shows the GOF plots resulted from these runs. For comparison, the the GOF plot resulted from the runs with the default starting region $[-4, 4] \times [-2, 2]^2$ was also included. It indicates that SAMCMC can avoid the model

degeneracy problem through re-initializations.

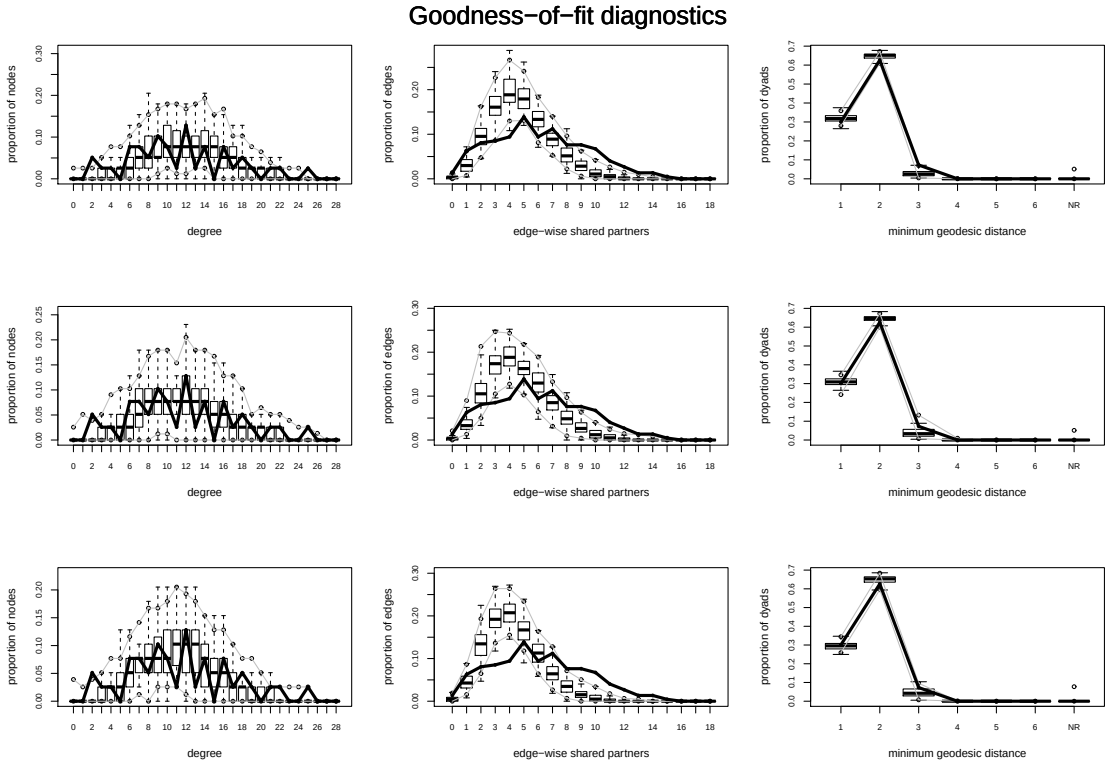

Fig. 10. Goodness-of-fit(GOF) plots for Kapferer's tailor shop network resulted from the runs with the default starting region $[-4,4]\times[-2,2]^2$ (row 1), the starting point $(-20,0,17)$ (row 2), and the starting point $(-350,0,350)$ (row 3).

As a summary of the first two examples, we conclude that SAMCMC is able to overcome the the model degeneracy problem in fitting ERGMs through its varying truncation mechanism. MCMLE and SSA can only converge to a local optimal solution near the starting point, and thus often fail to produce a reasonable estimate for ERGMs if the starting point is close to or lies in the degeneracy region. In the next two subsections, we will show that SAMCMC can work as well as or better than MCMLE and SSA for the ERGMs which do not suffer from the model degeneracy problem.

Finally, we note that MCMLE and SSA becomes significantly slower when they

suffer from the model degeneracy problem. This is mainly related to the data structure and sampling algorithm used in ERGM. There is a binary tree of edges incident on each node. As the network gains more edges, it takes longer to search through the binary tree or update it. ERGM employs the tie-no-tie (TNT) sampler (Morris *et al.*, 2008) to simulate auxiliary networks. The TNT proposal picks with equal probability a dyad with a tie or a dyad without a tie to propose a toggle. Thus it becomes significantly slower as the network grows more dense.

3. Lazega's Lawyer Network

This dataset comes from a network study of corporate law partnership [45] that was carried out in a Northeastern US corporate law firm, referred to as SG & R 1988-1991 in New England. It includes a friendship network among 36 partners of this firm. The members' attributes are also part of this dataset, including seniority, office location, gender, and their practices. The network is shown in Figure 6.

The likelihood function of the model we considered for this network is given by

$$f_2(\boldsymbol{y}|\theta) = \frac{1}{\kappa(\theta)} e^{\{\theta_1 S_1(\boldsymbol{y}) + \theta_2 M_{sen}(\boldsymbol{y}) + \theta_3 M_{pra}(\boldsymbol{y}) + \theta_4 H_{pra}(\boldsymbol{y}) + \theta_5 H_{sex}(\boldsymbol{y}) + \theta_6 H_{loc}(\boldsymbol{y}) + \theta_7 v(\boldsymbol{y}|\tau)\}}, \tag{5.9}$$

where $S_1(\boldsymbol{y})$ is the edge count, $M_{sen}(\boldsymbol{y})$ and $M_{pra}(\boldsymbol{y})$ are main effects of seniority and practice, $H_{pra}(\boldsymbol{y})$, $H_{sex}(\boldsymbol{y})$, and $H_{loc}(\boldsymbol{y})$ are uniform homophily effects of practice, gender, and location, and $v(\boldsymbol{y}|\tau)$ is a GWESP with a fixed value of $\tau = 0.778$ which is the same as that used in Koskinen (2008).

SAMCMC, MCMLE and SSA were applied to this example. The resulting estimates are summarized in Table X, and the resulting GOF plots are shown in Figure 11. The GOF plots indicate that SAMCMC produced a better fitting to the observed network than MCMLE and SSA, especially for the degree and edge-wise shared part-

Table X. Estimates produced by SAMCMC, MCMLE and SSA for Lazega's lawyer network. Standard deviations are shown in the parentheses.

	SAMCMC	MCMLE	SSA
Edge Counts (θ_1)	-6.507(9.7×10^{-4})	-6.442(7.1×10^{-3})	-6.503(2.6×10^{-2})
Main Effect			
Seniority (θ_2)	0.852(4.3×10^{-4})	0.874(5.5×10^{-3})	0.820(2.7×10^{-2})
Practice (θ_3)	0.410(2.0×10^{-4})	0.447(5.5×10^{-3})	0.393(1.7×10^{-2})
Homophily Effect			
Practice (θ_4)	0.760(2.7×10^{-4})	0.731(5.8×10^{-3})	0.733(2.0×10^{-2})
Sex (θ_5)	0.703(4.0×10^{-4})	0.668(9.6×10^{-3})	0.676(2.0×10^{-2})
Location (θ_6)	1.145(2.8×10^{-4})	1.168(9.7×10^{-3})	1.111(2.8×10^{-2})
GWESP (θ_7)	0.898(1.3×10^{-4})	0.908(1.4×10^{-2})	0.858(4.6×10^{-2})
time	2.7m	8.0m	60.2m

ners statistics. In addition, as shown in Table X, the SAMCMC estimates consistently have smaller standard deviations than the MCMLE and SSA estimates for all parameters. This implies that SAMCMC can perform stably with different initial values.

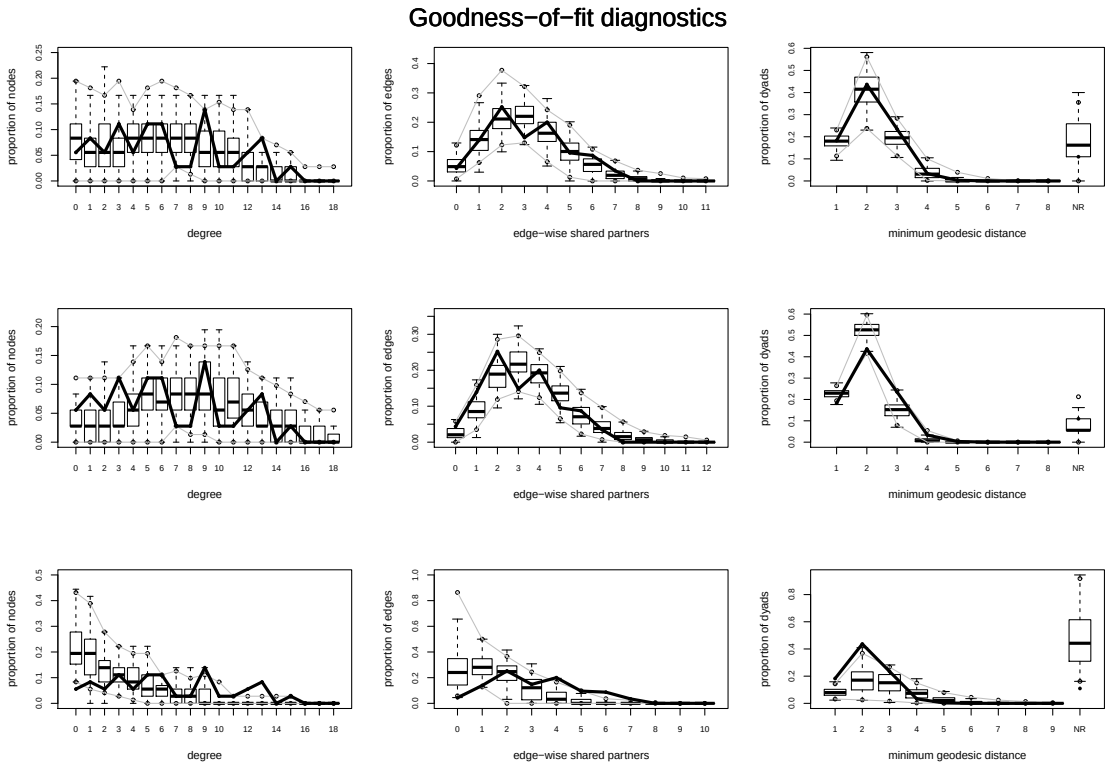

Fig. 11. Goodness-of-fit(GOF) plots for Lazega's lawyer network. Row 1: SAMCMC; Row 2: MCMLE; Row 3: SSA.

4. Zachary Karate Network

The Zachary Karate network was collected from 34 members of a university karate club, which represents the presence or absence of ties among the members of the club. [83] used this data and an information flow model of network conflict resolution to explain the split-up of this group following disputes among the members. The network is shown in Figure 6.

This network has been analyzed using an ERGM with edge counts, GWD and

GWESP with $\tau = 0.2$. The likelihood of the model is given by

$$f(\boldsymbol{y}|\theta) = \frac{1}{\kappa(\theta)} \exp\left\{\theta_1 S_1(\boldsymbol{y}) + \theta_2 u(\boldsymbol{y}|\tau) + \theta_3 v(\boldsymbol{y}|\tau)\right\},$$

where $s_1(\boldsymbol{y})$ is the edge count, $u(\boldsymbol{y}|\tau)$ is a GWD statistic with $\tau = 0.2$, and $v(\boldsymbol{y}|\tau)$ is a GWESP statistic with $\tau = 0.2$.

SAMCMC, MCMLE and SSA were applied to this network. The resulting estimates are summarized in Table XI, and the GOF plots are shown in Figure 12. The comparison indicates that all the three methods perform similarly for this network, and SAMCMC and SSA perform a little better than MCMLE. As shown in Figure 12, both SAMCMC and SSA produce better fitting for the observed edge-wise shared partners statistic than MCMLE.

Table XI. Parameter estimates produced by SAMCMC, MCMLE and SSA for the Karate network. Standard deviations are shown in the parentheses.

Method	Edge Count(θ_1)	GWD (θ_2)	GWESP (θ_3)	Time
SAMCMC	-3.730 (6.5×10^{-4})	3.725 (5.0×10^{-3})	1.303 (3.5×10^{-4})	2.5m
MCMLE	-2.909 (5.3×10^{-2})	7.901 (3.5×10^{-3})	0.361 (7.7×10^{-2})	2.9m
SSA	-3.637 (3.5×10^{-2})	3.584 (5.1×10^{-2})	1.224 (6.1×10^{-2})	22.2m

As a summary of the last two examples, we conclude that SAMCMC can work as well as or better than MCMLE and SSA for the ERGMs which do not suffer from the model degeneracy problem.

D. A Large Network Example

In this section, we considered a large network collected during the first wave (1994-1995) of National Longitudinal Study of Adolescent Health(AddHealth). The data were collected through a stratified sampling survey in the US schools containing

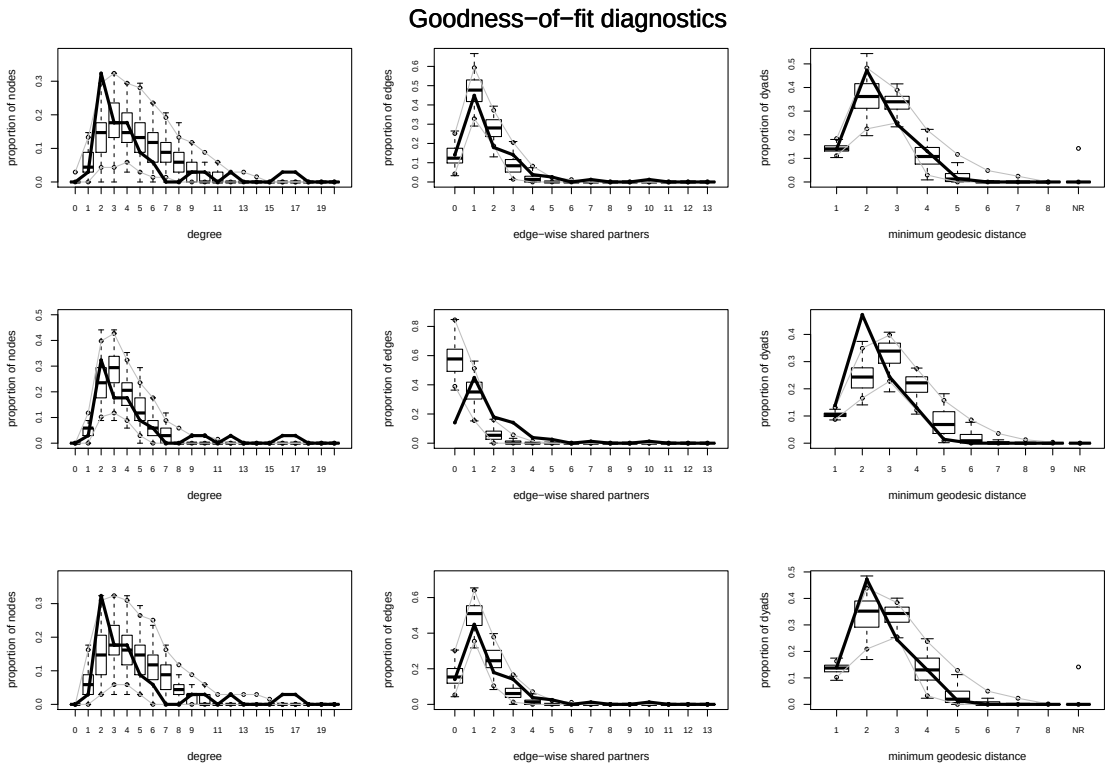

Fig. 12. Goodness-of-fit(GOF) plots for the Karate network. Row 1: SAMCMC; Row 2: MCMLE; Row 3: SSA.

grades 7 through 12. To collect the friendship, the school administrator made a roster of all students in each school and asked students to nominate five close male and female friends. Students were allowed to nominate their friends who were not in their school or not to nominate if they did not have five close male or female students. A detailed description of the dataset can be found in [62], [79], or at `http://www.cpc.unc.edu/projects/addhealth`.

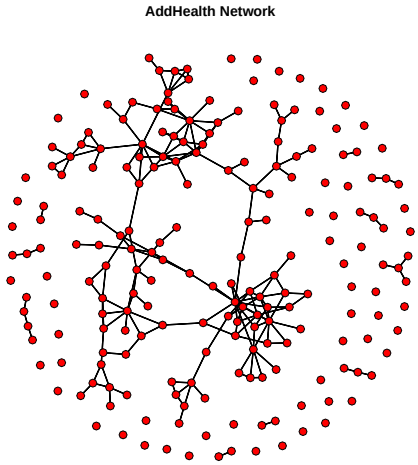

Fig. 13. A large network example: High school student friendship network.

The full dataset contains 86 schools and 90,118 students. In this paper, we analyzed a single school, school 10, which has 205 students, as shown in Figure 13. Also, we considered only the undirected network for the case of mutual friendship, although the true data is a directed network. We model the network using the following model

$$f(\boldsymbol{Y}|\theta) = \frac{1}{\kappa(\theta)} e^{\left\{\theta_1 S_1(\boldsymbol{y}) + \theta_2 u(\boldsymbol{y}|\tau) + \theta_3 w(\boldsymbol{y}|\tau) + \theta_4 v(\boldsymbol{y}|\tau) + \sum_{k=1}^{22} \sum\sum_{i<j} y_{ij} h_k(X_i, X_j)\right\}}, \qquad (5.10)$$

where $S_1(\boldsymbol{y})$ is the edge count, $u(\boldsymbol{y}|\tau)$ is a GWD statistic with $\tau = 0.25$, $w(\boldsymbol{y}|\tau)$ is a GWDSP statistic with $\tau = 0.25$, $v(\boldsymbol{y}|\tau)$ is a GWESP statistic with $\tau = 0.25$, and $\sum\sum_{i<j} y_{ij} h_k(X_i, X_j)$ are nodal covariates. There are a total of 22 nodal covariates

included in this model. Three factors were considered in modeling: Grade, Race and Sex. Grade is an ordinal variable with six levels, grade 7–grade 12. For Grade, we include nodal factor effects, differential homophily effects, and absolute different effects with $C = 1, 2, 3$. Race consists of five levels: white, black, hispanic, native American, and others. For Race, we include nodal factor effects and differential homophily effects. But we exclude the nodal factor for others whose level is 1, and the differential homophily factor for blacks and others whose value is 0. The Sex is a two-level factor: male and female. For Sex, we include the differential homophily effect or the nodal effect, but not both of them. Including both would entail redundant information. As aforementioned, sometimes we need to exclude some terms to avoid linear dependency among the model statistics. We note that model (5.10) is very similar to the model given in Hunter *et al.* (2008) except for some minor differences in covariate definition.

SAMCMC was applied to this network with the default setting given previously. SAMCMC was run 5 times, and each run costs about 12.8 hours on a 3.0GHz Intel Core 2 Duo computer. The resulting estimates are summarized in Table XII, which are calculated by averaging over five independent runs. The resulting GOF plot is shown in Figure 14, which implies that SAMCMC produces a good fit for this large network. This example indicate that SAMCMC can work for large networks.

SAMCMC costs a long CPU time for this example because the Gibbs sampler is used for generating auxiliary networks. To reduce the CPU time, we can switch the Gibbs sampler to the tie-to-tie (TNT) sampler, which selects with equal probability a dyad with a tie or a dyad without a tie to update at each MH step. Obviously, TNT can have better mixing than the Gibbs sampler for large spare networks for which the number of TNT updates can be significantly smaller than $\binom{n}{2}$, the number of MH updates made by the Gibbs sampler in a sweep.

Table XII. Estimates produced by SAMCMC for the high school student friendship network. The estimates are calculated by averaging over five independent runs with the standard deviations being given in the parentheses. NF: node factor effect, DHF: different homophily effect, UHF: uniform homophily effect, and AD: absolute different effect.

Coefficient	SAMCMC	Coefficient	SAMCMC
Edge Counts	-10.510(3.3×10^{-3})	AD(Grade 1)	-0.121(1.1×10^{-3})
GWD	0.006(1.8×10^{-4})	AD(Grade 2)	0.131(1.2×10^{-3})
GWDSP	0.007(3.6×10^{-5})	AD(Grade 3)	-0.103(1.2×10^{-3})
GWESP	1.377(7.7×10^{-5})	DHF(Grade 7)	6.005(2.7×10^{-3})
NF(Grade 8)	1.439(1.5×10^{-3})	DHF(Grade 8)	3.252(9.7×10^{-4})
NF(Grade 9)	2.183(1.9×10^{-3})	DHF(Grade 9)	1.594(1.6×10^{-3})
NF(Grade 10)	2.513(2.1×10^{-3})	DHF(Grade 10)	1.079(1.7×10^{-3})
NF(Grade 11)	2.294(1.9×10^{-3})	DHF(Grade 11)	1.869(1.3×10^{-3})
NF(Grade 12)	2.894(1.8×10^{-3})	DHF(Grade 12)	1.034(1.9×10^{-3})
NF(Race: B)	0.627(3.3×10^{-4})	DHF(Race: W)	0.682(8.1×10^{-4})
NF(Race: H)	-0.385(3.3×10^{-4})	DHF(Race: H)	0.566(4.1×10^{-4})
NF(Race: N)	-0.335(3.7×10^{-4})	DHF(Race: N)	1.052(4.9×10^{-4})
NF(Sex: F)	0.141(9.8×10^{-5})	UHF(Sex)	0.544(1.3×10^{-4})

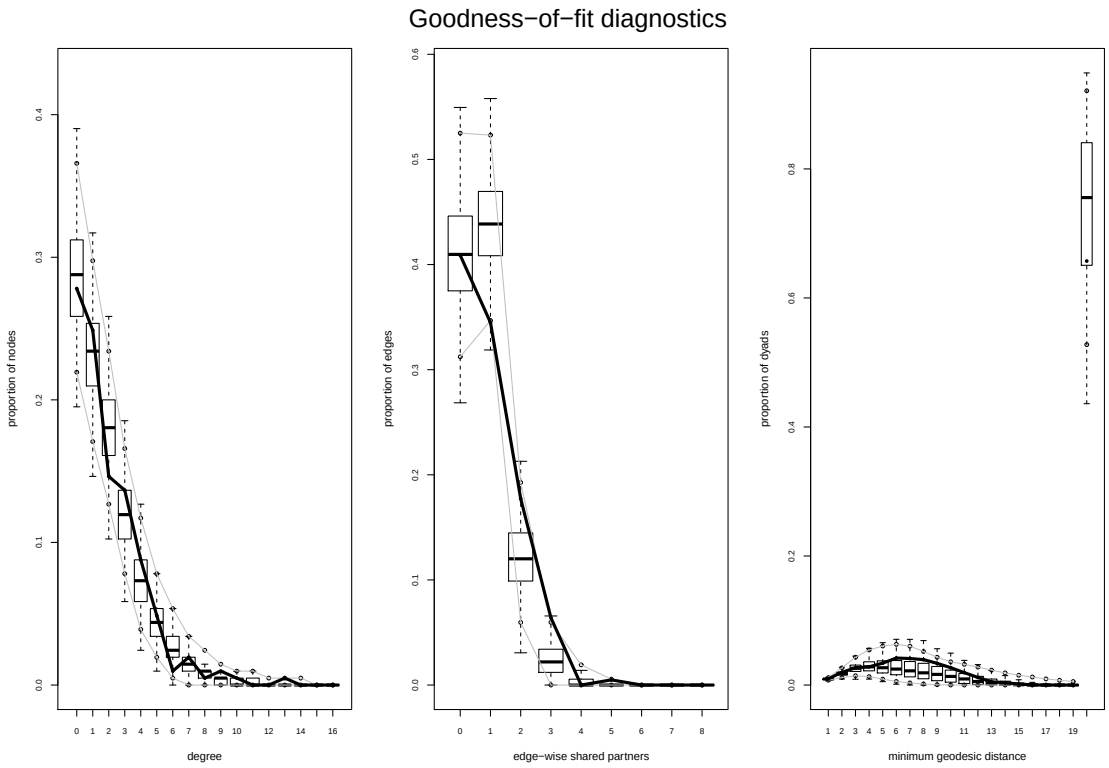

Fig. 14. Goodness-of-fit(GOF) plots for the high school student friendship network.

CHAPTER VI

CONCLUSION

In this dissertation, we have proposed two new algorithms for statistical inference for models with intractable normalizing constants: the Monte Carlo Metropolis-Hastings algorithm and the Bayesian Stochastic Approximation Monte Carlo algorithm. In addition, we has demonstrated how the SAMCMC method can be applied to estimate the parameters of ERGMs, which is one of the typical examples of statistical models with intractable normalizing constants, without the hinderance of model degeneracy.

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. Although the algorithm violates the detailed balance condition, it still converges, as shown in the paper, 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.

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. Comparing to the existing GIMH algorithm, the MCMH algorithm should be more efficient for these problems, as it recycles the auxiliary samples in simulations.

The Bayesian Stochastic Approximation Monte Carlo algorithm works by simu-

lating from a sequence of approximated distributions using the stochastic approximation Monte Carlo algorithm One significant advantage of BSAMC over the auxiliary variable MCMC methods is that it avoids the requirement for perfect samples, and thus it can be applied to many models for which perfect sampling is not available or very expensive. Although the normalizing constant approximation is also involved in BSAMC, as shown by our numerical results, BSAMC can perform very robustly to initial guesses of parameters due to the powerful ability of SAMC in sample space exploration. A strong law of large numbers has been established for BSAMC estimators under mild conditions.

BSAMC has provided a general framework for approximated Bayesian inference for the models for which the likelihood function is intractable: sampling from a sequence of approximated distributions with their average converging to the target distribution. From this point of view, MCMLE is just a special instance of BSAMC, for which there is only one approximate distribution is sampled from. Within this framework, BSAMC can also be implemented in different ways, for example, the so-called *"grid approach"*. In this approach, we choose k points of θ: $\theta_0^{(1)}, \cdots, \theta_0^{(k)}$, and wish the convex set formed by the k points covers the true value of θ. In practice, $\theta_0^{(i)}$'s can be selected around the MPLE of θ. We can define a mixture distribution

$$g(x|\theta_0^{(1)}, \ldots, \theta_0^{(k)}) = \frac{1}{k}\sum_{i=1}^{k} p(x, \theta_0^{(i)})/\kappa(\theta_0^{(i)}), \tag{6.1}$$

for which the normalizing constants $\kappa(\theta_0^{(1)}), \cdots, \kappa(\theta_0^{(k)})$ can be approximated by SAMC in an on-line manner. Then, (6.1) can replace (4.1) to work as a trial distribution for BSAMC. It is easy to see that the convergence results established still hold for the grid approach.

We note that BSAMC can be further improved using the smoothing SAMC

algorithm proposed by [47]. Smoothing SAMC includes a smoothing step at each iteration, which distributes the information contained in each sample to its neighboring subregions and thus improves the convergence of the simulation.

Varying trunction SAMCMC algorithm outperform for estimating the parameters of ERGMs. We showed that the resulting estimate is consistent, asymptotically normal and asymptotically efficient. Comparing to the MCMLE and SSA methods, a significant advantage of SAMCMC is that it overcomes the model degeneracy problem. This is remarkable. For the ERGMs which consist of basic Markovian statistics, the MCMLE and SSA methods often fail to produce any reasonable estimates due to the model degeneracy, while SAMCMC still works well. For the ERGMs which do not suffer from the model degeneracy, SAMCMC can work as well as or better than the MCMLE and SSA methods.

The strength of SAMCMC comes from its varying truncation mechanism, which enables SAMCMC to avoid the model degeneracy problem through re-initialization. MCMLE and SSA do not possess the re-initialization mechanism and tend to converge to a solution near the starting point, so they often fail for the models which suffer from the model degeneracy problem.

In addition to finding the MLE, parameters of the ERGM can also be estimated under the Bayesian framework, see e.g., [44] and [15]. It is known that the MH algorithm cannot be directly applied to sample from the posterior distribution of the ERGM, because its acceptance probability would involve an unknown normalizing constants ratio $\kappa(\theta)/\kappa(\theta')$, where θ' denotes the proposed value. To tackle this difficulty, [44] proposed to estimate this ratio using the linked importance sampler [58]. However, including a Monte Carlo estimate in the acceptance probability of the MH move would break its detailed balance condition, and thus the resulting estimate is only approximately correct, even when the number of samples used at each iteration

for estimation of $\kappa(\theta)/\kappa(\theta')$ is large. Instead of estimating the ratio $\kappa(\theta)/\kappa(\theta')$, [15] attempt to generate a perfect network $\boldsymbol{y}$ from $f(\boldsymbol{y}|\theta')$ at each iteration via a long MH run, and then to cancel the unknown ratio using the exchange technique proposed by [57]. Due to complex interdependency of social networks, perfect networks are usually difficult to be generated using the MH algorithm. Comparing to these Bayesian algorithms, SAMCMC avoids the requirement for estimation of unknown normalizing constants and the requirement for drawing perfect network samples, and thus can be easily used in practice.

REFERENCES

[1] D.F. Andrews and A.M. Herzberg, *Data*. New York: Springer, 1985.

[2] C. Andrieu, È. Moulines, and, P. Priouret, P. "Stability of stochastic approximation under verifiable conditions." *SIAM Journal of Control and Optimization*, vol. 44, pp. 283–312, 2005.

[3] C. Andrieu and G.O. Roberts, "The pseudo-marginal approach for efficient Monte Carlo computations." *Annals of Statistics*, vol. 37, pp. 697–725, 2009.

[4] Y. Bai, G.O. Roberts, and J. Rosenthal, "On the containment condition for adaptive Markov chain Monte Carlo algorithms." Ph.D. dissertation, University of Toronto, Toronto, Ontario, 2009

[5] N. Balram and J.M.F. Moura, "Noncausal Gauss Markov random fields: Parameter structure and estimation." *IEEE Transaction on Information Theory*, vol. 39, pp. 1333-1355, 1993.

[6] O.E. Barndorff-Nielsen, *Information and Exponential Families in Statistical Theory*. New York: Wiley, 1978.

[7] K. Bartz, J. Blitzstein, and J.S. Liu, "Monte Carlo maximum likelihood for exponential random graph models: From snowballs to umbrella densities." Technical Report, Department of Statistics, Harvard University, 2008.

[8] M.A. Beaumont, "Estimation of population growth or decline in genetically monitored populations." *Genetics*, vol. 164, pp. 1139-1160, 2003.

[9] J.E. Besag, "Spatial interaction and the statistical analysis of lattice systems (with Discussion)." *Journal of Royal Statistical Society, Series B*, vol. 36, pp 192-236, 1974.

[10] J.E. Besag and P.A.P. Moran, P.A.P, "On the estimation and testing of spatial interaction in Gaussian lattice processes." *Biometrika*, vol. 62, pp. 555-562, 1975.

[11] P.J. Bickel and K.A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*, vol. 1, New Jersey, Prentice Hall, 2nd edition, 2000.

[12] P. Billingsley, *Probability and Measure*. New York, Wiley, 1986.

[13] J.G. Booth and J.P. Hobert, "Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm." *Journal of Royal Statistical Society, Series B*, vol. 61, pp. 265-285, 1999.

[14] L.D. Brown, *Fundamentals of Statistical Exponential Families*. Hayward: Institute of Mathematical Statistics, CA., 1986.

[15] A. Caimo and N. Friel, "Bayesian inference for exponential random graph models." *Social Networks*, vol. 33, pp. 41-55, 2011.

[16] T.K. Chandra and A. Goswami, "Cesàro α-integrability and laws of large numbers-II." *Journal of Theoretical Probability*, vol. 19, pp. 789-816, 2005.

[17] H.F. Chen, *Stochastic Approximation and Its Applications*. Dordtrcht, The Nethelands.: Kluwer Academic Publishers, 2002.

[18] H.F. Chen and Y.M. Zhu, "Stochastic approximation procedures with randomly varying truncations." *Statistical Sinica*, vol. 29, pp. 914-926, 1986.

[19] M.-H. Chen and Q.-M. Shao, Q.-M. "On Monte Carlo methods for estimating ratios of normalizing constants." *Annals of Statistics*, vol. 25, pp. 1563-1594, 1997.

[20] M.-H. Chen, Q.-M. Shao, and J.G. Ibrahim, *Monte Carlo Methods in Bayesian Computation*. New York: Springer, 2000.

[21] A.M. Childs, R.B. Patterson, and D.J.C. MacKay, "Exact sampling from nonattractive distributions using summary states." *Physical Review E*, vol. 63, 036113, 2001.

[22] J. Corander, K. Dahmström, and P. Dahmström, "Maximum likelihood estimation for Markov graphs." Research Report 8, Department of Statistics, University of Stockholm, 1998.

[23] S.J. Cranmer and B.A. Desmarais, "Inferential network analysis with exponential random graph models." *Political Analysis*, vol. 19, pp. 66-86. 2011.

[24] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via EM algorithm." *Journal of Royal Statistical Society, Series B*, vol. 39, pp. 1-38, 1977.

[25] M.A.J. van Duijn, K.J. Gile, and M.S. Handcock, "A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models." *Social Networks*, vol. 31, pp. 52-62, 2009.

[26] B. Efron and R.J. Tibshirani, *An Introduction to the Bootstrap*. New York, Chapman & Hall, 1993

[27] I. Frank and D. Strauss, "Markov graphs." *Journal of American Statistical Association*, vol. 81, pp. 832-842, 1986.

[28] A. Gelman and X.-L. Meng, "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling." *Statistical Science*, vol. 13, pp. 163-185, 1998.

[29] A. Gelman and D.B. Rubin, "Inference from iterative simulation using multiple sequences (with Discussion)." *Statistical Sciences*, vol. 7, pp. 457-511, 1992.

[30] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721-741, 1984.

[31] C.J. Geyer, "Markov chain Monte Carlo maximum likelihood", in *Proceedings of the 23rd Symposium on the Interface: Computing Science and Statistics*. E.M. Keramigas, Ed., Fairfax, VA: Interface Foundation, 1991, pp. 156-163.

[32] C.J. Geyer and E. Thompson, "Constrained Monte Carlo maximum likelihood for dependent Data." *Journal of Royal Statistical Society, Series B*, vol. 54, pp. 657-699, 1992.

[33] M.G. Gu, and H. Zhu, "Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation." *Journal of Royal Statistical Society, Series B*, vol. 63, pp. 339-355, 2001.

[34] M. Handcock, 2003. "Statistical models for social networks: Degeneracy and inference," in *Dynamic Social Network Modeling and Analysis*. R. Breiger, K. Carley, P. Pattison, Eds., Washington, DC, National Academies Press, 2003, pp. 229-240.

[35] M.S. Handcock, D.R. Hunter, C.T. Butts, S.M. Goodreau, and M. Morris, "`statnet`: Software tools for the representation, visualization, analysis and simulation of network data." *Journal of Statistical Software*, vol. 24, 1, 2008.

[36] W. Hastings, "Monte Carlo sampling methods using Markov Chains and their applications." *Biometrika*, vol. 57, pp. 97-109, 1970.

[37] F. Huang and Y. Ogata, (2002). "Generalized pseudo-likelihood estimates for Markov random fields on lattice." *Annals of the Institute of Statistical Mathematics*, vol. 54, pp. 1-18, 2002.

[38] D. Hunter, "Curved exponential family models for social network." *Social Networks*, vol. 29, pp. 216-230, 2007.

[39] D. Hunter, S. Goodreau, and M. Handcock, "Goodness of fit of social network models." *Journal of American Statistical Association*, vol. 103, pp. 248-258, 2008.

[40] D. Hunter and M. Handcock, "Inference in curved exponential family models for network." *Journal of Computational and Graphical Studies*, vol. 15, pp. 565-583, 2006.

[41] D. Hunter, M. Handcock, C. Butts, S. Goodreau, and M. Morris, 2008, "`ergm`: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks," *Journal of Statistical Software*, vol. 24, 3, 2008.

[42] M. Hurn, , O. Husby, and H. Rue, *A tutorial on image analysis.* Lecture Notes in Statistics 173, New York: Springer, 2003, pp. 87-141.

[43] B. Kapferer, *Strategy and Transaction in an African Factory.* Manchester, UK: Manchester University Press, 1972.

[44] J.H. Koskinen, "The linked importance sampler auxiliary variable Metropolis-Hastings algorithm for distributions with intractable normalizing constants." Technical Report. University of Melbourne, 2008.

[45] E. Lazega, *The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership*. Oxford: Oxford University Press, 2001.

[46] F. Liang, "Continuous contour Monte Carlo for marginal density estimation with an application to a spatial statistical model." *Journal of Computational and Graphical Statistics*, vol. 16, pp. 608-632, 2007.

[47] F. Liang, "Improving SAMC using smoothing methods: Theory and applications to Bayesian model selection problem." *Annals of Statistics*, vol. 37, pp. 2626-2654, 2009.

[48] F. Liang, "A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants." *Journal of Statistical Computing and Simulation*, vol. 80, pp. 1007-1022, 2010.

[49] F. Liang, "Trajectory averaging for stochastic approximation MCMC algorithms." *Annals of Statistics*, vol. 38, pp. 2823-2656, 2010.

[50] F. Liang, C. Liu, and R.J. Carroll, "Stochastic approximation in Monte Carlo computation." *Journal of American Statistical Association*, vol. 102, pp. 305-320, 2007.

[51] M. Lubbers, and T.A.B. Snijders, "A comparison of various approaches to the exponential random graph model: A reanalysis of 104 student networks in school classes." *Social Networks*, vol. 29, pp. 489-507, 2007.

[52] C.E. McCulloch, S.R. Searle, and J.M Neuhaus, J.M. (2008). *Generalized, Linear, and Mixed Models*. New York, Wiley, 2nd Edition, 2008.

[53] X.-L. Meng and W.H. Wong, "Simulating ratios of normalizing constants via a simple identity: A theoretical exploration." *Statistica Sinica*, vol. 6, pp. 831-860, 1996.

[54] N. Metropolis, A. Rosenbluth, M. Rosenbluth, and A. Teller, Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, vol. 21, pp. 1087-1092, 1953.

[55] J. Møller, A.N. Pettitt, R. Reeves, and K.K. Berthelsen, "An efficient Markov chain Monte Carlo method for distributions with intractable normalizing constants." *Biometrika*, vol. 93, pp. 451-458, 2006.

[56] P. Müller, "Alternatives to the Gibbs sampling scheme." Technical report, Institute of Statistics and Decision Sciences, Duke University, 1993.

[57] I. Murray, Z. Ghahramani, and D.J.C. MacKay, " MCMC for doubly-intractable distributions." in *Proc. 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 359-366, 2006.

[58] R.M. Neal, "Estimating ratios of normalizing constants using linked importance sampling." Technical report No. 0511, Department of Statistics, University of Toronto, 2008.

[59] J.F. Padgett, *Marriage and elite structure in Renaissance Florence, 1282-1500.* Paper delivered to the Social Science History Association, 1994.

[60] H.K. Preisler, "Modeling spatial patterns of trees attacked by bark-beetles." *Applied Statistics*, vol. 42, pp. 501-514, 1993.

[61] J.G Propp and D.B. Wilson, "Exact sampling with coupled Markov chains and applications to statistical mechanics." *Random Structures and Algorithms*, vol. 9, pp. 223-252, 1996.

[62] M.D. Resnick, P.S. Bearman, R.W. Blum, K.E. Bauman, K.M. Harris, J. Jones, J. Tabor, T. Beuhring, R.E. Sieving, M. Shew, M. Ireland, L.H. Bearinger, J.R. Udry, "Protecting adolescents from harm: Findings from the national longitudinal study on adolescent health." *Journal of the American Medical Association*, vol. 278, pp. 823-832, 1997.

[63] W.B. Riggan, J.P.Creason, W.C. Nelson, K.G. Manton, M.A. Woodbury, E. Stallard, A.C.Pellom, and J.Beaubier, *U.S. Cancer Mortality Rates and Trends, 1950-1979.* (vol. IV: Maps), Washington, D.C.: U.S. Government Printing Office, 1987.

[64] H. Robbins and S. Monro, "A stochastic approximation method." *Annals of Mathematical Statistics*, vol. 22, pp. 400-407, 1951.

[65] G.O. Roberts, "Markov chain concepts related to sampling algorighms" in *Markov Chain Monte Carlo in Practice*. W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, Eds., London: Chapman & Hall/CRC, 1996, pp. 45-57.

[66] G.O. Roberts and J.S. Rosenthal, "Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms." *Journal of Applied Probability*, vol. 44, pp. 458-475, 2007.

[67] G.O. Roberts and R.L. Tweedie, "Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms." *Biometrika*, vol. 83, pp. 95-110, 1996.

[68] G. Robins, P. Pattison, Y. Kalish, and D. Lusher, "An introduction to exponential random graph (p^*) models for social networks." *Social Networks*, vol. 29, pp. 173-191, 2007.

[69] G. Robins, T. Snijers, P. Wang, M. Handcock, and P. Pattison, "Recent development in exponential random graph models for social networks." *Social Networks*, vol. 29, pp. 192-215, 2007.

[70] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*. London, Chapman & Hall, 2005.

[71] Z. Saul and V. Filkov, "Exploring biological network structure using exponential random graph models." *Bioinfomatics*, vol. 23, pp. 2604-2611, 2007.

[72] R.J. Serfling, *Approximation Theorems of Mathematical Statistics*. New York, Wiley, 1980.

[73] M. Sherman, T.V. Apanasovich, and R.J. Carroll, "On estimation in binary autologistic spatial models." *Journal of Statistical Computation and Simulation*, vol. 76, pp. 167-179, 2006.

[74] T.A.B. Snijders, "Markov chain Monte Carlo estimation of exponential random graph models." *Journal of Social Structure*, vol. 3, 2, 2002.

[75] T.A.B. Snijders, P.E. Pattison, G.L. Robins, and M.S. Handcock, "New specifications for exponential random graph models." *Sociological Methodology*, vol. 36, pp. 99-153, 2006.

[76] D. Strauss and M. Ikeda, "Pseudo-likelihood estimation for social network." *Journal of American Statistical Association*, vol. 82, pp. 204-212, 1990.

[77] L. Tierney, "Markov chains for exploring posterior distributions (with Discussion)." *Annals of Statistics*, vol. 22, pp. 1701-1762, 1994.

[78] G.M. Torrie and J.P. Valleau, "Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling." *Journal of Chemical Physics*, vol. 23, pp. 187-199, 1997.

[79] J.R. Udry, and P.S. Bearman, "New methods for new research on adolescent sexual behavior," in *New Perspectives on Adolescent Risk Behavior*. R. Jessor, Ed., New York: Cambridge University Press, 1998, pp. 241-269.

[80] S. Wasserman, and G. Robins, "An introduction to random graphs, dependence graphs, and p^*," in *Models and Methods in Social Network Analysis*. P.J. Carrington, J. Scott, and S. Wasserman, Eds., Cambridge: Cambridge University Press, 2005, Chapter 8, pp. 148-191.

[81] G. Wei and M. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithm." *Journal of American Statistical Assocation*, vol. 85, pp. 699-704, 1990.

[82] H. Wu and F.W. Huffer, "Modeling the distribution of plant species using the autologistic regression model." *Ecological Statistics*, vol. 4, pp. 49-64, 1997.

[83] W. Zachary, "An information flow model for conflict and fission in small groups." *Journal of Anthropological Research*, vol. 33, pp. 452-473, 1977.

APPENDIX A

PROOF OF THEOREM IN CHAPTER III

Proof of Lemma B.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 $A \in \mathcal{B}(\Theta)$,

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

where $\mathbb{I}(\cdot)$ is the indicator function. By condition (A_2), we deduce that the implication is 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 B.2 Let

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

and we therefore focus on the quantity

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

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

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

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

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

This completes the proof of Lemma B.2.

Proof of Theorem B.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 ($1 \le <$

$k_0 < k$),

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

For any $\epsilon > 0$, by Lemma B.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 B.2 has been applied to (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 B.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 B.2 To prove this theorem, we introduce the following lemma (Lemma 4.1 of [4]):

Lemma .1 *Consider an adaptive MCMC algorithm, on a state space $\mathcal{X}$, with adaptation index $\mathcal{Y}$, so $\pi(\cdot)$ is stationary for each kernel P_γ for $\gamma \in \mathcal{Y}$. If $\mathcal{Y}$ is finite and*

$\sum_{n=1}^{\infty} P(\Gamma_n \neq \Gamma_{n-1}) < \infty$, *then the adaptive Markov chain is ergodic.*

Since the transitional kernel of $\{\theta_t\}$ is independent of iterations (i.e., Γ_n takes a constant value in Lemma .1), the two conditions, $\mathcal{Y}$ is finite and $\sum_{n=1}^{\infty} P(\Gamma_n \neq \Gamma_{n-1}) < \infty$, trivially holds for the marginal chain $\{\theta_t\}$. Hence, the marginal chain $\{\theta_t\}$ is ergodic and has the same stationary distribution as MCMH-II.

APPENDIX B

PROOF OF THEOREM IN CHAPTER IV

Before giving details of the drift condition, we first define some notations. Let $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(m)})$ denote a vector of m MCMC samples drawn from a distribution defined on the space $\mathcal{X}$. Let $\mathcal{X}^m = \mathcal{X} \times \cdots \times \mathcal{X}$, and thus $\boldsymbol{x} \in \mathcal{X}^m$. Let $\mathcal{B}_{\mathcal{X}^m}$ denote the Borel set defined on $\mathcal{X}^m$. Let $P_\xi(\boldsymbol{x}, \boldsymbol{y})$ denote a Markov transition kernel indexed by ξ (which takes values in the space Ξ).

For a function $g : \mathcal{X}^m \to \mathbb{R}^d$, define the norm

$$\|g\|_V = \sup_{\boldsymbol{x} \in \mathcal{X}^m} \frac{\|g(\boldsymbol{x})\|}{V(\boldsymbol{x})},$$

and define the set $\mathcal{L}_V = \{g : \mathcal{X}^m \to \mathbb{R}^d, \sup_{\boldsymbol{x} \in \mathcal{X}^m} \|g\|_V < \infty\}$.

Given the above notations, the drift condition can be specified as follows:

For any $\xi \in \Xi$, the transition kernel P_ξ is irreducible and aperiodic. In addition, there exists a function $V : \mathcal{X}^m \to [1, \infty)$ and a constant $\alpha \geq 2$ such that for any compact subset $\mathcal{K} \subset \Xi$,

(i) There exist a set $\boldsymbol{C} \subset \mathcal{X}^m$, an integer l, constants $0 < \lambda < 1$, b, ς, $\delta > 0$ and a probability measure ν such that

- $$\sup_{\xi \in \mathcal{K}} P_\xi^l V^\alpha(\boldsymbol{x}) \leq \lambda V^\alpha(\boldsymbol{x}) + bI(\boldsymbol{x} \in \boldsymbol{C}), \quad \forall \boldsymbol{x} \in \mathcal{X}^m. \tag{B.1}$$
- $$\sup_{\xi \in \mathcal{K}} P_\xi V^\alpha(\boldsymbol{x}) \leq \varsigma V^\alpha(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{X}^m. \tag{B.2}$$
- $$\inf_{\xi \in \mathcal{K}} P_\xi^l(\boldsymbol{x}, A) \geq \delta \nu(A), \quad \forall \boldsymbol{x} \in \boldsymbol{C},\ \forall A \in \mathcal{B}_{\mathcal{X}^m}. \tag{B.3}$$

(ii) There exists a constant $c > 0$ such that, for all $\boldsymbol{x} \in \mathcal{X}^m$,

- $\sup_{\xi \in \mathcal{K}} \|H_\xi(\cdot)\|_V \leq c.$ (B.4)
- $\sup_{(\xi,\xi') \in \mathcal{K}} \|H_\xi(\cdot) - H_{\xi'}(\cdot)\|_V \leq c\|\xi - \xi'\|.$ (B.5)

(iii) There exists a constant $c > 0$ such that, for all $(\xi, \xi') \in \mathcal{K} \times \mathcal{K}$,

- $\|P_\xi g - P_{\xi'} g\|_V \leq c\|g\|_V \|\xi - \xi'\|, \quad \forall g \in \mathcal{L}_V.$ (B.6)
- $\|P_\xi g - P_{\xi'} g\|_{V^\alpha} \leq c\|g\|_{V^\alpha} \|\xi - \xi'\|, \quad \forall g \in \mathcal{L}_{V^\alpha}.$ (B.7)

APPENDIX C

PROOF OF THEOREM IN CHAPTER V

These conditions, given in [49], are necessary to analyze asymptotic efficiency of $\bar{\theta}_k$.

Lyapunov condition on $h(\theta)$. This condition assumes the existence of a global Lyapunov function v for the mean field h.

(A_1) Let $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ denote the Euclidean inner product. Θ is an open set, the function $h : \theta \to \mathbb{R}^d$ is continuous, and there exists a continuous differentiable function $v : \Theta \to [0, \infty)$ such that

(i) There exists $M_0 > 0$ such that

$$\mathcal{L} = \{\theta \in \Theta, \langle \nabla v(\theta), h(\theta) \rangle = 0\} \subset \{\theta \in \Theta, v(\theta) < M_0\}. \tag{C.1}$$

(ii) There exists $M_1 \in (M_0, \infty)$ such that $\mathcal{V}_{M_1}$ is a compact set, where $\mathcal{V}_M = \{\theta \in \Theta, v(\theta) \leq M\}$.

(iii) For any $\theta \in \Theta \setminus \mathcal{L}$, $\langle \nabla v(\theta), h(\theta) \rangle < 0$.

(iv) The closure of $v(\mathcal{L})$ has an empty interior.

Stability Condition on $h(\theta)$. This condition constraints the behavior of the mean field function around the solution points. It makes the trajectory averaging estimator sensible both theoretically and practically.

(A_2) The mean field function $h(\theta)$ is measurable and locally bounded. That is, there exists a stable matrix F (i.e., all eigenvalue of F are with negative real parts),

$\gamma > 0$, $\rho \in (0, 1]$, and a constant c such that, for any $\theta^* \in \mathcal{L}$,

$$||h(\theta) - F(\theta - \theta^*)|| \leq c||\theta - \theta^*||^{1+\rho}, \quad \forall \theta \in \{\theta : ||\theta - \theta^*|| \leq \gamma\},$$

where $\mathcal{L}$ is defined in (C.1).

Drift condition on the transition kernel P_θ. Before giving details of this condition, we first define some terms and notation. Assume that a transition kernel P_θ is irreducible, aperiodic and has a stationary distribution on a sample space denoted by $\mathcal{X}$. A set $\boldsymbol{C} \subset \mathcal{X}$ is said to be small if there exist a probability measure ν on $\mathcal{X}$, a positive integer l and $\delta > 0$ such that

$$P_\theta^l(\boldsymbol{x}, A) \geq \delta\nu(A), \quad \forall \boldsymbol{x} \in \boldsymbol{C},\ \forall A \in \mathcal{B}_\mathcal{X},$$

where $\mathcal{B}_\mathcal{X}$ is the Borel set defined on $\mathcal{X}$. A function $V : \mathcal{X} \to [1, \infty)$ is said to be a drift function outside $\boldsymbol{C}$ if there exist positive constants $\lambda < 1$ and b such that

$$P_\theta V(\boldsymbol{x}) \leq \lambda V(\boldsymbol{x}) + bI(\boldsymbol{x} \in \boldsymbol{C}), \quad \forall \boldsymbol{x} \in \mathcal{X},$$

where $P_\theta V(\boldsymbol{x}) = \int_\mathcal{X} P_\theta(\boldsymbol{x}, \boldsymbol{y})V(\boldsymbol{y})d\boldsymbol{y}$. For a function $g : \mathcal{X} \to \mathbb{R}^d$, define the norm

$$\|g\|_V = \sup_{\boldsymbol{x} \in \mathcal{X}} \frac{\|g(\boldsymbol{x})\|}{V(\boldsymbol{x})},$$

and define the set $\mathcal{L}_V = \{g : \mathcal{X} \to \mathbb{R}^d, \sup_{\boldsymbol{x} \in \mathcal{X}} \|g\|_V < \infty\}$. Given these terms and notation, the drift condition can be specified as follows.

(A_3) For any given $\theta \in \Theta$, the transition kernel P_θ is irreducible and aperiodic. In addition, there exists a function $V : \mathcal{X} \to [1, \infty)$ and a constant $\alpha \geq 2$ such that for any compact subset $\mathcal{K} \subset \Theta$,

(i) There exist a set $\boldsymbol{C} \subset \mathcal{X}$, an integer l, constants $0 < \lambda < 1$, b, ς, $\delta > 0$

and a probability measure ν such that

- $\sup_{\theta\in\mathcal{K}} P_\theta^l V^\alpha(\boldsymbol{x}) \le \lambda V^\alpha(\boldsymbol{x}) + bI(\boldsymbol{x}\in\boldsymbol{C}), \quad \forall \boldsymbol{x}\in\mathcal{X}.$ (C.2)
- $\sup_{\theta\in\mathcal{K}} P_\theta V^\alpha(\boldsymbol{x}) \le \varsigma V^\alpha(\boldsymbol{x}), \quad \forall \boldsymbol{x}\in\mathcal{X}.$ (C.3)
- $\inf_{\theta\in\mathcal{K}} P_\theta^l(\boldsymbol{x}, A) \ge \delta\nu(A), \quad \forall \boldsymbol{x}\in\boldsymbol{C},\ \forall A\in\mathcal{B}_\mathcal{X}.$ (C.4)

(ii) There exists a constant $c > 0$ such that, for all $x \in \mathcal{X}$,

- $\sup_{\theta\in\mathcal{K}} \|H(\theta, \boldsymbol{x})\|_V \le c.$ (C.5)
- $\sup_{(\theta,\theta')\in\mathcal{K}} \|H(\theta, \boldsymbol{x}) - H(\theta', \boldsymbol{x})\|_V \le c\|\theta - \theta'\|.$ (C.6)

(iii) There exists a constant $c > 0$ such that, for all $(\theta, \theta') \in \mathcal{K} \times \mathcal{K}$,

- $\|P_\theta g - P_{\theta'} g\|_V \le c\|g\|_V\|\theta - \theta'\|, \quad \forall g \in \mathcal{L}_V.$ (C.7)
- $\|P_\theta g - P_{\theta'} g\|_{V^\alpha} \le c\|g\|_{V^\alpha}\|\theta - \theta'\|, \quad \forall g \in \mathcal{L}_{V^\alpha}.$ (C.8)

Conditions on the step-size This condition gives constraints to gain factors and control the speed and accuracy of convergence in trajectory averaging.

(A_4) Let $\{a_k\}$ and $\{b_k\}$ be two monotone, non-increasing, and positive sequences which satisfy the following conditions:

$$\sum_{k=1}^{\infty} a_k = \infty, \quad \lim_{k\to\infty}(ka_k) = \infty, \quad \frac{a_{k+1} - a_k}{a_k} = o(a_{k+1}), \quad , b_k = O(a_k^{\frac{1+\tau}{2}}). \tag{C.9}$$

for some $\tau \in (0, 1]$,

$$\sum_{k=1}^{\infty} \frac{a_k^{(1+\tau)/2}}{\sqrt{k}} < \infty \tag{C.10}$$

and for some constants $\delta \ge 2$

$$\sum_{i=1}^{\infty} \{a_i b_i + (b_i^{-1} a_i)^\alpha\} < \infty. \tag{C.11}$$

For instance, $a_k = C_1/k^\eta$ for some constants $C_1 > 0$ and $\eta \in (\frac{1}{2}, 1)$, then we can set $b_k = C_2/k^\xi$ for some constants $C_2 > 0$ and $\xi \in (\frac{1}{2}, \eta - \frac{1}{\alpha})$, which satisfies (C.9) and (C.11). Under this setting, the existence of τ is obvious.

Theorem .1 *([49]; Theorems 2.1–2.3) Assume the conditions* (A_1), (A_2), (A_3), *and* (A_4) *hold. Let* $\mathcal{X}_0 \subset \mathcal{X}$ *be such that* $\sup_{\boldsymbol{x} \in \mathcal{X}_0} V(\boldsymbol{x}) < \infty$ *and that* $\mathcal{K}_0 \subset \mathcal{V}_{M_0}$, *where* $\mathcal{V}_{M_0}$ *is defined in* (A_1). *Then, as* $n \to \infty$, *we have*

1. *(Convergence)* $\theta^{(n)} \to \theta^*$ *almost surely for some point* $\theta^* \in \mathcal{L}$.
2. *(Asymptotic Normality)*

$$\sqrt{n}(\bar{\theta}_n - \theta^*) \longrightarrow N(\mathbf{0}, \Gamma),$$

   where Γ *is a negative definite matrix.*

3. *(Asymptotic Efficiency)* $\bar{\theta}_n$ *is asymptotically efficient.*

Proof of Theorem .1 To prove Theorem .1, it suffices to verify Conditions (A_1), (A_2), (A_3) and (A_4) for ERGMs. Let $l(\theta) = \log f_\theta(\boldsymbol{x})$ denote the log-likelihood function of an ERGM $\boldsymbol{x}$, where $f_\theta(\boldsymbol{x})$ is specified in Equation (2.1). Let $h(\theta) = \partial_\theta l(\theta)$ denote the partial derivative of $l(\theta)$ with respect to θ. Since $\mathcal{X}$ is finite for ERGMs, we set $\mathcal{X}_0 = \mathcal{X}$ and $V(\boldsymbol{x}) = 1$. Then the conditions (A_1)–(A_4) can be verified as follows.

(A_1) It is clear that the function $h(\theta)$ is continuous in θ, as the ERGM belongs to the exponential family. Set $v(\theta) = -l(\theta) + C$, where C is a constant chosen such that $v(\theta) > 0$. Existence of C is apparent, as $l(\theta)$ is up bounded. From Equation (2.1), it can be seen that $v(\theta)$ is continuously differentiable. Thus, we have

$$\langle \nabla v(\theta), h(\theta) \rangle = -\|\partial_\theta l(\theta)\|^2,$$

which implies that the set $\mathcal{L} = \{\theta : \langle \nabla v(\theta), h(\theta) \rangle = 0\}$ coincides with the solution set $\{\theta : \partial_\theta l(\theta) = 0\}$, and that $\langle \nabla v(\theta), h(\theta) \rangle < 0$ for any $\theta \in \Theta \setminus \mathcal{L}$. This verifies (A_1)-(iii). Given the condition (C_2), the verification of other conditions of (A_1) is straightforward.

(A_2) Set the matrix F as the Hessian matrix of $l(\theta)$, then (A_2) can be verified using the Taylor expansion by choosing $\rho = 1$.

(A_3) Theorem 2.2 of Roberts and Tweedie (1996) shows that if the target distribution is bounded away from 0 and ∞ on every compact set of its support $\mathcal{X}$, then the MH chain with a proposal distribution satisfying the condition (5.6) is irreducible and aperiodic, and every nonempty compact set is small. For ERGMs, $\mathcal{X}$ is finite, so $f(\boldsymbol{x}|\theta)$ is bounded away from 0 and ∞ for any θ. In addition, the Gibbs sampler we used in generating auxiliary networks satisfies the condition (5.6). Hence, P_θ is irreducible and aperiodic for any $\theta \in \Theta$.

Since $\mathcal{X}$ is compact (finite), $\mathcal{X}$ is a small set and thus the minorisation condition is satisfied; that is, there exists an integer l such that

$$\inf_{\theta \in \Theta} P_\theta^l(\boldsymbol{x}, A) \geq \delta \nu(A), \quad \forall \boldsymbol{x} \in \mathcal{X}, \forall A \in \mathcal{B}. \tag{C.12}$$

Define $P_\theta V(\boldsymbol{x}) = \int_{\mathcal{X}} P_\theta(\boldsymbol{x}, \boldsymbol{y}) V(\boldsymbol{y}) d\boldsymbol{y}$. Since $C = \mathcal{X}$ is small, the following conditions hold

$$\begin{aligned} &\sup_{\theta \in \mathcal{K}} P_\theta^l V^\alpha(\boldsymbol{x}) \leq \lambda V^\alpha(\boldsymbol{x}) + bI(\boldsymbol{x} \in C), \quad \forall \boldsymbol{x} \in \mathcal{X}, \\ &\sup_{\theta \in \mathcal{K}} P_\theta V^\alpha(\boldsymbol{x}) \leq \kappa V^\alpha(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{X}, \end{aligned} \tag{C.13}$$

by choosing the drift function $V(\boldsymbol{x}) = 1$, $0 < \lambda < 1$, $b = 1 - \lambda$, $\kappa > 1$, $\alpha \in [2, \infty)$ and any integer l. Equations (C.12) and (C.13) implies that (A_3)-(i) is satisfied.

By construction of our algorithm, we have

$$H(\theta, \boldsymbol{Y}) = \boldsymbol{S}(\boldsymbol{Y}) - \boldsymbol{S}(\boldsymbol{y}_{obs}). \tag{C.14}$$

Since $\mathcal{X}$ is finite, there exists a constant c such that $\sup_{\theta \in \mathcal{K}} \|H(\theta, \boldsymbol{Y})\|_V \leq c$ with respect to the norm $V(\cdot) = 1$. By (C.14), we have

$$H(\theta, \boldsymbol{Y}) - H(\theta', \boldsymbol{Y}) = 0.$$

which implies that (A_3)-(ii) is satisfied.

Let $s_\theta(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{x}, \boldsymbol{y}) \min\{1, r(\theta, \boldsymbol{x}, \boldsymbol{y})\}$, where $r(\theta, \boldsymbol{x}, \boldsymbol{y}) = \frac{f_\theta(\boldsymbol{y}) q(\boldsymbol{y}, \boldsymbol{x})}{f_\theta(\boldsymbol{x}) q(\boldsymbol{x}, \boldsymbol{y})}$. Thus, we have

$$\begin{aligned} \Big|\frac{\partial s_\theta(\boldsymbol{x}, \boldsymbol{y})}{\partial \theta_i}\Big| &= \big|q(\boldsymbol{x}, \boldsymbol{y}) I(r(\theta, \boldsymbol{x}, \boldsymbol{y}) < 1) r(\theta, \boldsymbol{x}, \boldsymbol{y})[\boldsymbol{S}(\boldsymbol{y}) - \boldsymbol{S}(\boldsymbol{x})]\big| \\ &\leq q(\boldsymbol{x}, \boldsymbol{y}) \|\boldsymbol{S}(\boldsymbol{y}) - \boldsymbol{S}(\boldsymbol{x})\|, \end{aligned}$$

where $I(\cdot)$ is the indicator function. By the boundedness of the term $\|\boldsymbol{S}(\boldsymbol{y}) - \boldsymbol{S}(\boldsymbol{x})\|$ and the mean-value theorem, there exists a constant c_2 such that

$$|s_\theta(\boldsymbol{x}, \boldsymbol{y}) - s_{\theta'}(\boldsymbol{x}, \boldsymbol{y})| \leq c_2 q(\boldsymbol{x}, \boldsymbol{y}) |\theta - \theta'|, \tag{C.15}$$

which implies that

$$\sup_{\boldsymbol{x}} \|s_\theta(\boldsymbol{x}, \cdot) - s_{\theta'}(\boldsymbol{x}, \cdot)\|_1 = \sup_{\boldsymbol{x}} \int_{\mathcal{X}} |s_\theta(\boldsymbol{x}, \boldsymbol{y}) - s_{\theta'}(\boldsymbol{x}, \boldsymbol{y})| d\boldsymbol{y} \leq c_2 |\theta - \theta'|. \tag{C.16}$$

In addition, for any measurable set $A \subset \mathcal{X}$ we have

$$\begin{aligned}
&|P_\theta(\boldsymbol{x}, A) - P_{\theta'}(\boldsymbol{x}, A)| \\
&= \Big| \int_A [s_\theta(\boldsymbol{x}, \boldsymbol{y}) - s_{\theta'}(\boldsymbol{x}, \boldsymbol{y})] d\boldsymbol{y} + I(\boldsymbol{x} \in A) \int_{\mathcal{X}} [s_{\theta'}(\boldsymbol{x}, \boldsymbol{z}) - s_\theta(\boldsymbol{x}, \boldsymbol{z})] d\boldsymbol{z} \Big| \\
&\leq \int_{\mathcal{X}} |s_\theta(\boldsymbol{x}, \boldsymbol{y}) - s_{\theta'}(\boldsymbol{x}, \boldsymbol{y})| d\boldsymbol{y} + I(\boldsymbol{x} \in A) \int_{\mathcal{X}} |s_{\theta'}(\boldsymbol{x}, \boldsymbol{z}) - s_\theta(\boldsymbol{x}, \boldsymbol{z})| d\boldsymbol{z} \\
&\leq 2 \int_{\mathcal{X}} |s_\theta(\boldsymbol{x}, \boldsymbol{y}) - s_{\theta'}(\boldsymbol{x}, \boldsymbol{y})| d\boldsymbol{y} \leq 2c_2 |\theta - \theta'|.
\end{aligned} \tag{C.17}$$

For $g : \mathcal{X} \to \mathbb{R}^d$, define the norm $\|g\|_V = \sup_{\boldsymbol{x} \in \mathcal{X}} \frac{|g(\boldsymbol{x})|}{V(\boldsymbol{x})}$. Then, for any function $g \in \mathcal{L}_V = \{g : \mathcal{X} \to \mathbb{R}^d, \|g\|_V < \infty\}$, we have

$$\begin{aligned}
&\|P_\theta g - P_{\theta'} g\|_V = \| \int (P_\theta(\boldsymbol{x}, d\boldsymbol{y}) - P_{\theta'}(\boldsymbol{x}, d\boldsymbol{y})) g(\boldsymbol{y}) \|_V \\
&= \| \int_{\mathcal{X}^+} (P_\theta(\boldsymbol{x}, d\boldsymbol{y}) - P_{\theta'}(\boldsymbol{x}, d\boldsymbol{y})) g(\boldsymbol{y}) + \int_{\mathcal{X}^-} (P_\theta(\boldsymbol{x}, d\boldsymbol{y}) - P_{\theta'}(\boldsymbol{x}, d\boldsymbol{y})) g(\boldsymbol{y}) \|_V \\
&\leq \Big\| \max \Big\{ \int_{\mathcal{X}^+} (P_\theta(\boldsymbol{x}, d\boldsymbol{y}) - P_{\theta'}(\boldsymbol{x}, d\boldsymbol{y})) g(\boldsymbol{y}), \\
&\quad - \int_{\mathcal{X}^-} (P_\theta(\boldsymbol{x}, d\boldsymbol{y}) - P_{\theta'}(\boldsymbol{x}, d\boldsymbol{y})) g(\boldsymbol{y}) \Big\} \Big\|_V \\
&\leq \|g\|_V \max\{|P_\theta(\boldsymbol{x}, \mathcal{X}^+) - P_{\theta'}(\boldsymbol{x}, \mathcal{X}^+)|, |P_\theta(\boldsymbol{x}, \mathcal{X}^-) - P_{\theta'}(\boldsymbol{x}, \mathcal{X}^-)|\} \\
&\leq 2c_2 \|g\|_V |\theta - \theta'|, \qquad \text{(following from (C.17))}
\end{aligned}$$

where $\mathcal{X}^+ = \{\boldsymbol{y} : \boldsymbol{y} \in \mathcal{X}, (P_\theta(\boldsymbol{x}, d\boldsymbol{y}) - P_{\theta'}(\boldsymbol{x}, d\boldsymbol{y})) g(\boldsymbol{y}) > 0\}$ and $\mathcal{X}^- = \mathcal{X} \setminus \mathcal{X}^+$. This implies that condition (A_3)-(iii) is satisfied by choosing $V(\boldsymbol{x}) = 1$ and $\beta = 1$.

(A_4) It is easy to see that (C_1) implies (A_4) by letting $\alpha = \infty$, where α is defined in (A_3).

The proof is completed.

VITA

Ick Hoon Jin received a Bachelor of Arts degree in business and applied statistics from Yonsei University, Korea in 2004. He received a Master of Arts degree in applied statistics from Yonsei University, Korea under the direction of Dr. Chul-Eung Kim in 2006. He was admitted to the Ph.D. program in the Department of Statistics, Texas A&M University in August, 2006, and studied Markov chain Monte Carlo methods under the guidance of Dr. Faming Liang. He received a Doctor of Philosophy degree in statistics from Texas A&M University in College Station, Texas, in August, 2011.

Ick Hoon is married to Youn Sil Hur. His address is Department of Statistics, Texas A&M University, TAMU 3143, College Station, TX 77843-3143. USA