Monte Carlo dynamically weighted importance sampling for spatial models with intractable normalizing constants

Faming Liang
Department of Statistics, Texas A&M University, College Station, TX 77843-3143, USA
E-mail: fliang@stat.tamu.edu

Sooyoung Cheon
KU Industry-Academy Cooperation Group Team of Economics and Statistics, Korea University, Jochiwon 339-700, Korea.
E-mail: stat-csy@hanmail.net

Abstract. The problem of simulating from distributions with intractable normalizing constants has received much attention in the recent literature. In this paper, we propose a new MCMC algorithm, the so-called Monte Carlo dynamically weighted importance sampler, for tackling this problem. The new algorithm is illustrated with the spatial autologistic models. The novelty of our algorithm is that it allows for the use of Monte Carlo estimates in MCMC simulations, while still leaving the target distribution invariant under the criterion of dynamically weighted importance sampling. Unlike the auxiliary variable MCMC algorithms, the new algorithm removes the need of perfect sampling, and thus can be applied to a wide range of problems for which perfect sampling is not available or very expensive. The new algorithm can also be used for simulating from the incomplete posterior distribution for the missing data problem.

1. Introduction
Spatial models, e.g., the autologistic model, the Potts model, and the autonormal model [1], have been used in modeling of many scientific problems. Examples include image analysis [2], disease mapping [3], genetic analysis [4], among others. A major problem with these models is that the normalizing constant is intractable. The problem can be described as follows. Suppose we have a dataset $X$ generated from a statistical model with the likelihood function

$$f(x|\theta) = \frac{p(x, \theta)}{Z(\theta)}, \quad x \in \mathcal{X}, \quad \theta \in \Theta,$$

(1)

where $\theta$ is the parameter, and $Z(\theta)$ is the normalizing constant which depends on $\theta$ and is not available in closed form. Let $\pi(\theta)$ denote the prior density of $\theta$. The posterior distribution of $\theta$ given $X = x$ is then given by

$$\pi(\theta|x) \propto \frac{1}{Z(\theta)}p(x, \theta)\pi(\theta).$$

(2)
Since the closed form of $Z(\theta)$ is not available, inference for the model poses a great challenge on the current statistical methods.

The Metropolis-Hastings (MH) algorithm [5, 6] cannot be directly applied to simulate from $\pi(\theta|x)$, because the acceptance probability would involve the unknown ratio $Z(\theta)/Z(\theta')$, where $\theta'$ denotes the proposed value. To circumvent this difficulty, various methods have been proposed in the literature. These methods can be roughly divided into two categories, namely, the likelihood approximation-based methods and the auxiliary variable MCMC methods. They are described, respectively, as follows.

- **The likelihood approximation-based methods.** A very first work of this category is the pseudo-likelihood method [1], which approximates the likelihood function by a tractable pseudo-likelihood function. This method is easy to use, but it typically performs less well for the models for which neighboring dependence is strong. Later, Geyer and Thompson [7] proposed to approximate $Z(\theta)$ and thus the likelihood function using Monte Carlo samples generated from a distribution $f(x|\theta_0)$ with a fixed value of $\theta_0$. This approach was refined in [8], where the likelihood is approximated using the Monte Carlo samples generated from a mixture distribution defined on a lattice of points of $\theta$. Liang [9] proposed an alternative Monte Carlo approach to approximate $Z(\theta)$, where $Z(\theta)$ is viewed as a marginal distribution of the unnormalized distribution $p(x, \theta)$ and is estimated by an adaptive kernel density estimator using Monte Carlo draws. A very recent work belonging to this category is the double MH algorithm [10], which does not approximate the likelihood directly, but uses the samples generated with a finite number of MH moves to substitute for perfect samples of $f(x|\theta')$ in a MH update of $\theta$.

- **Auxiliary variable MCMC methods.** The algorithms belonging to in this category include the Møller algorithm [11] and the exchange algorithm [12], which work by including some auxiliary variables in state transitions such that the normalizing constant can be canceled. A serious weakness of the two algorithms is that they require the auxiliary variables to be generated by a perfect sampler [13]. Unfortunately, perfect sampling is very expensive or impossible for many statistical models whose normalizing constant is intractable.

In this paper, we propose a new MCMC algorithm, the so-called Monte Carlo dynamically weighted importance sampler (MCDWIS), for simulating from the distributions with intractable normalizing constants. The MCDWIS removes the need of perfect sampling, and thus can be applied to many statistical models for which perfect sampling is not available or very expensive.

The MCDWIS is a Monte Carlo version of the dynamically weighted importance sampling (DWIS) algorithm [14]. As in DWIS, the state space of the Markov chain is augmented to a population, a collection of weighted samples $(\theta, w) = \{\theta_1, w_1; \ldots; \theta_n, w_n\}$, where $n$ is called the population size. With a slight abuse of notation, $(\theta_i, w_i)$ is called an individual state of the population $(\theta, w)$. Given the current population $(\theta_i, w_i)$, the population at the $t$th iteration, one iteration of the MCDWIS, as illustrated in Figure 1, involves two steps (described in detail later):

1. **Monte Carlo Dynamic weighting (MCDW):** Update each individual state of the current population by a MCDW transition.
2. **Population control:** Split or replicate the individual states with large weights and discard the individual states with small weights.

The MCDW step allows for the use of Monte Carlo estimates in MCMC simulations. The bias induced thereby is counterbalanced by giving different weights to the new samples produced. Hence, a Monte Carlo estimate of $Z(\theta)/Z(\theta')$ can be incorporated into the MCDWIS simulations of $\pi(\theta|x)$, while leaving $\pi(\theta|x)$ invariant with respect to the dynamic importance weights. Note that the conventional MCMC algorithms, e.g., the MH algorithm, do not allow for the use
of Monte Carlo estimates in simulations. Otherwise, the detailed balance condition will be violated. Inclusion of the population control step is also quite crucial to the MCDWIS, which has the weights and population size controlled to suitable ranges. Otherwise, the MCDWIS will perform like a multi-chain dynamic weighting algorithm, and the resulting estimates would be highly variable as in a single chain dynamic weighting algorithm. Refer to [15,16] for detailed descriptions and theoretical properties of the dynamic weighting algorithm.

The remainder of this paper is organized as follows. In Section 2, we provide a brief review for the theory of dynamically weighted importance sampling. In Section 3, we describe the MCDWIS and justify it with the theory introduced in Section 2. In Section 4, we illustrate the MCDWIS with spatial autologistic models. In Section 5, we conclude the paper with a brief discussion.

2. A theory of dynamically weighted importance sampling

In this section, we provide a brief review for the theory of dynamically weighted importance sampling [14]. Let \( g_t(\theta, w) \) denote the joint density of \((\theta, w)\), an individual state of \((\theta_t, w_t)\), and let \( \psi(\theta) \) denote the target distribution we are working with. Dynamically weighted importance sampling differs from conventional importance sampling [17] in that for any given \( \theta \), the weight \( w \) is a random variable instead of a constant defined as the ratio of the true and trial densities at \( \theta \).

**Definition 2.1** The distribution \( g_t(\theta, w) \) defined on \( \Theta \times (0, \infty) \) is called correctly weighted with
respect to $\psi(\theta)$ if the following conditions hold,
\[
\int w g_t(\theta, w) dw = c_\theta \psi(\theta),
\]
and
\[
\int_{\mathcal{A}} c_\theta \psi(\theta) d\theta = \int_{\Theta} \psi(\theta) d\theta,
\]
where $\mathcal{A}$ is any Borel set, $\mathcal{A} \subseteq \Theta$.

**Definition 2.2** If $g_t(\theta, w)$ is correctly weighted with respect to $\psi(\theta)$, and samples $(\theta_{t,i}, w_{t,i})$ are simulated from $g_t(\theta, w)$ for $i = 1, 2, \ldots, n_t$, then $(\theta_t, w_t) = (\theta_{t,1}, w_{t,1}; \ldots; \theta_{t,n_t}, w_{t,n_t})$ is called a correctly weighted population with respect to $\psi(\theta)$.

Let $(\theta_t, w_t)$ be a correctly weighted population with respect to $\psi(\theta)$, and let $\theta'_1, \ldots, \theta'_m$ be distinct states in $\theta_t$. Generate a random variable/vector $\vartheta$ such that
\[
P\{\vartheta = \theta'_i\} = \frac{\sum_{j=1}^{n_t} w_j I(\theta_j = \theta'_i)}{\sum_{j=1}^{n_t} w_j}, \quad i = 1, 2, \ldots, m,
\]
where $I(\cdot)$ is an indicator function. Then $\vartheta$ is approximately distributed as $\psi(\cdot)$ if $n_t$ is large. This can be expressed in the following theorem:

**Theorem 2.1** As the population size $n_t \to \infty$, the random variable $\vartheta$ generated in (5) converges in distribution to a random variable $\theta$ which is distributed with the pdf $\psi(\theta)$.

Let $(\theta_1, w_1), \ldots, (\theta_N, w_N)$ be a series of correctly weighted populations generated by a DWIS algorithm with respect to $\psi(\theta)$. Then the quantity $\mu = E_{\psi} \rho(\theta)$, provided its existence, can be estimated by
\[
\hat{\mu} = \frac{\sum_{t=1}^{n} \sum_{i=1}^{n_t} w_{t,i} \rho(\theta_{t,i})}{\sum_{t=1}^{n} \sum_{i=1}^{n_t} w_{t,i}}.
\]
Let $U_t = \sum_{i=1}^{n_t} w_{t,i} \rho(\theta_{t,i})$, $S_t = \sum_{i=1}^{n_t} w_{t,i}$, $S = ES_t$, and $V_t = U_t - \mu S_t$. If the variance of $U_t$ and $V_t$ are both finite, then the standard error of $\hat{\mu}$ can be calculated using the ratio estimate as in finite population sampling [18]. As shown by Liang[14], $\hat{\mu}$ is consistent and asymptotically normally distributed; that is,
\[
\sqrt{N}(\hat{\mu} - \mu) \to N(0, \sigma^2),
\]
where $\sigma^2$ is defined as $\text{Var}(V_t)/S^2$.

**Definition 2.3** A transition rule for a population $(\theta, w)$ is said to be invariant with respect to the dynamic importance weights (IDIW) if the joint density of $(\theta, w)$ remains correctly weighted whenever the initial joint density is correctly weighted.

3. Monte Carlo dynamically weighted importance sampling

In this section, we describe a Monte Carlo dynamic weighting (MCDW) sampler and a population control scheme for the MCDWS, and show that they both are IDIW. Let $(\theta_t, w_t)$ denote the current population, let $(\theta, w)$ denote an individual state of $(\theta_t, w_t)$, and let $(\theta', w')$ denote the individual state transmitted from $(\theta, w)$ in one transition step.
3.1. A general Monte Carlo dynamic weighting sampler

A general Monte Carlo dynamic weighting scheme can be described as follows.

1. Draw \( \theta^* \) from some proposal distribution \( T(\theta^*|\theta) \).
2. Simulate auxiliary samples \( y_1, \ldots, y_m \) from \( f(y|\theta^*) \) using a MCMC algorithm, say, the MH algorithm. Estimate the normalizing constant ratio \( R_t(\theta, \theta^*) = Z(\theta)/Z(\theta^*) \) by

\[
\hat{R}_t(\theta, \theta^*) = \frac{1}{m} \sum_{i=1}^{m} \frac{p(y_i|\theta)}{p(y_i, \theta^*)},
\]

which is also known as the importance sampling (IS) estimator of \( R_t(\theta, \theta^*) \).
3. Calculate the Monte Carlo dynamic weighting ratio

\[
r_d = r_d(\theta, \theta^*, w) = w\hat{R}_t(\theta, \theta^*) \frac{p(x, \theta^*) T(\theta|\theta^*)}{p(x, \theta) T(\theta|\theta)}.\]

4. Choose \( \beta_t = \beta_t(\theta_t, w_t) \geq 0 \) and draw \( U \sim \text{unif}(0, 1) \). Update \( (\theta, w) \) as \( (\theta', w') \)

\[
(\theta', w') = \begin{cases} 
(\theta^*, r_d/a), & \text{if } U \leq a, \\
(\theta, w/(1 - a)), & \text{otherwise},
\end{cases}
\]

where \( a = r_d/(r_d + \beta_t) \); \( \beta_t \) is a function of \( (\theta_t, w_t) \), but remains a constant for each individual state of the same population.

The sampler is termed “Monte Carlo” in the sense that \( R_t(\theta, \theta^*) \) is substituted by its Monte Carlo estimator in calculation of \( r_d \). If \( R_t(\theta, \theta^*) \) is available analytically, then the sampler is reduced to the dynamic weighting sampler \([14]\). For this sampler, we have two remarks as follows.

**Remark 1.** As pointed out in \([19]\), \( \hat{R}_t(\theta, \theta^*) \) is an unbiased and consistent estimator of \( R_t(\theta, \theta^*) \). Following from the central limit theorem,

\[
\sqrt{m} \left( \hat{R}_t(\theta, \theta^*) - R_t(\theta, \theta^*) \right) \to N \left( 0, \sigma_t^2 \right),
\]

where \( \sigma_t^2 \) can be expressed as

\[
\sigma_t^2 = \text{Var} \left( \frac{p(y_1, \theta)}{p(y_1, \theta^*)} \right) + 2 \sum_{i=2}^{\infty} \text{Cov} \left( \frac{p(y_1, \theta)}{p(y_1, \theta^*)}, \frac{p(y_i, \theta)}{p(y_i, \theta^*)} \right).
\]

Alternatively, we can write \( \hat{R}_t(\theta, \theta^*) = R_t(\theta, \theta^*)(1 + \epsilon_t) \), where \( \epsilon_t \sim N(0, \sigma_t^2/\{m R_t^2(\theta, \theta^*)\}) \).

**Remark 2.** The parameter \( \beta_t \) can be chosen as a function of the population \( (\theta_t, w_t) \). For simplicity, this article concentrates only on the cases where \( \beta_t = 0 \) or \( 1 \). If \( \beta_t \equiv 0 \), then the sampler is essentially a random walk induced by \( T(\cdot|\cdot) \) on the space \( \Theta \). If \( \beta_t \equiv 1 \), then the sampler is essentially the R-type move \([15]\), which has the capability of moving across energy barriers by self-adjusting the weight values.

The following theorem shows that the Monte Carlo dynamic weighting sampler is IDIW when the initial condition

\[
\int c_{0, \theta} T(\theta|\theta')d\theta = c_{0, \theta'}
\]

holds for some population \( (\theta_{0, \theta}, w_{0, \theta}) \).
**Theorem 3.1** The Monte Carlo dynamic weighting sampler is IDIW; that is, if the joint distribution \( g_t(\theta, w) \) for \((\theta_t, w_t)\) is correctly weighted with respect to \( \pi(\theta|x) \) and (10) holds, after one Monte Carlo dynamic weighting step, the new joint density \( g_{t+1}(\theta', w') \) for \((\theta_{t+1}, w_{t+1})\) is also correctly weighted with respect to \( \pi(\theta|x) \), and (10) still holds.

**Proof:** For the case \( \beta_t > 0 \),

\[
\int_0^\infty w' g_{t+1}(\theta', w') dw' = \int_\Theta \int_0^\infty \int_{-\infty}^\infty \left[ r_d(\theta, \theta', w) + \beta_t g_t(\theta, w) T(\theta'|\theta) \varphi(\epsilon_t) \right] \frac{r_d(\theta, \theta', w)}{r_d(\theta, \theta', w) + \beta_t} d\epsilon_t dw d\theta \\
+ \int_\Theta \int_0^\infty \int_{-\infty}^\infty \frac{w [r_d(\theta', \theta, w) + \beta_t] g_t(\theta', w) T(\theta'|\theta') \varphi(\epsilon_t)}{\beta_t} \frac{\beta_t}{r_d(\theta, \theta', w) + \beta_t} d\epsilon_t dw d\theta \\
= (I) + (II),
\]

where \( \varphi(\cdot) \) denotes the density of \( \epsilon_t \). For (I), we have

\[
(I) = \int_\Theta \int_0^\infty \int_{-\infty}^\infty w R(\theta, \theta')(1 + \epsilon_t) p(x, \theta') \pi(\theta') \frac{T(\theta'|\theta) T(\theta'|\theta)}{T(\theta|\theta')} g_t(\theta, w) T(\theta'|\theta) \varphi(\epsilon_t) d\epsilon_t dw d\theta \\
= \int_\Theta \int_0^\infty \int_{-\infty}^\infty w R(\theta, \theta') \frac{p(x, \theta') \pi(\theta')}{p(x, \theta) \pi(\theta)} T(\theta'|\theta) g_t(\theta, w) dwd\theta \\
= \int_\Theta \frac{\pi(\theta'|x)}{\pi(\theta|x)} T(\theta'|\theta) \left( \int_0^\infty w g_t(\theta, w) d\theta \right) \\
= \pi(\theta'|x) \int_\Theta c_{t,\theta} T(\theta'|\theta) d\theta \\
= c_{t,\theta'} \pi(\theta'|x).
\]

For (II), we have

\[
(II) = \int_\Theta \int_0^\infty \int_{-\infty}^\infty w g_t(\theta', w) T(\theta'|\theta') \varphi(\epsilon_t) d\epsilon_t dw d\theta \\
= \int_\Theta \int_0^\infty w g_t(\theta', w) T(\theta'|\theta') dwd\theta \\
= \int_\Theta c_{t,\theta'} \pi(\theta'|x) T(\theta'|\theta') d\theta \\
= c_{t,\theta'} \pi(\theta'|x).
\]

In summary of (I) and (II), we have

\[
\int_0^\infty w' g_{t+1}(\theta', w') dw' = 2 c_{t,\theta'} \pi(\theta'|x).
\]

For the case \( \beta_t = 0 \), only the term (I) remains. Thus,

\[
\int_0^\infty w' g_{t+1}(\theta', w') dw' = c_{t,\theta'} \pi(\theta'|x).
\]

By defining \( c_{t+1,\theta'} = 2 c_{t,\theta'} \) for the case \( \beta_t > 0 \) and \( c_{t+1,\theta'} = c_{t,\theta'} \) for the case \( \beta_t = 0 \), it is easy to see that the condition (10) still holds for the new population. Hence, \( g_{t+1}(\theta', w') \) is still correctly weighted with respect to \( \pi(\theta|x) \). □
In practice, the initial condition (10) can be easily satisfied. For example, choose \( g_0(\theta, w) = g(\theta) \) and set \( w = \bar{R} p(x, \theta) \pi(\theta)/g(\theta) \), where \( \bar{R} \) denotes an unbiased estimator of \( 1/Z(\theta) \), then \( \int \int w g_0(\theta, w) \phi(e) de dw = \pi(\theta|x) \); that is, \( c_{t, \theta} = 1 \). The initial condition is satisfied because \( \int T(\theta|\theta') d\theta = 1 \) always holds.

To avoid an extremely large weight caused by a nearly zero divisor, similarly to [14, 16], we make the following assumption for the target distribution: For any pair of \( (\theta, \theta^*) \in \Theta \times \Theta \) and any sample \( y \in X \), there exists a constant \( r_1 \) such that

\[
r_1 \leq \frac{p(y, \theta)}{p(y, \theta^*)} \leq \frac{1}{r_1}.
\]  

This condition can be easily satisfied by restricting both \( \Theta \) and \( X \) to be compact. This is natural for many spatial problems, such as the autologistic model studied in Section 4. In addition, for the proposal distribution, we assume that there exists a constant \( r_2 \) such that for any pair \( (\theta, \theta^*) \in \Theta \times \Theta \),

\[
r_2 \leq \frac{T(\theta|\theta^*)}{T(\theta^*|\theta)} \leq \frac{1}{r_2}.
\]

This condition can be easily satisfied by using a symmetric proposal, which implies that \( T(\theta|\theta^*)/T(\theta^*|\theta) = 1 \), or using a bounded proposal, e.g., \( \theta^* \sim Uniform(\theta_1, \theta_r) \) with \( \theta_1 \) and \( \theta_r \) being fixed constants. Following from (8), (11) and (12), it is easy to see that there exists a constant \( r_0 \) such that for any pair \( (\theta, \theta^*) \in \Theta \times \Theta \),

\[
r_0 \leq \frac{\hat{R}_t(\theta, \theta^*) p(x, \theta^*)}{\hat{R}_t(x, \theta) p(x, \theta)} \leq \frac{1}{r_0}.
\]

### 3.2. A population control scheme

Given the current population \((\theta_t, w_t)\), to adjust the dynamic weights and the population size to suitable ranges, we propose the following adaptive pruned-enriched population control scheme (APEPCS), which is similar to the one used in [14].

Let \((\theta_{t,i}, w_{t,i})\) be the \(i\)th individual state of the population, let \(n_t\) and \(n'_t\) denote the current and new population sizes, let \(W_{\text{low},t}\) and \(W_{\text{up},t}\) denote the lower and upper weight control bounds, let \(n_{\text{min}}\) and \(n_{\text{max}}\) denote the minimum and maximum population size allowed by the user, and let \(n_{\text{low}}\) and \(n_{\text{up}}\) denote the lower and upper reference bound of the population size.

1. **Initialization** Initialize the parameters \(W_{\text{low},t}\) and \(W_{\text{up},t}\) by

\[
W_{\text{low},t} = \frac{\sum_{i=1}^{n_t} w_{t,i}}{n_{\text{up}}}, \quad W_{\text{up},t} = \frac{\sum_{i=1}^{n_t} w_{t,i}}{n_{\text{low}}}.
\]

Set \(n'_t = 0\) and \(\lambda > 1\). Do steps 2–4 for \(i = 1, 2, \ldots, n_t\).

2. **Pruned** If \(w_{t,i} < W_{\text{low},t}\), prune the state with probability \(q = 1 - w_{t,i}/W_{\text{low},t}\). If it is pruned, drop \((\theta_{t,i}, w_{t,i})\) from \((\theta_t, w_t)\); otherwise, update \((\theta_{t,i}, w_{t,i})\) as \((\theta_{t,i}, W_{\text{low},t})\) and set \(n'_t = n_t + 1\).

3. **Enriched** If \(w_{t,i} > W_{\text{up},t}\), set \(d = \lceil w_{t,i}/W_{\text{up},t} \rceil\), \(w_{t,i}' = w_{t,i}/d\), replace \((\theta_{t,i}, w_{t,i})\) by 

   \[d\] identical states \((\theta_{t,i}, w_{t,i}')\), and set \(n'_t = n_t + d\), where \(\lceil z \rceil\) denotes the integer part of \(z\).

4. **Unchanged** If \(W_{\text{low},t} \leq w_{t,i} \leq W_{\text{up},t}\), keep \((\theta_{t,i}, w_{t,i})\) unchanged, and set \(n'_t = n_t + 1\).

5. **Checking** If \(n'_t > n_{\text{max}}\), set \(W_{\text{low},t} \leftarrow \lambda W_{\text{low},t}\), \(W_{\text{up},t} \leftarrow \lambda W_{\text{up},t}\) and \(n'_t = 0\), do step 2–4 again for \(i = 1, 2, \ldots, n_t\). If \(n'_t < n_{\text{min}}\), set \(W_{\text{low},t} \leftarrow W_{\text{low},t}/\lambda\), \(W_{\text{up},t} \leftarrow W_{\text{up},t}/\lambda\) and \(n'_t = 0\), do step 2–4 again for \(i = 1, 2, \ldots, n_t\). Otherwise, stop.
In this scheme, \( \lambda \) is required to set to a number greater than 1. In all examples of this article, we set \( \lambda = 2 \). In addition, \( n_{\text{low}}, n_{\text{up}}, n_{\text{min}}, \) and \( n_{\text{max}} \) are required to satisfy the constraint \( n_{\text{min}} < n_{\text{low}} < n_{\text{up}} < n_{\text{max}} \). With the APEPCS, the population size is strictly controlled to the range \([n_{\text{min}}, n_{\text{max}}]\), and the weights are adjusted to the range \([W_{\text{low},t}, W_{\text{up},t}]\). Therefore, the APEPCS avoids the possible overflow or extinction of a population in simulations.

The APEPCS is closely related to the resampling technique that is often used in the context of importance sampling [20]. It can also be viewed as a generalization of the pruned-enriched Rosenbluth method [21] and the rejection controlled sequential importance sampler [22]. Refer to [14] for more discussions on the relationship between APEPCS and the latter two methods.

The following theorem shows that the APEPCS is IDIW, whose proof can be found in [14].

**Theorem 3.2** The APEPCS is IDIW; that is, if the joint distribution \( g_t(\theta, w) \) for \( (\theta_t, w_t) \) is correctly weighted with respect to \( \pi(\theta|x) \), then after one run of the scheme, the new joint distribution \( g_{t+1}(\theta', w') \) for \( (\theta_{t+1}, w_{t+1}) \) is also correctly with respect to \( \pi(\theta|x) \).

**Proof:** In the population control step, only the weights \( w \)'s are possibly modified, whereas the \( \theta \)'s are not changed. Thus,

\[
\int_0^\infty w' g_{t+1}(\theta', w') d\omega' = \int_0^{W_{\text{low},t}} 0 g_t(\theta', w) \left( 1 - \frac{w}{W_{\text{low},t}} \right) d\omega + \int_0^{W_{\text{low},t}} W_{\text{low},t} g_t(\theta', w) \frac{w}{W_{\text{low},t}} d\omega
\]

\[
+ \int_{W_{\text{low},t}}^{W_{\text{up},t}} w g_t(\theta', w) d\omega + \int_{W_{\text{up},t}}^\infty \sum_{i=1}^d \frac{w}{d} g_t(\theta', w) d\omega
\]

\[
= \int_0^{W_{\text{low},t}} w g_t(\theta', w) d\omega = c_t, \theta' \psi(\theta').
\]

Hence, \( g_{t+1}(x', w') \) is still correctly weighted with respect to \( \psi(\cdot) \) by setting \( c_{t+1}, \theta' = c_t, \theta' \). \( \Box \)

### 3.3. A Monte Carlo dynamically weighted importance sampler

Since both the MCDW move and the population control scheme are IDIW, they can be used together in any fraction, while leaving the IDIW property unchanged. Based on this observation, we compose the following sampler, which alters between the MCDW step and the population control step. Let \( W_c \) denote a dynamic weighting move switching parameter, which switches the value of \( \beta_t \) between 0 and 1 depending on the value of \( W_{\text{up},t} \).

- (Move type setting) If \( W_{\text{up},t} \leq W_c \), then set \( \beta_t = 1 \). Otherwise, set \( \beta_t = 0 \).
- (MCDW) Apply the Monte Carlo dynamic weighting move to the population \((\theta_t, w_t)\). The new population is denoted by \((\theta'_{t+1}, w'_{t+1})\).
- (Population Control) Apply APEPCS to \((\theta'_{t+1}, w'_{t+1})\). The new population is denoted by \((\theta_{t+1}, w_{t+1})\).

Let \((\theta_1, w_1), \ldots, (\theta_N, w_N)\) denote a series of populations generated by MCDWIS. Then, according to (6), the quantity \( \mu = E_{\pi} \rho(\theta) \) can be estimated by

\[
\hat{\mu} = \frac{\sum_{t=N_0+1}^N \sum_{i=1}^{n_t} w_{t,i} \rho(\theta_{t,i})}{\sum_{t=N_0}^N \sum_{i=1}^{n_t} w_{t,i}},
\]

(14)

where \( N_0 \) denotes the number of burn-in iterations. Following the generalized theory of importance sampling, \( \hat{\mu} \) is consistent and asymptotically normally distributed.
3.4. Weight behavior analysis

To analyze the weight behavior of the MCDWIS, we first introduce the following lemma.

**Lemma 3.1** Let \( f(x|\theta) = p(x, \theta)/Z(\theta) \) denote the likelihood function of \( x \), let \( \pi(\theta) \) denote the prior distribution of \( \theta \), and let \( T(\cdot|\cdot) \) denote a proposal distribution of \( \theta \). Define \( p(\theta, \theta'|x) = p(x, \theta)\pi(\theta)T(\theta'|\theta) \), and \( r(\theta, \theta') = \hat{R}(\theta, \theta') p(\theta'|\theta|x)/p(\theta|\theta'|x) \) to be a Monte Carlo MH ratio, where \( \hat{R}(\theta, \theta') \) denotes an unbiased estimator of \( Z(\theta)/Z(\theta') \). Then

\[
e_0 = E \log r(\theta, \theta') \leq 0,
\]

where the expectation is taken with respect to the joint density \( \varphi(\hat{R}) \times p(\theta, \theta'|x)/Z(\theta) \).

**Proof:** By Jensen’s inequality,

\[
e_0 = E \log \left[ \hat{R}(\theta, \theta') \frac{p(\theta', \theta|x)}{p(\theta, \theta'|x)} \right] \leq \log E \left[ \hat{R}(\theta, \theta') \frac{p(\theta', \theta|x)}{p(\theta, \theta'|x)} \right] = 0,
\]

where the equality holds when \( p(\theta', \theta|x) = p(\theta, \theta'|x) \), and \( \varphi(\cdot) \) is a Dirac measure with \( \varphi(\hat{R} = R) = 1 \) and 0 otherwise. \( \square \)

For simplicity, let \((\theta_t, w_t)\) denote an individual state of the population \((\theta_t, w_t)\). When \( \beta_{t-1} = 0 \), the weights in the MCDW moves evolve as

\[
\log w_t = \log w_{t-1} + \log r(\theta_{t-1}, \theta_t),
\]

which results in

\[
\log w_t = \log w_0 + \sum_{s=1}^{t} \log r(\theta_{s-1}, \theta_s). \tag{15}
\]

Following from Lemma 1 and the ergodicity theorem (under stationarity),

\[
\frac{1}{t} \sum_{s=1}^{t} \log r(\theta_{s-1}, \theta_s) \rightarrow e_0 < 0, \quad a.s. \tag{16}
\]

as \( t \rightarrow \infty \). Hence, \( w_t \) will go to 0 almost surely as \( t \) becomes large.

When \( \beta_{t-1} = 1 \), the expectation of \( w_t \), conditioned on \( \theta_{t-1}, \theta_t \) and \( w_{t-1} \), can be calculated as

\[
E[w_t|\theta_{t-1}, \theta_t, w_{t-1}] = (r_d + 1) \frac{r_d}{r_d + 1} + w_{t-1}(r_d + 1) \frac{1}{r_d + 1} = r_d + w_{t-1}
\]

\[
= w_{t-1}[1 + r(\theta_{t-1}, \theta_t)]. \tag{17}
\]

Since \( r(\theta_{t-1}, \theta_t) \geq 0 \), the weight process is driven by an inflation drift. Hence, \( w_t \) will go to infinity almost surely as \( t \) becomes large.

To prevent the weight process from going to 0 and \( \infty \), the MCDWIS alters the use of \( \beta_t = 0 \) and \( \beta_t = 1 \). When \( W_{mp,t} > W_c \), \( \beta_t \) is set to 0, so the weight process of MCDWIS can be bounded above by

\[
\log w_{t+s} = \log w_t + \sum_{j=1}^{s} \log r(\theta_{t+j-1}, \theta_{t+j}) - \sum_{j=1}^{s} \log (d_{t+j}),
\]

This completes the analysis of the weight behavior in the MCDWIS.
assuming that $\beta_t = \beta_{t+1} = \cdots = \beta_{t+s} = 0$, where $d_{t+j}$ is a positive integer as defined in the APEPCS. When $W_{up,t+s} < W_t$, $\beta_{t+s}$ is set to 1, so the weight can be boosted to a large value.

In summary, the weight process of MCDWIS can be characterized by the following process:

$$Z_t = \begin{cases} Z_{t-1} + \log r(\theta_{t-1}, \theta_t) - \log(d_t), & \text{if } Z_{t-1} > 0, \\ 0, & \text{if } Z_{t-1} < 0, \end{cases}$$

(18)

where there exists a constant $C$ such that $|Z_t - Z_{t-1}| \leq C$ almost surely. The existence of $C$ follows directly from condition (13). Let $T_0 = 0$, $T_i = \min\{t : t > T_{i-1}, Z_t = 0\}$, and $L_i = T_i - T_{i-1}$ for $i \geq 1$. From (16) and the fact that $d_i \geq 1$, it is easy to see that $L_i$ is almost surely finite; that is, there exists an integer $M$ such that $P(L_i < M) = 1$. This implies that for any fixed $\eta > 0$,

$$E \exp(\eta Z_t) \leq \exp(\eta MC), \quad a.s.$$

This leads to the following theorem:

**Theorem 3.3** Under the assumptions (11) and (12), the MCDWIS almost surely has finite moments of any order.

Finally, we would like to mention that the ratio $\kappa = W_{up,t}/W_{low,t}$ determines the moving ability of the MCDWIS. In [14], $\kappa$ is called the freedom parameter of DWIS. In the APEPCS, $\kappa$ is also equal to the ratio $n_{up}/n_{low}$. As suggested by Liang [14], when choosing the value of $\kappa$, the efficiency of the resulting sampler should be taken into account. A large $\kappa$ will ease the difficulty of escaping from local energy minima, but the variability of the weights will also increase accordingly. The efficiency of the MCDWIS may be reduced by an excessively large $\kappa$.

In this paper, we set $\kappa = 2.5$ in all simulations by setting $n_{low} = 200$ and $n_{up} = 500$.

4. Bayesian analysis for spatial autologistic models

The autologistic model [1] has been widely used for spatial data analysis [23–25]. Let $x = \{x_i : i \in D\}$ denote the observed binary data, where $x_i$ is called a spin and $D$ is the set of indices of the spins. Let $|D|$ denote the total number of spins in $D$, and let $n(i)$ denote the set of neighbors of spin $i$. The likelihood function of the model is

$$f(x|\theta) = \frac{1}{Z(\theta)} \exp \left\{ \theta_a \sum_{i \in D} x_i + \frac{\theta_b}{2} \sum_{i \in D} x_i \left( \sum_{j \in n(i)} x_j \right) \right\}, \quad (\theta_a, \theta_b) \in \Theta,$$

(19)

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

$$Z(\theta) = \sum_{\text{for all possible } x} \exp \left\{ \theta_a \sum_{j \in D} x_j + \frac{\theta_b}{2} \sum_{i \in D} x_i \left( \sum_{j \in n(i)} x_j \right) \right\}.$$  

An exact evaluation of $Z(\theta)$ is impossible even for a moderate system.

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

$$(\theta_a, \theta_b) \in \Theta = [-1, 1] \times [0, 1]$$

for all examples studied in this section. Then the MCDWIS can be applied to simulate from the posterior distribution $\pi(\theta|x)$. Since, for this model, $\mathcal{X}$ is finite and $\Theta$ is compact, the condition (11) is satisfied. As long as the proposal distribution satisfies the condition (12), the weights will have finite moments of order. In all simulations of this model, we used a Gaussian random walk proposal. Thus, $T(\theta|\theta^*)/T(\theta^*|\theta) = 1$, and condition (12) is satisfied.
4.1. U.S. cancer mortality data

United States cancer mortality maps have been compiled for investigating possible association of cancer with unusual demographic, environmental, industrial characteristics, or employment patterns [26]. Figure 2(a) 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 clustering. Refer to [25] for more descriptions of the data. As in [25], we modeled the data by a spatial autologistic model. The total number of spins is $|D| = 2293$. A free boundary condition is assumed for the model, under which the boundary points have less neighboring points than the interior points. The assumption is natural to this example, as the lattice has an irregular shape.

![Figure 2. US cancer mortality data. (a) 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. (b) Fitted cancer mortality rates by the autologistic model with the parameters being replaced by its approximate Bayesian estimates. The cancer mortality rate of each county is represented by the gray level of the corresponding square.](image)

The MCDWIS was applied to this example with the setting as described below. In each run, the MCDWIS was initialized with $n_0 = 250$ random samples of $\theta$ generated in a short run of the double MH algorithm [10], with each sample being assigned an equal weight of 1. Refer to Appendix for the details of the double MH algorithm. To generate the initial samples, the double MH algorithm was run for 3500 iterations, where the first 1000 iterations were discarded for the burn-in process, and then 250 samples were collected equally-spacedly from the remaining 2500 iterations and were used as the initial samples. In each iteration, the auxiliary variable is generated with a single MH update. As aforementioned, the double MH algorithm is very fast, but only can sample from the posterior distribution $\pi(\theta|x)$ approximately, even when the run becomes very long.

After initialization, the MCDWIS iterates between the MCDW and population control steps. In the MCDW step, the normalizing constant ratio $R_t(\theta, \theta^*)$ was estimated using 50 auxiliary samples, which were generated from 50 cycles of Gibbs updates starting with $x$; and $T(\cdot|\cdot)$ was a Gaussian random walk proposal $N_2(\theta, s^2 I_2)$, where $s$ is the step size, and $I_2$ is the $2 \times 2$ identity matrix. In this article, we set $s = 0.03$ for all examples. In the population control step, we set $\log(W_c) = 5$, $n_{\text{low}} = 200$, $n_{\text{up}} = 500$, $n_{\text{min}} = 100$ and $n_{\text{max}} = 1000$. The MCDWIS was run 10
times independently, and each run consisted of 100 iterations. The CPU time cost by each run was about 5.8m on a 3.0GHz personal computer (all computations reported in this paper were done in the same computer). Figure 3 shows the time plots of the population size, $\beta_t$ and $W_{up,t}$ produced in a MCDWIS run. It indicates that after a certain number of burn-in iterations, the population size and the magnitude of the weights can evolve stably with iterations. In each run, the first 20 iterations were discarded for the burn-in process, and the samples generated from the remaining iterations were used for inference. Averaging over the estimates obtained from the ten runs, we got the following estimate: $(\hat{\theta}_a, \hat{\theta}_b) = (-0.3016, 0.1229)$ with the standard error $(1.9 \times 10^{-3}, 8.2 \times 10^{-4})$.

For comparison, the exchange algorithm [12] was also applied to this example. Refer to the Appendix for the details of the algorithm. 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 is large. Hence, following the standard theory of Markov chain Monte Carlo [27], the estimates produced by the exchange algorithm in long runs will converge to their true values, and can be used as a test standard for assessing whether the MCDWIS works for this example. As in [12], we adopted the summary state algorithm [28] as the perfect sampler for the exchange algorithm. The exchange algorithm was also run 10 times. Each run consisted of 55000 iterations, where first 5000 iterations were discarded for the burn-in process, and the samples generated from the remaining iterations were used for inference. The CPU time cost by each run was about 6.5m. The overall acceptance rate of the perfect auxiliary variables was 0.2. Averaging over the estimates obtained from the ten runs, we got the estimate $(\hat{\theta}_a, \hat{\theta}_b) = (-0.3018, 0.1227)$ with the standard error $(2.7 \times 10^{-4}, 1.4 \times 10^{-4})$.

![Figure 3. Simulation results of the MCDWIS for the U.S. Cancer Mortality example: (a) time plot of population size; (b) time plot of $\beta_t$; and (c) time plot of log($W_{up,t}$). The dotted line in plot (c) shows the value of log($W_c$).](image)

It is easy to see that the MCDWIS and the exchange algorithm produced almost identical estimates for this example. These estimates are also almost identical to the estimate $(-0.3008, 0.1231)$ obtained using contour Monte Carlo [9], and the estimate $(-0.2999, 0.1234)$ obtained using stochastic approximation Monte Carlo [8]. We note that both the contour Monte Carlo and stochastic approximation Monte Carlo algorithms try to first approximate the unknown normalizing constant function, and then estimate the parameters based on the
approximated normalizing constant function. As reported by the authors, both the algorithms take hours of CPU time to approximate the normalizing constant function. This data has also been analyzed in [25] using the Monte Carlo maximum likelihood algorithm [7], resulting in the estimate \((-0.304, 0.117)\) which is slightly far from the estimate \((-0.3018, 0.1227)\) resultant from the exchange algorithm.

In summary, for this example the MCDWIS produced almost identical estimates of \(\theta\) with the exchange algorithm, although whose estimates have slightly larger variation. This is reasonable, as the MCDWIS is an importance sampling algorithm. It is known that an importance sampling algorithm is usually less efficient than a MCMC algorithm for the problems for which the latter works. The advantage of the MCDWIS can be seen in the next subsection, where for some cases the exchange algorithm does not work while the MCDWIS still works well.

4.2. Simulation studies

To assess general accuracy of the estimates produced by the MCDWIS, we simulated 50 independent samples for the U.S. cancer mortality data under each setting of \((\theta_a, \theta_b)\) given in Table 1. Since the lattice is irregular, the free boundary condition was again assumed in the simulations. We then re-estimated the parameters using the MCDWIS. The sampler was run as for the previous example. The computational results were summarized in Table 1.

For comparison, the exchange algorithm [12] was also applied to this example. It was run as for the previous example, except that each run was shortened to 10500 iterations. In each run, the first 500 iterations were discarded for the burn-in process, and the samples generated in the remaining iterations were used for inference. The results were summarized in Table 1. For a thorough comparison, we also included in Table 1 the maximum pseudo-likelihood estimators (MPLE) of \(\theta\), which were obtained in [10] for the same datasets. The MPLE were obtained by maximizing the pseudo-likelihood function

\[
\hat{L}(\theta|x) = \prod_{i \in D} \left[ e^{\theta_a x_i + \theta_b \sum_{j \in n(i)} x_i x_j} + e^{-\theta_a - \theta_b \sum_{j \in n(i)} x_j} \right]
\]

using the downhill simplex method [29]. The advantage of this optimization method is that it does not require gradient information of the objective function, and can thus be easily applied to the constraint optimization problems.

Table 1 indicates that the MCDWIS can produce almost the same accurate results as does the exchange algorithm, and more accurate results than does the MPLE especially when both \(\theta_a\) and \(\theta_b\) are large. It is remarkable that the CPU time cost by the MCDWIS is independent of the values of \(\theta\). Whilst as \(\theta_b\) increases, the CPU time cost by the exchange algorithm increases exponentially. Childs et al. [28] studied the behavior of the perfect sampler for the Ising model, a simplified autologistic model with \(\theta_a\) being restricted to 0. For the Ising model, they fitted an exponential law for the convergence time, and reported that the perfect sampler may diverge at a value of \(\theta_b\) lower than the critical value (\(\approx 0.44\)). Their finding is consistent with our results. It takes an extremely long CPU time for the perfect sampler to generate a sample under the settings \((0, 0.4)\) and \((0.5, 0.5)\). We note that due to the effect of \(\theta_a\), it usually takes a longer CPU time for the perfect sampler to generate a sampler under the setting \((0, \theta_b)\) than under the setting \((\theta_a, \theta_b)\); and that when both \(\theta_a\) and \(\theta_b\) are large, the accuracy of the estimates tend to be reduced by their correlation.

5. Conclusion

In this paper, we have proposed the MCDWIS as a general MCMC algorithm for simulating from distributions with intractable normalizing constants. The algorithm is illustrated with spatial autologistic models. Unlike other auxiliary variable MCMC algorithms, the new
Table 1. Computational results for the simulated U.S. cancer mortality data. The numbers in the parentheses denote the standard error of the estimates. Notes: —, not available; \(a\) the CPU time (in minutes) cost by a single run of the algorithm; \(b\) the data were simulated using the perfect sampler; \(c\) the data were simulated using the Gibbs sampler, starting with a random configuration and then iterating for 100000 Gibbs cycles.

| \((\theta_a, \theta_b)\) | \(\hat{\theta}_a\) | \(\hat{\theta}_b\) | CPU\(^a\) | \(\hat{\theta}_a\) | \(\hat{\theta}_b\) | CPU\(^a\) | \(\hat{\theta}_a\) | \(\hat{\theta}_b\) |
|----------------|----------------|----------------|---------|----------------|----------------|---------|----------------|----------------|
| \((0,0.1)^b\) | \(0.0041\) | \(0.1004\) | \(5.8\) | \(-0.0038\) | \(0.1004\) | \(1.2\) | \(-0.0035\) | \(0.1016\) |
| | \((0.0025)\) | \((0.0019)\) | | \((0.0024)\) | \((0.0018)\) | | \((0.0024)\) | \((0.0019)\) |
| \((0,0.2)^b\) | \(0.0022\) | \(0.2020\) | \(5.8\) | \(-0.0023\) | \(0.2007\) | \(2.8\) | \(-0.0024\) | \(0.2025\) |
| | \((0.0021)\) | \((0.0019)\) | | \((0.0020)\) | \((0.0019)\) | | \((0.0022)\) | \((0.0022)\) |
| \((0,0.3)^b\) | \(0.0013\) | \(0.2980\) | \(5.8\) | \(-0.0011\) | \(0.2975\) | \(7.9\) | \(-0.0019\) | \(0.2981\) |
| | \((0.0013)\) | \((0.0017)\) | | \((0.0014)\) | \((0.0017)\) | | \((0.0016)\) | \((0.0022)\) |
| \((0,0.4)^b\) | \(0.0004\) | \(0.3963\) | \(5.8\) | \(0.0001\) | \(0.3979\) | \(145.8\) | \(0.0020\) | \(0.4013\) |
| | \((0.0012)\) | \((0.0020)\) | | \((0.0005)\) | \((0.0012)\) | | \((0.0012)\) | \((0.0020)\) |
A. The Exchange Algorithm [12]  Let \( \theta_t \) denote the sample generated from \( \pi(\theta|x) \) at iteration \( t \). One iteration of the algorithm consists of the following steps.

(a) Propose \( \theta' \sim q(\theta'|\theta_t, x) \), where \( q(\cdot|\cdot) \) denotes the proposal distribution.
(b) Generate an auxiliary variable \( y \sim f(y|\theta') \) using an perfect sampler.
(c) Accept \( \theta' \) with probability \( \min\{1, r(\theta_t, \theta', y|x)\} \), where

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

If it is accepted, set \( \theta_{t+1} = \theta' \). Otherwise, set \( \theta_{t+1} = \theta_t \).

Since swapping changes are involved in the simulations, it is called the exchanged algorithm.

B. The Double MH Algorithm [10]  Let \( \theta_t \) denote the sample generated at iteration \( t \) of the algorithm.

(a) Simulate a new sample \( \theta' \) from \( \pi(\theta) \) using the MH algorithm starting with \( \theta_t \).
(b) Generate an auxiliary variable \( y \) from \( f(\cdot|\theta') \) through \( m \) MH updates starting with the observation \( x \). The probability of transition from \( x \) to \( y \) is

\[
P_\theta^{(m)}(y|x) = K_{\theta'}(x \rightarrow x_1) \cdots K_{\theta'}(x_{m-1} \rightarrow y),
\]

where \( K_{\theta'}(\cdot \rightarrow \cdot) \) denotes a MH transition kernel. Accept \( y \) with probability \( \min\{1, r(\theta_t, \theta', y|x)\} \), where

\[
r(\theta_t, \theta', y|x) = \frac{f(y|\theta_t)P_\theta^{(m)}(y|x)}{f(x|\theta_t)P_\theta^{(m)}(y|x)} = \frac{f(y|\theta_t)f(x|\theta')}{f(x|\theta_t)f(y|\theta')}.
\]

(c) Set \( \theta_{t+1} = \theta' \) if the auxiliary variable is accepted in step (b), and set \( \theta_{t+1} = \theta_t \) otherwise.

Since two types of MH updates are performed in step (b), one for generating the auxiliary variable \( y \) and one for acceptance of \( \theta' \), the algorithm is called the double MH algorithm in [10].

References
[1] Besag J E 1974 J. R. Stat. Soc. B 36 192
[2] Hurn M, Husby O and Rue H 2003 Lecture Notes in Statistics 173 87
[3] Green P J and Richardson S 2002 J. Am. Stat. Assoc. 97 1055
[4] Francois O, Aneilet S and Guillot G 2006 Genetics 174 805
[5] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 J. Chem. Phys. 21 1087
[6] Hastings W K 1970 Biometrika 57 97
[7] Geyer C J and Thompson E A 1992 J. R. Stat. Soc. B 54 657
[8] Liang F, Liu C and Carroll R J 2007 J. Am. Stat. Assoc. 102 305
[9] Liang F 2007 J. Comput. Graph. Stat. 16 608
[10] Liang F 2009 J. Stat. Comput. Sim. In press
[11] Moller J, Pettitt A N, Reeves R and Berthelsen K K 2006 Biometrika 93 451
[12] Murray I, Ghahramani Z and MacKay D J C 2006 Proc. 22nd Annual Conf. on Uncertainty in Artificial Intelligence (AUAI Press) p 359
[13] Propp G and Wilson D B 1996 Random Struct. Algor. 9 223
[14] Liang F 2002 J. Am. Stat. Assoc. 97 807
[15] Wong W H and Liang F 1997 Proc. Natl. Acad. Sci., USA 94 14220
[16] Liu J S, Liang F and Wong W H 2001 J. Am. Stat. Assoc. 96 561
[17] Geweke J 1989 Econometrica 57 1317
[18] Ripley B D 1987 Stochastic Simulation (New York: John Wiley)
[19] Chen M H, Shao Q M and Ibrahim J G 2000 Monte Carlo Methods in Bayesian Computation (New York: Springer)

[20] Liu J S and Chen R 1998 J. Am. Stat. Assoc. 93 1032

[21] Grassberger P 1997 Phys. Rev. E 56 3682

[22] Liu J S, Chen R and Wong W H 1998 J. Am. Stat. Assoc. 93 1022

[23] Preisler H K 1993 Appl. Statist. 42 501

[24] Wu H and Huffer F W 1997 Ecological Statistics 4 49

[25] Sherman M, Apanasovich T V and Carroll R J 2006 J. Stat. Comput. Sim. 76 167

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

[27] Tierney L 1994 Ann. Stat. 22 1701

[28] Childs A M, Patterson R B and MacKay D J C 2001 Phys. Rev. E 63 036113

[29] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 1992 Numerical Recipes in C (2nd ed) (Cambridge University Press)

[30] Wasserman S and Faust K 1994 Social Network Analysis: Methods and Applications (Cambridge: Cambridge University Press)