# Markov Chain Monte Carlo and Related Topics

Jun S. Liu
Department of Statistics
Stanford University
Sequoia Hall, Stanford, CA 94305
Email: *jliu@stat.stanford.edu*

**Summary:** This article provides a brief review of recent developments in Markov chain Monte Carlo methodology. The methods discussed include the standard Metropolis-Hastings algorithm, the Gibbs sampler, and various special cases of interest to practitioners. It also devotes a section on strategies for improving mixing rate of MCMC samplers, e.g., simulated tempering, parallel tempering, parameter expansion, dynamic weighting, and multigrid Monte Carlo with its generalizations. Other related topics are the simulated annealing, the reversible jump method, and the multiple-try Metropolis rule. Theoretical issues such as bounding the mixing rate, diagnosing convergence, and conducting perfect simulations are only briefly mentioned.

# 1 Introduction

Computer simulation techniques are indispensable tools for solving difficult computational problems in many scientific disciplines. Their wide applications range from biology (Leach 1996; Karplus and Petsko 1990; Lawrence et al. 1993), chemistry (Alder and Wainwright1959), computer science (Kirkpatrick et al. 1983; Ullman 1984), economics and finance, engineering (Geman and Geman 1984), material science (Frenkel and Smit 1996), physics (Metropolis et al. 1953; Goodman and Sokal 1989), to statistics. Among all simulation methods, Monte Carlo methodology, especially *Markov chain Monte Carlo* (MCMC), provides an enormous scope for realistic statistical modeling and has attracted much attention from statisticians.

A fundamental step in all Monte Carlo methods is to generate pseudo-random samples that follow a target probability distribution function $\pi(x)$. The variable of interest $x$ usually takes value in $R^k$ but occasionally can take values in a topological group (Diaconis 1988; Liu and Wu 1999). In most applications, directly generating independent samples from the distribution of interest $\pi$ is infeasible. It is often the case that either the generated samples have to be dependent, or the distribution used to generate the samples is different from $\pi$, or both. Rejection sampling (von Neumann 1951), importance sampling (Marshall 1956) and sampling-importance-resampling (Rubin 1987b) are schemes that make use of dependent or independent samples generated from a *trial distribution* $p(x)$, which differs from, but should be similar to, the target distribution $\pi$. The Metropolis-Hastings algorithm (Metropolis et al. 1953; Hastings 1970), the basic building block of MCMC, is the one that generates dependent samples from a Markov chain with $\pi$ as its equilibrium distribution. In this view, MCMC is essentially a Monte Carlo integration procedure in which the random samples are produced by evolving a Markov chain. Because of the increasing complexities of statistical models encountered in practice, MCMC provides a much needed unifying framework within which many complex problems can be analyzed.

Both Bayesians and frequentists need to integrate over possibly high-dimensional probability distributions, such as missing data and nuisance parameters, to make inference for the parameter of interest or to make predictions. This basic need underlies the potential role of MCMC methodology in statistical modeling and inference. The past few years have witnessed an explosive growth of interest in MCMC methodology from researchers in almost all areas of statistics. Gilks, Richardson, and Spiegelhalter (1995) provided a good recent survey on how MCMC has been used. Steve Brooks administered a useful website *http://www.stats.bris.ac.uk/MCMC/* for entertaining new research papers in MCMC.

## 2 Prelude: Random Variable Generation

In order to generate random variables that follow a general pdf $\pi$, we need to first generate *uniformly* distributed random variables in [0,1]. However, this "simple-looking" task is not achievable on a computer. What we can do is to generate *pseudo-random* numbers. More formally, we can define a *uniform pseudo-random number generator* as an algorithm which, starting from an initial value $u_0$ (i.e., the *seed*), produces a sequence $(u_i) = (D^i(u_0))$ of values in [0,1]. For all $n$, the values $(u_1, \ldots, u_n)$ should reproduce the behavior of an *iid* sample $(V_1, \ldots, V_n)$ of uniform random variables. There are a few very good pseudo-random number generators available. We refer the reader to Marsaglia and Zaman (1993) and Knuth (1981) for further reference. From now on, we *assume* that uniform random variables can be satisfactorily produced on computer. Then we have the following simple result, whose proof is left as an exercise for the reader.

**Lemma 2.1** *Suppose $U \sim$ Unif [0,1] and $F$ is a 1-dimensional cumulative distribution function (cdf). Then $X = F^{-1}(U)$ has the distribution $F$. Here we define $F^{-1}(u) = \inf\{x;\ F(x) \geq u\}$.*

This lemma provides us an explicit way of generating a one-dimensional random variable when its cdf is available. Since many distributions (e.g., Gaussian distribution) do not have a closed-form cdf, it is often difficult to directly apply the above inversion method. To overcome this limitation, von Neumann (1951) proposed the popular *rejection method*, which can also be applied to draw from multi-dimensional distributions.

**Lemma 2.2** *(van Neumann 1951). Suppose $\pi(x)$ and $g(x)$ are two pdfs defined on the same sample space and that there exists $M$ so that $\pi(x) \leq Mg(x)$, $\forall x$. Then the output $X$ from the following algorithm follows distribution $\pi$:*

1. *Generate $Y \sim g$, and $U \sim$ Unif [0,1];*
2. *Accept $X = Y$ if $U \leq \pi(Y)/Mg(Y)$; otherwise go back to step 1.*

For a given target distribution $\pi$, we can implement this rejection method by first employing a distribution $g$ that is easy to generate sample from and then finding the "envelop constant" $M$. Clearly, the efficiency of the method depends on how large $M$ is. Some comparisons of this method with other approaches (such as importance sampling) can be found in Liu (1996a).

## 3 Metropolis-Hastings Algorithms

Let $\pi(x) = c \exp\{-h(x)\}$ be the target probability distribution function under investigation (presumably all pdfs can be written in this form). Metropolis et al. (1953) introduced the fundamental

idea of evolving a Markov process to achieve the sampling of $\pi$. Hastings (1970) later provided a more general form of this type of algorithms.

### 3.1 The Metropolis algorithm

Starting with any configuration $x^{(0)}$, the Metropolis algorithm iterates the following two steps.

Step 1: Propose a random "perturbation" of the current state, i.e., $x^{(t)} \to x'$, where $x'$ can be seen as generated from a *symmetric* probability transition function $T(x^{(t)}, x')$ (i.e., $T(x, x') = T(x', x)$); calculate the change $\Delta h = h(x') - h(x^{(t)})$.

Step 2: Generate a random number $u \sim$ uniform(0,1). Let $x^{(t+1)} = x'$ if $u \leq \exp(-\Delta h)$, and let $x^{(t+1)} = x^{(t)}$ otherwise.

The Metropolis scheme has been extensively used in statistical physics over the past 40 years and is the cornerstone of all Markov chain Monte Carlo (MCMC) techniques recently adopted and further developed in the statistics community.

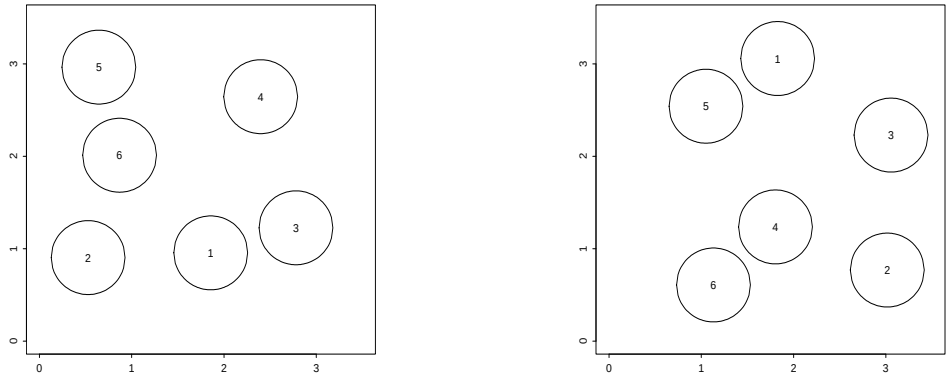



Figure 1: Simulation of balls' movements in a closed box by the Metropolis algorithm. Left: after 1000 iterations; right: after 2000 iterations.

As an example, we consider simulating uniformly distributed positions of $K$ hard-shell balls in the box $[0, A] \times [0, B]$. These balls are assumed to have equal diameter $d$. Let $(X, Y) = \{(x_i, y_i),\ i = 1, \ldots, K\}$ denote the positions of these balls. The target distribution of interest $\pi(X, Y)$ is equal to a positive constant when the balls are all in the box and have no overlaps, and is equal to zero otherwise. The Metropolis algorithm can be implemented as follows: (a) pick a ball at random, say, its position is $(x_i, y_i)$; (b) move it to a tentative position $(x_i', y_i') = (x_i + \delta_1, y_i + \delta_2)$, where

$\delta_j \sim N(0, \sigma_0^2)$; and (c) accept the proposal $(x_i', y_i')$ if it does not violate the constraints (i.e., within the box and has no overlap with others). Otherwise stay put. With $K = 6$, $d = 0.8$, $A = B = 3.5$, and starting positions of the balls at regular grids, we adjusted $\sigma_0^2$ to 0.5 which gave us an acceptance rate of about 30%. Figure 1 shows two snap-shots of this simulation: the first one was taken after 1000 iterations, and the second one taken after 2000 iterations.

### 3.2 Hastings' generalization and mathematical formulation

For any given $\pi(x)$, the Metropolis-Hastings algorithm prescribes a *transition rule* for a Markov chain so that the equilibrium distribution of the chain is $\pi(x)$. To start the algorithm, one first gives an *arbitrary*, but easy to sample from, transition function $T(x, y)$ (often called a *proposal distribution*). With the proposal distribution, one can implement the following iteration:

**Metropolis-Hastings Algorithm:** Given current state $x^{(t)}$,

- Draw $y$ from the proposal distribution $T(x^{(t)}, y)$.
- Draw $U \sim$ Uniform[0,1] and update

$$x^{(t+1)} = \begin{cases} y, & \text{if } U \le \rho(x^{(t)}, y) \\ x^{(t)} & \text{Otherwise} \end{cases} \tag{1}$$

  where Metropolis et al. (1953) and Hastings (1970) suggested to use

$$\rho(x, y) = \min\left\{1, \ \frac{\pi(y)T(y, x)}{\pi(x)T(x, y)}\right\}.$$

Baker (1965) suggested another acceptance/rejection function:

$$\rho(x, y) = \frac{\pi(y)T(y, x)}{\pi(y)T(y, x) + \pi(x)T(x, y)}.$$

A more general formula for $\rho(x, y)$ is given by Charles Stein (personal communication):

$$\rho(x, y) = \frac{\delta(x, y)}{\pi(x)T(x, y)}, \tag{2}$$

where $\delta(x, y)$ is any *symmetric function* in $x$ and $y$ that makes $\rho(x, y) \le 1$ for all $x, y$. Note that in all of the foregoing formulas, $T(x, y)$ cancels with $T(x, y)$ if it is a *symmetric* proposal, as originally required by Metropolis et al. (1953). The intuition behind the ratio $T(y, x)/T(x, y)$ is to compensate the "flow-bias" of the proposal chain.

If a rejection function of form (2) is used, then for any $y \neq x$ the actual transition probability from $x$ to $y$ implied by the Metropolis-Hastings's rule is

$$A(x, y) = T(x, y)\rho(x, y) = T(x, y)\frac{\delta(x, y)}{\pi(x)T(x, y)} = \pi(x)^{-1}\delta(x, y).$$

Because $\delta(x,y) = \delta(y,x)$, we have that $\pi(x)A(x,y) = \pi(y)A(y,x)$ for all $x \neq y$. This implies that the Markov chain induced by the Metropolis-Hastings rule is *reversible* and has $\pi$ as its invariant distribution. However, convergence rate of this chain is highly dependent of both $T(x,y)$ and the target distribution $\pi$. See Roberts and Tweedie (1996).

For discrete state spaces, Peskun (1973) showed that the optimal choice of $\rho(x,y)$ in terms of statistical efficiency is that of Metropolis et al. (1953)'s. But the issue is less clear in terms of convergence rate of the induced Markov chain (Frigessi et al. 1993; Liu 1996b).

# 4 Gibbs Sampling and Data Augmentation in Statistics

The Gibbs sampler (Geman and Geman 1984) is a special MCMC scheme. Its most prominent feature is that the underlying Markov chain is constructed by using a sequence of conditional distributions which are so chosen that $\pi$ is invariant with respect to each of these "conditional" moves. Thus, the Gibbs sampler effectively reduces a high-dimensional simulation problem to a series of lower dimensional ones.

## 4.1 The Gibbs sampler

Suppose $x = (x_1, \ldots, x_d)$. In the Gibbs sampler, one randomly or systematically choose a coordinate, say $x_1$, and then update it with a new sample $x_1'$ drawn from the conditional distribution $\pi(\cdot \mid x_{[-1]})$, where $x_{[-A]}$ refers to $\{x_j,\ j \in A^c\}$ for any subset $A$ of the coordinates. Algorithmically, we can describe the Gibbs sampler as follows:

*Random Scan Gibbs sampler.* Suppose currently $x^{(t)} = (x_1^{(t)}, \ldots x_d^{(t)})$. Then

- Randomly select $i$ from $\{1, \ldots, d\}$ according to a given probability vector $(\alpha_1, \ldots, \alpha_d)$.
- Draw $x_i^{(t+1)}$ from the conditional distribution $\pi(\cdot \mid x_{[-i]}^{(t)})$, and let $x_{[-i]}^{(t+1)} = x_{[-i]}^{(t)}$.

*Systematic Scan Gibbs sampler.* Currently $x^{(t)} = (x_1^{(t)}, \ldots x_d^{(t)})$.

- For $i = 1, \ldots, d$, we draw $x_i^{(t+1)}$ from the conditional distribution

$$\pi(x_i \mid x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_d^{(t)}).$$

It is easy to check that *every* individual conditional update leaves $\pi$ invariant. Suppose currently $x^{(t)} \sim \pi$. Then $x_{[-i]}^{(t)}$ follows its marginal distribution under $\pi$. Thus,

$$\pi(x_i^{(t+1)} \mid x_{[-i]}^{(t)}) \times \pi(x_{[-i]}^{(t)}) = \pi(x_i^{(t+1)}, x_{[-i]}^{(t)}),$$

which means that after one conditional update, the joint distribution of $(x_{[-i]}^{(t)}, x_i^{(t+1)})$ is still $\pi$.

Under regularity conditions, one can show that the Gibbs sampler chain converges geometrically and its convergence rate is related to how the variables correlate with each other (Schervish and Carlin 1993; Liu 1994; Tierney 1994). It was argued that grouping highly correlated variables together in the Gibbs update can greatly speed up the sampler (Liu et al. 1994; Liu 1994). Some researchers have also shown that random scan can outperform systematic scan in terms of convergence speed (Roberts and Sahu 1997).

A simple restatement of the conditional updates in the Gibbs sampler can be potentially useful: they can be seen as a way to move the point $x$ along a direction:

$$x_i \to x_i' = x_i + c,$$

where $c$ is drawn from an appropriate distribution. It is not difficult to show that if $c$ is drawn from $p(c) \propto \pi(x_i + c, x_{[-i]})$, then the move leaves $\pi$ invariant. This view is critical in generalizing the Gibbs sampler under a transformation group setting (Liu and Wu 1999), which is useful for designing more efficient MCMC samplers. See Section 7 for more discussions.

The Gibbs sampler's popularity in statistics community stems from its extensive use of *conditional distributions* in each iteration. Tanner and Wong (1987)'s data augmentation (see the Chapter for EM algorithm) first linked the Gibbs sampling structure with missing data problems and the EM-type algorithms (see the Chapter for EM algorithm). Gelfand and Smith (1990) further pointed out that the conditionals needed in Gibbs iterations are commonly available in many Bayesian and likelihood computations.

### 4.2 Data augmentation: a two component Gibbs sampler

Suppose random variable $x$ can be partitioned into two parts, $x = (x_1, x_2)$, and the current state is $(x_1^{(t)}, x_2^{(t)})$. Then a two-component Gibbs sampler updates as follows:

- Draw $x_1^{(t+1)}$ from the conditional distribution $\pi_{1|2}(\cdot \mid x_2^{(t)})$;
- Draw $x_2^{(t+1)}$ from the conditional distribution $\pi_{2|1}(\cdot \mid x_1^{(t+1)})$.

This sampler is especially interesting for the following two reasons. Firstly, it corresponds to the data augmentation algorithm (Tanner and Wong 1987), which was designed for handling Bayesian missing data problems. In such problems, one of the components, say $x_1$, often corresponds to the parameter of interest, and the other one, $x_2$ corresponds to missing data. The Gibbs iterations are then the ones between drawing the parameter value conditional on currently imputed missing data and then imputing the missing data conditional on the current parameter value. This idea is

closely related to the EM algorithm (Dempster, Laird and Rubin 1977) and multiple imputation (Rubin 1987), and has long been appealing to applied statisticians.

Secondly, the two-component Gibbs sampler has some nice theoretical properties. Under weak regularity conditions, Liu et al. (1994, 1995) showed that the sampler converges geometrically and monotonically. The convergence rate of the sampler is equal to the *maximal correlation* between the two components, which is closely related to a statistical concept, the *faction of missing information* (Rubin 1987; Liu 1994) in Bayesian missing data problems. It was found that under stationarity

$$cov[h(x_1^{(0)}), h(x_1^{(1)})] = \mathrm{var}_\pi[E_\pi\{h(x_1) \mid x_2\}],$$

holds for any function $h$, which can be used to derive an expression for lag-$n$ auto-covariances:

$$\begin{aligned} \mathrm{cov}[t(x_1^{(0)}), t(x_1^{(n)})] &= \mathrm{var}_\pi[E_\pi[\cdots E_\pi[E_\pi\{t(x_1) \mid x_2\} \mid x_1] \mid \cdots]], \quad (3) \\ \mathrm{cov}[s(x_2^{(0)}), s(x_2^{(n)})] &= \mathrm{var}_\pi[E_\pi[\cdots E_\pi[E_\pi\{s(x_2) \mid x_1\} \mid x_2] \mid \cdots]], \quad (4) \end{aligned}$$

where the right hand sides of both (3) and (4) have $n$ expectation signs conditioned alternately on $x_1$ and $x_2$. These formulas were then used to compare different imputation schemes and to show that Rao-Blackwellization *always* increases efficiency of Monte Carlo estimates.

### 4.3 An example

Efron and Morris (1975) used empirical Bayes method to analyze data of the first 45 at-bats in the middle of a season for $n = 18$ major league players (shown in column 2 of Table1). They estimated the 18 "true" betting probabilities based on this data set, and then used them as predictions of each person's betting average for the remainder of the season. Here we apply a hierarchical Bayes model and data augmentation for the same task. Let $Y_i$ denote the observed betting average (column two in the table) in the first 45 at bats of the $i$th person, and let $p_i$ denote his true betting percentage. A variance-stabilizing transformation of $Y_i$ was first performed in Efron and Morris (1975):

$$X_i = \sqrt{45}\ \arcsin(2Y_i - 1), \quad \text{and let} \quad \theta_i = \sqrt{45}\ \arcsin(2p_i - 1).$$

Then approximately $X_i$ can be regarded as a $N(\theta_i, 1)$ random variable. To build a hierarchical model, we assume that the $\theta_i$ are iid from $N(\mu, \sigma^2)$. Furthermore, we assume that the prior distribution for $\mu$ and $\sigma$ is uniform on $(-\infty, \infty) \times (0, \infty)$, thus improper. As an exercise, the reader may try out with other priors, but note that the prior for $\sigma$ can not be singular at 0.

We implemented a Gibbs sampler for the problem which iterates the following 2 steps:
— Draw $\theta_i$, $i = 1, \ldots, 18$, conditional on $\mu$ and $\sigma^2$.
— Draw $\mu$ and $\sigma^2$ conditional on all the values of $\theta_i$.
Figure 2 shows the posterior density of $\mu$ via Gibbs sampling approximation as well as the shrinkage

Table 1: Batting Averages and Their Estimates

| Player | Batting average for first 45 at-bats | Batting average for remainder | Stein's estimator | Efron-Morris's estimator |
|---|---|---|---|---|
| 1 | .400 | .346 | .290 | .334 |
| 2 | .378 | .298 | .286 | .313 |
| 3 | .356 | .276 | .281 | .292 |
| 4 | .333 | .222 | .277 | .277 |
| 5 | .311 | .273 | .273 | .273 |
| 6 | .311 | .270 | .273 | .273 |
| 7 | .289 | .263 | .268 | .268 |
| 8 | .267 | .210 | .264 | .264 |
| 9 | .244 | .269 | .259 | .259 |
| 10 | .244 | .230 | .259 | .259 |
| 11 | .222 | .264 | .254 | .254 |
| 12 | .222 | .256 | .254 | .254 |
| 13 | .222 | .303 | .254 | .254 |
| 14 | .222 | .264 | .254 | .254 |
| 15 | .222 | .226 | .254 | .254 |
| 16 | .200 | .285 | .249 | .249 |
| 17 | .178 | .316 | .244 | .233 |
| 18 | .156 | .200 | .239 | .208 |

estimates of $p_i$. In fact this Gibbs procedure can be further modified to improve efficiency. See Liu (1994), Gelfand et al. (1995), and Liu and Sabatti (1998b) for more discussions.

# 5 Special Markov Chain Monte Carlo Algorithms

To illustrate how the Metropolis-Hastings rule and the Gibbs sampler are adopted in practice, we now describe a few such algorithms that have appeared frequently in the literature.

## 5.1 The independence chain

A special choice of the proposal transition $T(x, y)$ in the Metropolis-Hastings algorithm is an independent trial density $p(y)$. That is, the proposed move $y$ is generated from $p(\cdot)$ independent of the previous state $x^{(t)}$. This method, as first proposed by Hastings (1970), appears to be an alternative to the rejection sampling and importance sampling. Its convergence properties was studied in Liu (1996a).

*Metropolized Independence Sampler (MIS):* given current state $x^{(t)}$,

- Draw $y \sim p(y)$
- Simulate $u \sim$ Uniform[0,1] and let

$$x^{(t+1)} = \begin{cases} y, & \text{if } u \leq \min\left\{1, \frac{w(y)}{w(x^{(t)})}\right\} \\ x^{(t)}, & \text{Otherwise.} \end{cases}$$

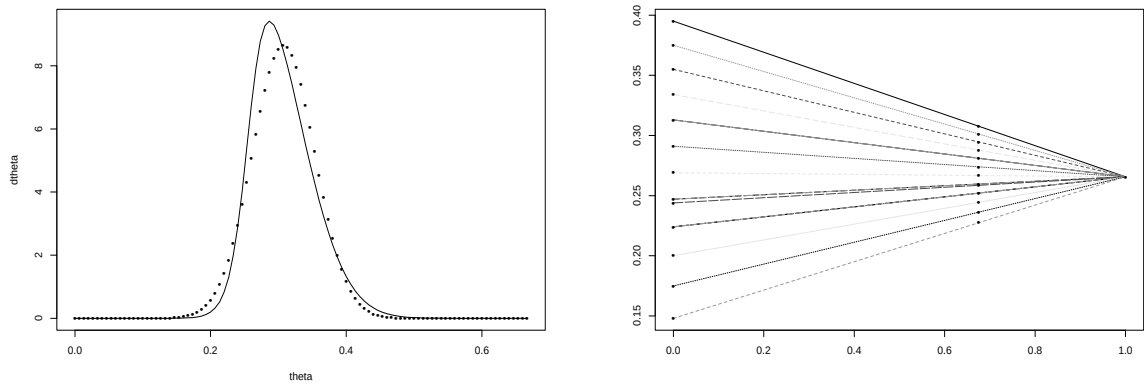



Figure 2: Left: Posterior density of $\mu$ — Gibbs sampling (solid) versus normal (dotted) approximation. The solid curve is almost indistinguishable from the true posterior of $\mu$. Right: a graphical view of how shrinkage estimates are related to their respective MLE's.

where $w(x) = \pi(x)/p(x)$ is the usual *importance sampling weight*.

As with the rejection method, the efficiency of MIS depends on how close the trial density $p(y)$ is to the target $\pi$. To ensure robust performance, it is advisable to let $p(\cdot)$ be a relatively long-tailed distribution. Tierney (1994) and Gelman and Rubin (1992) suggested that one can insert a couple of MIS steps into Gibbs iteration when correctly sampling from a conditional distribution is difficult. The idea is useful in many Bayesian computations in which each conditional density can be approximated reasonably well by a Gaussian distribution. To accommodate irregular tail behaviors, it is essential to use a long-tailed $t$-distribution as $p(x)$.

### 5.2 Random-walk Metropolis

This is exactly the algorithm we have used for simulating position distribution of the six balls in a box. Suppose $\pi(x)$ is defined on $R^d$ and is of interest. Given current state $x^{(t)}$, the algorithm iterates as follows:

- Draw $\epsilon \sim g_\sigma$ and set $y = x^{(t)} + \epsilon$. where $g_\sigma$ is a spherically symmetric distribution and $\sigma$ can be controlled by the user. An often used distribution is $N(0, \sigma^2 I)$.
- Simulate $u \sim$ Uniform[0,1] and update

$$x^{(t+1)} = \begin{cases} y, & \text{if } u \le \pi(y)/\pi(x^{(t)}), \\ x^{(t)}, & \text{Otherwise.} \end{cases}$$

Gelman, Roberts, and Gilks (1995) gave some interesting results and heuristic guidance on how to choose $\sigma$ so as to achieve fast convergence.

### 5.3 Hit-and-run algorithm

For a given current sample $x^{(t)}$ one does the following: (a) uniformly select a random direction $\boldsymbol{e}^{(t)}$; (b) sample a scalar $r^{(t)}$ from density $f(r) \propto \pi(x^{(t)} + r\boldsymbol{e}^{(t)})$; and (c) update $x^{(t+1)} = x^{(t)} + r^{(t)}\boldsymbol{e}^{(t)}$. This algorithm behaves like a random-direction Gibbs sampler and allows for a complete exploration of a randomly chosen direction. It tends to be especially helpful when there are severl modes (with comparable sizes) in the target distribution.

A main difficulty in implementing the algorithm, however, is that one is rarely able to draw from $f(r)$ in practice. Then s/he may end up only using a single step of Metropolis update (Chen and Schmeiser 1993) — which renders the algorithm equivalent to the random-walk Metropolis.

### 5.4 Adaptive directional sampling

Gilks, Roberts and George (1994) proposed a multiple-chain MCMC method, *adaptive directional sampling* (ADS), which allows for the exchange of information across different chains. At each iteration of the ADS (or snooker algorithm), one has a population of samples, say $\mathcal{S}^{(t)} = \{X_1^{(t)}, \ldots, X_m^{(t)}\}$, of size $m$. Then the next generation $\mathcal{S}^{(t+1)}$ is produced as follows: (a) a member $X_c^{(t)}$ from $\mathcal{S}^{(t)}$ is selected at random; (b) a random direction $\boldsymbol{e}^{(t)}$ is generated as $\boldsymbol{e}^{(t)} = (X_c^{(t)} - X_a^{(t)})/\|X_c^{(t)} - X_a^{(t)}\|$, where the anchor point $X_a^{(t)}$ is chosen at random from $\mathcal{S}^{(t)} \backslash \{X_c^{(t)}\}$; (c) a scalar $r^{(t)}$ is generated from an appropriate distribution $f(r)$; and, finally, (d) update $X_c^{(t+1)} = X_a^{(t)} + r^{(t)}\boldsymbol{e}^{(t)}$, and $X_j^{(t+1)} = X_j^{(t)}$ for $j \neq c$. Gilks et al. (1994) and Roberts and Gilks (1994) show that $f(r)$ should be of the form

$$f(r) \propto |r|^{k-1}\pi(X_a^{(t)} + r\boldsymbol{e}^{(t)}).$$

They also gave a more general form of this algorithm and provided cautionary advice on its use. Again, a main difficulty for using the ADS in practice is that sampling from $f(r)$ is often infeasible. Liu, Liang and Wong (1998) proposed a way to overcome it.

### 5.5 Slice sampler

Suppose $\pi(x)$ is a density function of interest and $x \in R^d$. Then drawing $x \sim \pi(x)$ is equivalent to generating $z = (z_1, \ldots, z_{d+1})$ so that it is uniformly distributed in the region $S$ under the surface of $\pi$, i.e., $S = \{z \in R^{d+1} : \ z_{d+1} \leq \pi(z_1, \ldots, z_d)\}$. However, generating uniformly distributed random variables in an arbitrary region is equally difficult as any other simulation problem. One can apply the following Gibbs iteration to achieve the sampling:

- draw $y^{(t+1)} \sim \text{Unif}\,[0, \pi(x^{(t)})]$ ;
- draw $x^{(t+1)}$ uniformly from region $S^{(t+1)} = \{x : \ \pi(x) \geq y^{(t+1)}\}$.

However, region $S^{(t+1)}$ in the iteration is still difficult to deal with. When $\pi$ can be written as the product of $k$ functions, i.e., $\pi(x) = f_1(x) \times \cdots \times f_k(x)$, Edwards and Sokal (1988) introduced $k$ auxiliary variables $y_1, \ldots, y_k$ and described a Gibbs sampler for sampling $(x, y_1, \ldots, y_k)$ uniformly over the region $0 < y_i < f_i(x), \ i = 1, \ldots, k$:

- Draw $y_i^{(t+1)} \sim \text{Unif}[0, f_i(x^{(t)})], \ i = 1, \ldots, k$.
- Draw $x^{(t+1)}$ uniformly from region $S^{(t+1)} = \bigcap_{i=1}^{k} \{x : \ f_i(x) \geq y_i^{(t+1)}\}$.

Damien, Wakefield and Walker (1997) showed that in many cases one can find a decomposition of $\pi$ so that the intersection set $S^{(t+1)}$ is easy to compute, which leads to an easily implemented sampler. But others noticed that its convergence may be slowed by the presence of many auxiliary variables. Applications of this approach to image analysis have been discussed by Besag and Green (1993) and Higdon (1996).

### 5.6 Metropolized Gibbs sampler

When the state space of interest is discrete, Liu (1996b) suggested a way to improve the ordinary Gibbs sampler by using an "over-relaxation" and proved its superiority.

Suppose that $X = (X_1, \ldots, X_d)$, where $X_i$ takes $m_i$ possible values, and that $\pi(x)$ is the distribution of interest. In the random scan Gibbs sampler described in Section 4.1, a coordinate $i$ is first randomly chosen and the current value $x_i$ is replaced by a value $y_i$ drawn from the corresponding full conditional distribution. Here we consider a modification of the above procedure in which a value $y_i$, different from $x_i$, is drawn with probability

$$\frac{\pi(y_i \mid x_{[-i]})}{1 - \pi(x_i \mid x_{[-i]})},$$

then $y_i$ replaces $x_i$ with the Hastings (1970) acceptance probability,

$$\min\left\{1, \frac{1 - \pi(x_i \mid x_{[-i]})}{1 - \pi(y_i \mid x_{[-i]})}\right\},$$

else $x_i$ is retained. Liu (1996b) proves that the modified Gibbs sampler for discrete random variables as defined above is statistically more efficient than the random scan Gibbs sampler.

When $m_i = 2$, the Gibbs sampler is essentially Barker's (1965) method, whereas the modified procedure becomes a Metropolis et al. (1953) algorithm. Peskun (1973) makes some general comparisons between these two samplers. Besag et al. (1995) note that the superiority of Metropolis for binary systems results from its increased mobility around the state space. This rationale applies more generally to the modified Gibbs sampler.

# 6 Convergence Diagnosis

Our view on this issue concurs with that of Cowles and Carlin (1995): a combination of Gelman and Rubin (1992) and Geyer (1992) can usually provide an effective, yet simple, method for monitoring convergence in MCMC sampling. Many other approaches, which typically consume a few times more computing time, can only provide marginal improvements. The *perfect simulation* method recently proposed by Propp and Wilson (1996) is an exciting theoretical breakthrough and its value in assessing convergence of MCMC schemes has been noticed (Robert 1998). But the method is still not ready (i.e., general and practical enough) for a routine use for MCMC computation. Interested reader may find Robert (1998) an inspiring reference, which provided an extensive study on convergence diagnostics.

Based on normal theory approximations to the target distribution $\pi(x)$, Gelman and Rubin (1992) proposed a method that involves the following steps:

1. Before sampling begins, obtain a simple "trial" distribution $f(x)$ which is over-dispersed relatively to the target distribution $\pi$. Generate $m$ (say, 10) iid samples from $f(x)$.

2. Start $m$ independent samplers with starting values obtained in Step 1. Run each chain for $2n$ iterations.

3. For a scalar quantity of interest (after appropriate transformation to approximate normality), say $\theta$, we use the sample from the last $n$ iterations to compute $W$, the average of $m$ *within-chain* variances, and $B$, the variance between the means $\theta$ from the $m$ parallel chains.

4. Compute the "shrink factor"

$$\sqrt{\hat{R}} = \sqrt{\left(\frac{n-1}{n} + \frac{m+1}{mn}\frac{B}{W}\right)\frac{df}{df-2}}$$

   Here $df$ refers to the degree of freedom in a t-distribution approximation to the empirical distribution of $\theta$.

Gelman and Rubin (1992) suggested to use $\theta = \log \pi(x)$ as a general diagnosis benchmark. Other choices of $\theta$ have been reviewed in Cowles and Carlin (1995).

Geyer's (1992) main criticism to Gelman and Rubin's approach is that for difficult MCMC computation, one should concentrate all the resources to a single chain iteration: the latter 9000 samples from a single run of 10,000 iterations is much more likely to come from the target distribution $\pi$ than those samples from 10 parallel runs of 1,000 iterations. In addition, good convergence criterion such as autocorrelation time used by physics can be produced with a single chain.

Concerning generic use of MCMC methods, we advocate a variety of diagnostic tools rather than any single plot or statistic. In our own work, we often run a few (3 to 5) parallel chains

with relatively scattered starting points. Then we inspect these chains by comparing many of their aspects, such as the histogram of some parameters, autocorrelation plots, and Gelman-Rubin's $\hat{R}$.

# 7 Towards More Efficient MCMC Sampler and Optimizer

In this section, we describe a few innovative ideas built upon the fundamental MCMC framework for global optimization and for more efficient Monte Carlo simulations. Some of these ideas have made tremendous impact in many scientific research areas and computer industry.

## 7.1 Simulated annealing

In condensed matter physics, *annealing* is known as a thermal process for obtaining low energy states of a solid in a *heat bath*. The process has two steps:

- Raise the temperature of the heat bath high enough for the solid (metal) to melt.
- Decrease carefully the temperature of the heat bath until the particles arrange themselves in the ground state of the solid (i.e., crystallize) .

At high temperature phase, the solid metal becomes a liquid and all its particles can "flow" around to rearrange themselves freely; whereas at low temperature the particles are gradually forced to "line-up" so as to attain the lowest-energy state. Realizing that the Metropolis algorithm can be used to simulate particle movements at various temperature to reach *thermal equilibrium*, Kirkpatrick, Gelatt and Vecchi (1983) proposed a computer imitation of the annealing process, called the *simulated annealing* (SI), and applied it to solve combinatorial optimization problems.

*The Algorithm.* Suppose our task is to find the *minimum* of a target function $h(x)$. This is equivalent to finding the maximum of $\exp\{-h(x)/T\}$ at any given *"temperature"* $T$. Let $T_1 > T_2 > \cdots > T_k > \cdots$ be a sequence of monotone decreasing temperatures in which $T_1$ is reasonably large and $\lim_{k\to\infty} T_k = 0$. At each temperature $T_k$, we run $N_k$ iterations of a Metropolis-Hastings or the Gibbs sampler, with $\pi_k(x) \propto \exp\{-h(x)/T_k\}$ as its equilibrium distribution. Because as $k$ increases $\pi_k$ puts more and more of its probability mass (converging to 1) into a vicinity of the global maximum of $h$, we will almost surely be in a vicinity of the global optimum if the number of M-H iterations $N_k$ is sufficiently large. Algorithmically, we do the following:

- Initialize at an arbitrary configuration $x_0$ and temperature level $T_1$.
- For each $k$, we run $N_k$ steps of MCMC iterations with $\pi_k(x)$ as its target distribution. Pass the final configuration of $x$ to the next iteration.
- Increase $k$ to $k+1$.

It can be shown that the global maximum can be reached by SA with probability 1 if temperature $T_k$ decreases sufficiently slowly, i.e., at the speed of order $O(\log(L_k)^{-1})$, where $L_k = N_1 + \cdots + N_k$ (Geman and Geman 1984). In practice, no one can afford to have such a slow annealing schedule. Most frequently people use a linear or even exponential temperature decreasing schedule, which can no longer guarantee that the global optimum will be reached. However, many researchers' experiences during the past fifteen years have testified that the SA is a very attractive general-purpose optimization tool. See Aarts and Korst (1989) for further analysis.

### 7.2 Simulated tempering and parallel tempering

To increase mixing rate of a MCMC scheme, Marinari and Parisi (1992) and Geyer and Thompson (1995) proposed a technique, *simulated tempering* (ST), in the same spirit as simulated annealing. To implement ST, one first constructs a family of distributions $\Pi = \{\pi_i(x) \;\; i \in I\}$ by varying a single parameter, the *temperature*, in the target distribution $\pi$. Distribution $\pi$ corresponds to the member of this family with the highest temperature. Then a new target distribution, $\pi_{\rm st}(x, i) \propto c_i \pi_i(x)$, is defined on the augmented space $(x, i) \in \mathcal{X} \times I$. Here $c_i$ is a controllable constant, whose role is to allow each temperature level to have reasonable chance of being visited. Finally, a MCMC sampler is used to draw samples from $\pi_{\rm st}$. The intuition behind ST is that by heating up the distribution repeatedly, the new sampler can escape from a local mode and increase its mixing rate. Initiated with $i^{(0)} = 0$ and any $x^{(0)}$ in the space of interest, the ST algorithm consists of the following steps:
*ST Algorithm.* With the current state $(x^{(t)}, i^{(t)}) = (x, i)$, we draw $u \sim$ Unif [0,1].

- If $u \leq \alpha_0$, we let $i^{(t+1)} = i$ and let $x^{(t+1)}$ be drawn from a MCMC transition $T_i(x, x^{(t+1)})$ that leaves $\pi_i$ invariant.
- If $u > \alpha_0$, we let $x^{(t+1)} = x$ and propose a temperature transition $i^{(t)} \to i'$ (usually a simple nearest-neighbor random walk with reflecting boundary), and let $i^{(t+1)} = i'$ with probability $\min\left\{1, \frac{c_{i'}\pi_{i'}(x)}{c_i \pi_i(x)}\right\}$; otherwise let $i^{(t+1)} = i$.

In order for ST to work well, the two adjacent distributions $\pi_i$ and $\pi_{i+1}$ need to have sufficient overlap and the $\alpha_i$ need to be tuned carefully. This requirement sometimes demands one to prescribe many temperature levels which adversely affect the efficiency of the algorithm. For optimization purpose, we have applied a *relaxed* version of the ST in a VLSI design problem and obtained good results (Cong et al. 1999). Dynamic weighting method described in the next section can also be used to overcome steep energy barriers encountered in temperature transitions.

*Parallel tempering* (Geyer 1991) is an interesting and powerful twist of the ST. Instead of augmenting $\mathcal{X}$ to $\mathcal{X} \times I$, Geyer suggested augmenting $\mathcal{X}$ to a product space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_I$, where the $\mathcal{X}_i$ are identical copies of $\mathcal{X}$. Suppose $(x_1, \ldots, x_I) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_I$. For the family of distributions

$\Pi = \{\pi_i, i = 1, \ldots I\}$, we define a joint probability distribution on the product space as

$$\pi_{\text{pt}}(x_1, \ldots, x_I) = \prod_{i \in I} \pi_i(x_i),$$

and run parallel MCMC schemes on each space $\mathcal{X}_i$. An "index swapping" operation is conducted in the place of temperature transition in ST. The PT algorithm can be more rigorously defined as follows: suppose the current state is $(x_1^{(t)}, \ldots x_I^{(t)}) \in \times_{i \in I} \mathcal{X}$. Draw $u \sim$ Unif [0,1].

- If $u \leq \alpha_0$, we conduct the *parallel step*. That is, we update every $x_i^{(t)}$ to $x_i^{(t+1)}$ via their respective MCMC scheme.
- If $u > \alpha_0$, we conduct the *swapping step*. That is, we randomly choose a neighboring pair, say $i$ and $i+1$, and propose to "swap" $x_i^{(t)}$ and $x_{i+1}^{(t)}$. Accept the swap with probability

$$\left\{ 1, \ \frac{\pi_i(x_{i+1}^{(t)})\pi_{i+1}(x_i^{(t)})}{\pi_i(x_i^{(t)})\pi_{i+1}(x_{i+1}^{(t)})} \right\}.$$

This scheme is very powerful in simulating complicated systems such as bead polymers and other molecular structures. It has also been very popular in dealing with statistical physics models (Hukushima and Nemoto 1996). Compared with the ST, PT does not need fine tuning (to adjust normalizing constants $\alpha_i$) and can utilize information in multiple MCMC chains.

### 7.3 Dynamic weighting in MCMC

Wong and Liang (1997) introduced the use of a dynamic weighting variable for controlling Markov chain simulation. By using this scheme, they were able to obtain better results for many optimization problems such as the traveling salesman problem and neural network training; and high-dimensional integration problems such as the Ising model simulation.

To start a dynamic weighting scheme, we first augment the sample space $\mathcal{X}$ to $\mathcal{X} \times R^+$ so as to include a weight variable. Similar to the Metropolis algorithm, we also need a proposal function $T(x, y)$ on the space $\mathcal{X}$. Suppose at iteration $t$ we have $(x^{(t)}, w^{(t)}) = (x, w)$. Then an *R-type Move* is defined as

- Draw $Y = y$ from $T(x, y)$ and compute the Metropolis ratio $r(x, y) = \frac{\pi(y)T(y,x)}{\pi(x)T(x,y)}$.
- Choose $\theta = \theta(w, x) > 0$, and draw $U$ from uniform(0,1). Then let

$$(x^{(t+1)}, w^{(t+1)}) = \begin{cases} (y, wr(x, y) + \theta), & \text{if } U \leq \frac{wr(x,y)}{wr(x,y)+\theta}; \\ (x^{(t)}, \frac{w(wr(x,y)+\theta)}{\theta}), & \text{Otherwise.} \end{cases} \tag{5}$$

It is easy to check the R-type move does not have $\pi$ as its equilibrium distribution. Wong and Liang (1997) propose to use *invariance with respect to importance-weighting* (IWIW) for justifying the above scheme. That is, if the joint distribution of $(x, w)$ is $f(x, w)$ and $x$ is said *correctly weighted* by $w$ with respect to $\pi$ if $\sum_w wf(x, w) \propto \pi(x)$. A transition rule is said to satisfy IWIW if it *maintains* the *correctly-weightedness* for the joint distribution of $(x, w)$. Clearly, the R-type move satisfies IWIW.

The purpose of introducing importance weights into dynamic Monte Carlo process is to provide a means for the system to make large transitions not allowable by the standard Metropolis transition rules. The weight variable is updated in a way that allows for an adjustment of the bias induced by such non-Metropolis moves.

### 7.4 Reversible jump

In applications such as Bayesian model selections (Green 1995), one often need to have a sampler that jumps between different dimensional spaces. In principle, one still can follow the Metropolis-Hastings's rule to guide for the design of such a sampler.

Suppose $\mathcal{X}$ corresponds to a higher dimensional space and $\mathcal{Y}$ to a lower one. To communicate between the two spaces, one needs to have two "proposals," one for $\mathcal{Y} \to \mathcal{X}$ and another for $\mathcal{X} \to \mathcal{Y}$. Since $\mathcal{Y}$ is of lower dimensional, any transition from $\mathcal{Y}$ to $\mathcal{X}$ must have a degenerate density with respect to the dominant measure of $\mathcal{X}$, implying that not all the moves from $\mathcal{X} \to \mathcal{Y}$ can be "reversed" by the transition from $\mathcal{Y}$ to $\mathcal{X}$. To overcome this difficulty, one must have a " matching space" $\mathcal{Z}$, so that $\mathcal{Y} \times \mathcal{Z}$ has the same dimension as $\mathcal{X}$, and a matching sampling distribution $p(z \mid y)$. With the matched space, one can come up with two nondegenerate proposals, $T_1 : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ and $T_2 : \mathcal{X} \to \mathcal{Y} \times \mathcal{Z}$, and follows the Metropolis-Hastings's rule to design jumps.

For example, to jump from $\mathcal{Y}$ to $\mathcal{X}$, we first draw $z \in \mathcal{Z}$ that follows $p(z \mid y)$ and then draw $x'$ from $T_1[(y, z), \cdot]$. Accept the move with probability

$$\alpha = \min \left\{ 1, \frac{T_2[x', (y, z)]}{T_1[(y, z), x']g_1(z \mid y)} \right\}.$$

Jumping from $\mathcal{X}$ to $\mathcal{Y}$ is achieved by proposing $x \to (y', z')$ by $T_2$ and accepting it with

$$\beta = \min \left\{ 1, \frac{T_1[(y'z'), x]g_1(z' \mid y')}{T_2[x, (y', z')]} \right\}.$$

Green (1995) presented a more formal treatment of this type of move and named them *reversible jumps*. A method for combining the reversible jump and simulated tempering to speed up MCMC sampling was discussed in Liu and Sabatti (1998b).

### 7.5 Multigrid Monte Carlo and generalized Gibbs

The multigrid method was first developed in the field of computational mathematics for solving partial differential equations (McCormick 1989). Its main idea is to alternately apply iterative algorithm (such as Gauss-Seidel method) on different grid-discretizations of the space. In doing so, the slow-varying components of the error can be damped more rapidly during "coarser-grid" iterations, whereas the high frequency components can be removed during "finer-grid" iterations. Goodman and Sokal (1989) applied the idea to Monte Carlo computation for statistical physics models and named it *multigrid Monte Carlo* (MGMC). They translated the multigrid idea into a way of designing progressively more global moves. Instead of updating one component a time as in the Gibbs sampler, MGMC suggests moving several highly correlated ones simultaneously along certain subspace. Liu and Sabatti (1998a,b) generalized this key step of MGMC for statistical applications.

Liu and Wu (1999) discovered that MGMC is in fact a generalization of the Gibbs sampler and each move in MGMC can be understood from a transformation group viewpoint. Let $\mathcal{G}$ be a locally compact group of transformations on the space of $x$. Starting from an initial point $x^{(0)}$, a *group move* from $x^{(t)}$ to a new point is achieved by first drawing a $g \in \mathcal{G}$ from the *conditional distribution*

$$p(g) \propto \pi(g(x^{(t)}))|J_g(x^{(t)})|H(dg),$$

where $J_g$ is the Jacobian of the transformation and $H$ is the left Haar measure for $\mathcal{G}$, and then letting $x^{(t+1)} = g(x^{(t)})$. It can be shown that this move leaves $\pi$ invariant. Gibbs sampling update corresponds to using a translation group on one coordinate a time.

### 7.6 Multiple-Try Metropolis (MTM)

We end this section with an interesting generalization of the Metropolis-Hastings's transition rule. This new rule (Frenkel and Smit 1996; Liu et al. 1998) enables a MCMC sampler to make large step-size jumps. It is particularly useful when one identifies certain directions of interest but has difficulty to implement a Gibbs-sampling type move because of unfavorable conditional distributions. Therefore, the MTM can be readily combined with the hit-and-run and ADS algorithms (Section 5). The following two versions are similar but *not equivalent* even when $T$ is symmetric.

*Algorithm (I):*

- Draw $k$ trials $y_1, \ldots, y_k$ from the proposal distribution $T(x, y)$. Compute

$$g(x, y_j) = \pi(x)T(x, y_j) \tag{6}$$

  and $g(y_j, x)$, for $j = 1, \ldots, k$.

- Select $Y = y_l$ among the $y$'s with probability proportional to $g(y_j, x)$, $j = 1, \ldots, k$. Then draw $x_1^*, \ldots, x_{k-1}^*$ from the distribution $T(y_l, x^*)$, and let $x_k^* = x$.

- Accept $y_l$ with probability

$$\min\left\{1, \frac{g(y_1, x) + \cdots + g(y_k, x)}{g(x_1^*, y_l) + \cdots + g(x_k^*, y_l)}\right\}$$

  and reject with the remaining probability.

*Algorithm (II):*

- Draw $k$ trials $y_1, \ldots, y_k$ from a *symmetric* proposal distribution $T(x, y)$.

- Select $Y = y_l$ among the $y$'s with probability proportional to $\pi(y_j)$, $j = 1, \ldots, k$. Then draw $x_1', \ldots, x_{k-1}'$ from the distribution $T(y_l, x')$. Denote $x_k' = x$.

- Accept $y_l$ with probability

$$\min\left\{1, \frac{\pi(y_1) + \cdots + \pi(y_k)}{\pi(x_1') + \cdots + \pi(x_k')}\right\}$$

  and reject with the remaining probability.

It can be easily shown that the two new transitions also satisfy the detailed balance condition, thus, induce reversible Markov chains with $\pi(x)$ as its invariant distribution.

# 8 Final Remarks

Given the limited space, we are only able to provide a very sketchy, and perhaps very biased, review of recent developments in Markov chain Monte Carlo methodology. There are a lot of interesting new theoretical and methodological developments we are unable to cover. Among those Monte Carlo methods that are left out from this article, *sequential importance sampling* is perhaps the most significant recent development. Interested reader is referred to a recent article by Liu and Chen (1998). They provided a general framework of the methodology and reviewed connections between some related methods, namely, the *bootstrap filter* (Gorden et al. 1993) and sequential imputation with rejuvenation (Kong et al. 1994; Liu and Chen 1995). Much of theoretical work on convergence rates of MCMC algorithms and on convergence diagnostics are omitted. A lot of useful Monte Carlo techniques developed by physicists, biochemists, and structural biologists are not included. Connections of MCMC with neural network training and with genetic algorithms are not commented on. Almost all applications of MCMC are left to the reader for further reading.

With all these defects, we still hope that the reader will find MCMC methodology exciting and the methods described in this article useful. We also hope that some of the reader will join us

in the effort to discover new and more efficient MCMC techniques and to understand theoretical properties of them.

## References


Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines.* Wiley: New York.

Alder, B.J. and Wainwright, T.E. (1959). Studies in molecular dynamics. *J. Chem. Phys.* **31**, 459-466.

Barker, A.A. (1965). Monte Carlo calculations of the radial distribution functions for a proton-electron plasma. *Austral. J. Phys.* **18**, 119-133.

Besag, J. and Green, P. (1993). Spatial statistics and Bayesian computation (with discussion). *J. Roy. Statist. Soc.* B **55**, 25-37.

Besag, J., Green, P., Higdon, D. & Mengersen, K. (1995). Bayesian computation and stochastic systems (with discussion). *Statist. Sci.* **10**, 3-66.

Chen, M.-H. and Schmeiser, B.W. (1993). Performances of the Gibbs, hit-and-run, and Metropolis samplers. *J. Comput. Graph. Statist.* **2**, 251-272.

Cong, J., Kong, T., Xu, D., Liang, F., Liu, J.S., and Wong, W.H. (1999). Simulated Tempering for VLSI floorplan designs. *Proc. of Asia and South Pacific Design Automation Conference*, Accepted.

Cowles, M.K. and Carlin, B.O. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative study. *J. Amer. Statist. Assoc.* **91**, 883-904.

Damien, P., Wakefield, J., and Walker, S. (1997). Gibbs sampling for Bayesian nonconjugate and hierarchical models using auxiliary variables. *Technical Report*, Business School, University of Michigan.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum Likelihood Estimation from Incomplete Data Via the EM Algorithm (with Discussion). *J. Roy. Statist. Soc.* B **39**, 1-38.

Diaconis, P. (1988). *Group Representations in Probability and Statistics.* Lecture Notes-Monograph Series **11**, IMS, Hayward, California.

Efron, B. and Morris, C.N. (1975). Data analysis using Stein's estimator and its generalizations. *J. Amer. Statist. Assoc.* **70**, 311-319.

Frenkel, D. and Smit, B. (1996). *Understanding Molecular Simulation.* Academic Press: New York.

Frigessi, A., Hwang, C.-R., di Stefano, P. & Sheu, S.-J. (1993). Convergence rates of the Gibbs sampler, the Metropolis algorithm and other single-site updating dynamics. *J. Roy. Statist. Soc.* B **55**, 205-219.

Gelfand, A.E., Sahu, S.K., and Carlin, B.P. (1995). Efficient Parameterizations for Normal Linear Mixed Models. *Biometrika* **82**, 479-88.

Gelfand, A.E. and Smith, A.F.M. (1990). Sampling-based Approaches to Calculating Marginal Densities. *J. Amer. Statist. Assoc.* **85**, 398–409.

Gelman, A. and Rubin, D.B. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statist. Sci.* **7**, 457-472.

Gelman, A. Roberts, G.O. and Gilks, W.R. (1995). Efficient Metropolis Jumping Rules. In *Bayesian Statistics V*, J.M. Bernardo et al (eds.), Oxford University Press.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. Pattn. Anal. and Mach. Intell.* **6**,721-741.

Geyer, C.J. (1991). Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface* (ed. E.M. Keramigas), 156-163. Fairfax: Interface Foundation.

— (1992). Practical Monte Carlo Markov chain (with discussion). *Statist. Sci.* **7**, 473-511.

Geyer, C.J. and Thompson, E.A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *J. Amer. Statist. Assoc.* **90**, 909-920.

Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. (1995). *Markov chain Monte Carlo in Practice.* New York: Chapman & Hall.

Gilks, W.R., Roberts, R.O., and George, E.I. (1994). Adaptive direction sampling. *The Statistician* **43**, 179-189.

Goodman, J. and Sokal, A.D. (1989). Multigrid Monte Carlo method. Conceptual foundations. *Physical Review D* **40** 2035-71.

Gordon, N.J., Salmon, D.J. and Smith, A.F.M. (1993). A novel approach to nonlinear/non Gaussian Bayesian state estimation. *IEE Proc. Radar and Signal Processing* **140**, 107-113.

Green, P.J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82**, 711-32.

Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97-109.

Higdon, D.M. (1998). Auxiliary variable methods for Markov chain Monte Carlo with applications. *J. Amer. Statist. Assoc.* **93**, 585-595.

Hukushima, K. and Nemoto, K. (1996). Exchange Monte Carlo method and application to spin glass simulations. *J. Phys. Soc. Jpn.* **65**, 1604-1608.

Karplus, M. and Petsko, G.A. (1990). Molecular dynamics simulations in biology. *Nature* **347**, 631-639.

Kirkpatrick, S., Gelatt Jr., C.D., and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science* **220**, 671-680.

Knuth, D. (1981) *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms.* Addison-Wesley, Reading.

Kong, A., Liu, J.S. and Wong, W.H. (1994). Sequential imputation method and missing data problems. *J. Amer. Statist. Assoc.* **89**, 278-288.

Lamoureux, C.G. and Witte, H.D. (1997). Empirical analysis of the yield curve: the information in the data viewed through the window of Cox, Ingersoll and Ross. *Technical Report*, Department of Finance, Univ. of Arizona.

Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A. and Wootton, J. (1993). Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262**, 208-214.

Leach, A.R. (1996). *Molecular Modelling: Principles and Applications.* Addison Wesley Longman: Singapore.

Liu, J.S. (1994). The collapsed Gibbs sampler with applications to a gene regulation problem. *J. Amer. Statist. Assoc.* **89**, 958-966.

— (1996a). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing* **6**, 113-119.

— (1996b). Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika* **83**, 681-682.

Liu, J.S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *J. Amer. Statist. Assoc.* **90**, 567-576.

— (1998). Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.* **93**, 1032-1044.

Liu, J.S., Liang, F., and Wong, W.H. (1998). The use of multiple-try method and local optimization in Metropolis sampling. *Tech. Rep.*, Department of Statistics, Stanford University.

Liu, J.S. and Sabatti, C. (1998a). Generalized multigrid Monte Carlo for Bayesian computation. *Technical Report*, Department of Statistics, Stanford University.

— (1998b). Simulated sintering: Markov chain Monte Carlo with spaces of varying dimensions (with Discussion). In *Bayesian Statistics* 6, J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith (eds). New York: Oxford University Press. *In press.*

Liu, J.S., Wong, W.H. and Kong, A. (1994). Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika* **81**, 27-40.

— (1995). Covariance structure and convergence rate of the Gibbs sampler with various scans. *J. Roy. Statist. Soc.* B **57**, 157-169.

Liu, J.S. and Wu, Y.N. (1998). Parameter expansion scheme for data augmentation. *J. Am. Statist. Assoc.*, tentatively accepted.

Marinari, E. and Parisi, G. (1992). Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters* **19**, 451.

Marsaglia, G. and Zaman, A. (1993). The KISS generator. *Tech. Rept.*, Dept. of Statistics, U. of Florida.

Marshall, A.W. (1956). The use of multi-stage sampling schemes in Monte Carlo computations. *symposium on Monte Carlo Methods*, 123-140, edited by M.A. Meyer, Wiley, New York.

McCormick, S.F. (1989). *Multilevel Adaptive Methods for Partial Differential Equations.* Society for Industrial and Applied Mathematics, PA.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953). Equations of state Calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087-1091.

Peskun, P.H. (1973). Optimal Monte Carlo sampling using Markov chains. *Biometrika* **60**, 607-612.

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

Robert, C.P. (1998). *Discretization and MCMC Convergence Assessment.* Springer: New York.

Roberts, G.O. and Gilks, W.R. (1994). Convergence of Adaptive Direction Sampling. *J. Mult. Anal.* **49**, 287-298.

Roberts, G.O. and Sahu, S.K. (1997). Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *J. Roy. Statist. Soc.* B **59**, 291-317.

Roberts, G.O. and Tweedie, R.L. (1996). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika* **83**, 95-110.

Rubin, D.B. (1987). *Multiple Imputation for Nonresponse in Surveys*, New York: Wiley.

— (1987). A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: the SIR algorithm. *J. Amer. Statist. Assoc.* **52**, 543-546.

Schervish, M.J. and Carlin, B. (1992). On the convergence of successive substitution sampling. *J. Comp. Graph. Statist.* **1**, 111-127.

Tanner, M.A. and Wong, W.H. (1987). The calculation of posterior distribution by data augmentation (with discussion). *J. Amer. Statist. Assoc.* **82**, 528-550.

Tierney, L.(1994). Markov chains for exploring posterior distributions (with discussion). *Ann. Statist.* **22**, 1701-1762.

Ullman, J.D. (1984). *Computational Aspects of VLSI*, Computer Science Press, Rockville (MD).

von Neumann, J. (1951). Various techniques used in connection with random digits. *Natl. Bureau of Standards Appl. Math. Ser.* **12**, 36-38.

Wong, W.H. and Liang, F. (1997). Dynamic Importance Weighting in Monte Carlo and Optimization. *Proc. Natl. Acad. Sci.* **94**, 14220-14224.