The Use of Multiple-Try Method and Local Optimization in Metropolis Sampling

Jun S. Liu, Faming Liang, and Wing Hung Wong [1]

Abstract

This article describes a new Metropolis-like transition rule, the *multiple-try Metropolis*, for Markov chain Monte Carlo (MCMC) simulations, which is especially useful for doing directional sampling. By using this new transition rule together with the adaptive direction sampling (Gilks, Roberts, and George 1994), we propose a novel approach for incorporating local optimization steps into any MCMC sampler for simulating random variables in continuous state-space. Numerical studies show that the new method performs significantly better than the traditional Metropolis-Hastings sampler. With minor tailoring in using the rule, the multiple-try method can also be exploited to achieve the effect of a Giddy Gibbs sampler without having to bear with griddy approximations, and the effect of a hit-and-run algorithm without having to figure out the required conditional distribution in a random direction.

KEYWORDS: Adaptive Direction Sampling; Biased Monte Carlo; Conjugate Gradient; Damped Sinusoidal; Gibbs Sampling; Griddy Gibbs Sampler; Hit-and-Run Algorithm; Markov Chain Monte Carlo; Metropolis Algorithm; Mixture Model.

1 Introduction

Markov chain Monte Carlo (MCMC) methods have been increasingly recognized by scientists as indispensable tools for difficult computational problems. They have also been central to many recent developments in statistical computing and statistical practice. A common feature of these Monte Carlo methods for simulating complex systems is that they rely on cumulative evolutions of small, albeit random, local changes. Such "local-changes" allow one to break a complex task into a series of manageable pieces. On the other hand, these local moves often lead to a very slow-converging algorithm. Among all the MCMC techniques, the Metropolis-Hastings transition (Metropolis et al. 1953; Hastings 1970) is the basic building block of local moves employed by many Markov-chain based simulation techniques; whereas the conditional

[1] Jun S. Liu is Assistant Professor, Department of Statistics, Stanford University, Stanford, CA 94305. Faming Liang is Postdoctoral Fellow and Wing Hung Wong is Professor, Department of Statistics, UCLA. Liu's research is supported in part by NSF grant DMS 95-96096 and the Stanford Terman fellowship. Wong's research is supported in part by NSF grant DMS-9703918.

updating of Gibbs sampling (Geman and Geman 1984) is a good alternative (less local) when the required conditional distributions are easy to sample from.

Let $\pi(x)$ be a high-dimensional density function and the task of interest is to draw random samples from it. In MCMC, it is often a powerful move if one can sample along certain directions, such as in a Gibbs sampler and a hit-and-run algorithm (Chen and Schmeiser 1993). However, an inherent difficulty is that these conditional distributions are often difficult to deal with. The adaptive rejection method of Gilks and Wild (1992) is powerful for log-concave densities but does not apply to others. Alternatively, one could draw from an approximation of the conditional distribution, such as the griddy approximation in the Griddy Gibbs algorithm (Ritter and Tanner 1994). But in this case there is no guarantee of the existence of the equilibrium distribution and it is difficult to assess differences between the output and the target distribution. A remedy of the problem is to incorporate the Metropolis rule in using the approximated distribution, but a good approximation is usually too expensive to come by and such approximates get worse very fast as dimension increases. The Multiple-Try Metropolis (MTM) method we present in the next section provides an effective means to conduct directional sampling without having to suffer the shortcomings in those previous approaches.

Another widely pursued approach for improving a MCMC sampler is via adaptation. That is, roughly speaking, one attempts to use information generated by the current iterations to guide future simulations. A particularly interesting idea is the *adaptive direction sampling* (ADS) proposed by Gilks, Roberts, and George (1994). They suggest the use of multiple MCMC chains and to adapt movements of one chain according to information from others. We describe how to combine the ADS framework with the MTM method to make use of information revealed by (deterministic) local optimization steps. Our numerical examples show that the new sampler offers significant improvement over the traditional Metropolis sampler, especially in difficult problems. A major advantage of the new algorithm is that it makes explicit use of local optimality information for adaptation and the companion MTM allows for a very large step-size in local movement.

This paper is arranged as follows. Section 2 introduces the basic MTM methodology and provides simple proofs for the correctness of the method; Section 3 presents our new sampler, the *conjugate-gradient Monte Carlo*; Section 4 shows a few other variations in using the MTM; Section 5 demonstrates the use of the method in several numerical examples; Section 6 concludes with a brief discussion.

2 General Methodology of Multiple-Try Metropolis

As in a standard Metropolis-Hastings algorithm, we let $T(x, y)$ be the proposal transition function. There are two versions of the MTM algorithm, depending on whether T is symmetric or not. We first state the general version where T may or may not be symmetric, and then describe its second version for symmetric T. Suppose $T(x, y)$ is the proposal function. Let the *current state* be $X_t = x$. In a MTM transition, the next state is generated as follows:

Multiple-Try Metropolis (I):

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

$$g(x, y_j) = \pi(x)T(x, y_j) \tag{1}$$

  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.

When T is symmetric, we have a slightly different version of the MTM.

Multiple-Try Metropolis (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.

Theorem 2.1 *The two MTM transition rules described above satisfy detailed balance and, hence, induce a reversible Markov chain with π as its equilibrium distribution.*

PROOF: We first show that MTM (I) satisfies detailed balance. Suppose $x \neq y$ and let $A(x, y)$ be the actual transition probability for moving from x to y. Without loss of generality, we let this y be y_k chosen in the MTM (I) sampler. Let $g(x, y) = \pi(x)T(x, y)$. We can explicitly write out $A(x, y)$ and derive that

$$\begin{aligned}
\pi(x)A(x,y) &= \pi(x) \int \cdots \int T(x,y)T(x,y_1)\cdots T(x,y_{k-1}) \frac{g(y,x)}{g(y,x)+\sum_{j=1}^{k-1} g(y_j,x)} \\
&\times \min\left\{1, \frac{g(y,x)+\sum_{j=1}^{k-1} g(y_j,x)}{g(x,y)+\sum_{j=1}^{k-1} g(x_j^*,y)}\right\} T(y,x_1^*)\cdots T(y,x_{k-1}^*) dy_1\cdots dy_{k-1} dx_1^* \cdots dx_{k-1}^* \\
&= g(x,y)g(y,x) \int \cdots \int \min\left\{\frac{1}{g(y,x)+\sum_j g(y_j,x)}, \frac{1}{g(x,y)+\sum_j g(x_j^*,y)}\right\} \\
&\times T(x,y_1)\cdots T(x,y_{k-1})T(y,x_1^*)\cdots T(y,x_{k-1}^*) dy_1 \ldots dy_{k-1} dx_1^* \cdots dx_{k-1}^* \qquad (2)
\end{aligned}$$

The expression (2) is apparently symmetric in x and y. Thus we proved that $\pi(x)A(x, y) = \pi(y)A(y, x)$, which is the detailed balance condition.

For MTM (II), we also let $A(x, y)$ be the actual transition function. A similar derivation to the foregoing one can be carried out:

$$\begin{aligned}
\pi(x)A(x,y) &= \pi(x) \int \cdots \int T(x,y)T(x,y_1)\cdots T(x,y_{k-1}) \frac{\pi(y)}{\pi(y)+\sum_{j=1}^{k-1} \pi(y_j)} \\
&\times \min\left\{1, \frac{\pi(y)+\sum_{j=1}^{k-1} \pi(y_j)}{\pi(x)+\sum_{j=1}^{k-1} \pi(x_j^*)}\right\} T(y,x_1^*)\cdots T(y,x_{k-1}^*) dy_1\cdots dy_{k-1} dx_1^* \cdots dx_{k-1}^* \\
&= \pi(x)\pi(y)T(x,y) \int \cdots \int \min\left\{\frac{1}{\pi(y)+\sum_j \pi(y_j)}, \frac{1}{\pi(x)+\sum_j \pi(x_j^*)}\right\} \\
&\times T(x,y_1)\cdots T(x,y_{k-1})T(y,x_1^*)\cdots T(y,x_{k-1}^*) dy_1 \ldots dy_{k-1} dx_1^* \cdots dx_{k-1}^* \qquad (3)
\end{aligned}$$

Again, it is easy to see that expression (3) is symmetric in x and y for the reason that $T(x, y) = T(y, x)$. Hence the transition of MTM (II) also satisfies the detailed balance. □

A precursor to the MTM method is the "orientational biased-Monte Carlo" described by Frenkel and Smit (1996), where they provided a specialized proof in the context of simulating molecular structures of materials. Although the comparison between MTM (I) and MTM (II) is similar to that between the Hastings (1970)'s algorithm and the Metropolis algorithm, it is especially interesting to note that the two versions of the MTM are *different* even when T is symmetric, whereas the Hastings's algorithm reduces to the Metropolis in this situation. The MTM transition allows one to explore more thoroughly in the "neighboring region" defined by $T(x, y)$, which is especially useful when such a region (or direction) is obtained from a relatively expensive adaptation method.

3 Using Local Optimization for Adaptation in MCMC

In this section we show how to combine the MTM with the ADS of Gilks et al. (1994) to produce a better sampler. Gilks et al. (1994)'s ADS method resembles the hit-and-run algorithm (see Section 4.1) but has its sampled direction, $\boldsymbol{e}_t$, determined by other previously sampled points. Another distinctive feature of the method is that it attempts to use information across multiple chains. We briefly describe a version of the ADS in the following subsection.

3.1 Adaptive Direction Sampling

At each iteration of the ADS (or snooker algorithm), one has a population of samples, say $\mathcal{S}_t = \{X_t^{(1)}, \ldots, X_t^{(m)}\}$, of size m. Then the next generation $\mathcal{S}_{t+1}$ is generated as follows: (a) a member $X_t^{(c)}$ from $\mathcal{S}_t$ is selected at random; (b) a random direction $\boldsymbol{e}_t$ is generated as $\boldsymbol{e}_t = (X_t^{(c)} - X_t^{(a)})/\|X_t^{(c)} - X_t^{(a)}\|$, where the anchor point $X_t^{(a)}$ is chosen at random from $\mathcal{S}_t \setminus \{X_t^{(c)}\}$; (c) a scalar r_t is generated from an appropriate distribution $f(r)$; and, finally, (d) update $X_{t+1}^{(c)} = X_t^{(a)} + r_t\boldsymbol{e}_t$, and $X_{t+1}^{(j)} = X_t^{(j)}$ for $j \neq c$. Gilks et al. (1994) and Roberts and Gilks (1994) show that $f(r)$ should be of the form

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

They also give a more general form of this algorithm and provide cautionary advice on the use of their algorithm. They particularly note that the adaptation may or may not improve the performance of the algorithm.

The ADS is a powerful formulation, but it leaves several issues unsettled. One question is how one can select a meaningful direction $\boldsymbol{e}_t$. We feel that a more-or-less arbitrary random choice of $\boldsymbol{e}_t$ can only bring in marginal improvement. In the following subsection, we demonstrate that if the choice of $\boldsymbol{e}_t$ is guided by a local optimization search, the resulting algorithm can be more effective. The second unanswered question as we mentioned in the introduction is how to sample from $f(r)$ effectively. We tackle this problem by using the MTM.

3.2 Local Optimization-Based MTM

We follow the ADS approach of evolving a population of samples, say, $\mathcal{S}_t = \{X_t^{(1)}, \ldots, X_t^{(m)}\}$, at each iteration. To update one of the sample, say, $X_t^{(c)}$, we use the other samples to construct a good reference point Y_t, and then update $X_t^{(c)}$ by a MTM transition along the direction defined by $X_t^{(c)}$ and Y_t. With the theory established by Roberts and Gilks (1994), one can see that essentially *any* way of choosing the reference point Y_t is appropriate provided that Y_t is independent of $X_t^{(c)}$ and that the distribution along the line, $f(r)$, is properly adjusted. For example, we can use a conjugate gradient search to construct the reference point. The algorithm is then specified as follows: suppose at time t we have a population of samples $\mathcal{S}_t = \{X_t^{(1)}, \dots, X_t^{(m)}\}$. At time $t+1$,

Conjugate-Gradient Monte Carlo:

- Randomly choose a member, say $X_t^{(c)}$, from $\mathcal{S}_t$.
- Randomly choose *another* member, say $X_t^{(r)}$, from $\mathcal{S}_t \backslash \{X_t^{(c)}\}$. Compute either the gradient or conjugate gradient of π at $X_t^{(r)}$ and denote this direction as $\boldsymbol{u}_t$. Conduct a *deterministic* line search to find the mode of π along $X_t^{(r)} + r\boldsymbol{u}_t$. Let this mode be the reference point Y_t.
- Let $\boldsymbol{e}_t = (Y_t - X_t^{(c)})/\|Y_t - X_t^{(c)}\|$, and sample along the line $Y_t + r\boldsymbol{e}_t$ by using the MTM method, with the target distribution for $r \in (-\infty, \infty)$ being

$$f(r) \propto |r|^{d-1}\pi(Y_t + r\boldsymbol{e}_t). \tag{4}$$

The conjugate-gradient local optimization procedure (Step 2) can be replaced by *any* effective local optimization method, such as the iterative conditional maximization (Besag, 1974), the gradient method, or a few EM steps. In all of our examples, we have used the conjugate gradient directional method coupled with a 1-dimensional minimization algorithm taken from Press et al. (1996), pp. 418.

The last step is realized by using MTM (I) or (II) with π substituted by $f(r)$. The population size needs not be too large. In fact we found that it is already quite satisfactory with $m = 2$. However, it should be a worthwhile topic to study the effect of m on the convergence of the algorithm. The following theorem shows that the CGMC is indeed a proper transition. Our proof is modeled after that in Roberts and Gilks (1994).

Theorem 3.1 *If we have population $\mathcal{S}_t = \{x_t^{(1)}, \dots, x_t^{(m)}\}$ in a CGMC setting, then the invariant distribution of $\mathcal{S}_t$ under the CGMC move is $\pi(x_t^{(1)}) \times \cdots \times \pi(x_t^{(m)})$.*

Proof: Suppose at time t the distribution of $\mathcal{S}$ is

$$\pi^*(\mathcal{S}) = \pi(x_t^{(1)}) \times \cdots \times \pi(x_t^{(m)}).$$

Without loss of generality, we assume that in the next step $x_t^{(1)}$ is chosen to be updated, and y_t is the anchor point which is obtained by a local optimization step started from $x_t^{(m)}$, say. Because the local optimization step induces a deterministic function of the anchor point, we can write $y_t = g(x_t^{(m)})$. Since $x_t^{(m)}$ follows distribution π, the anchor point y_t must follow a probability distribution h, which is determined by π and g. The theorem then follows from the following lemma which is a slightly more general version of Lemma 3.1 of Roberts and Gilks (1994). □

Lemma 3.1 *Suppose $x \sim \pi$ and y is any fixed point in a d-dimensional space. Let $\boldsymbol{e} = (x-y)/\|x-y\|$ be a unit vector. If r is drawn from distribution $f(r) \propto r^{d-1}\pi(y+r\boldsymbol{e})$, then $x' = y+r\boldsymbol{e}$ follows distribution π. If y is generated from a distribution $D(y)$ independent of x, then x' is independent of y and has density $\pi(x')$.*

Proof: Without loss of generality, we can let y be the origin. Then $\boldsymbol{e} = x/\|x\|$. If r is drawn from $f(r) \propto |r|^{d-1}\pi(rx/\|x\|)$, then for any measurable function $h(x)$ we see that

$$E\{h(x')\} = \int\int h(rx/\|x\|)\frac{|r|^{d-1}\pi(rx/\|x\|)}{\int |r'|^{d-1}\pi(r'x/\|x\|)dr'}\pi(x)drdx.$$

By letting $s = r/\|x\|$, we can rewrite the above equation as

$$E\{h(x')\} = \int\int h(sx)\frac{|s|^{d-1}\pi(sx)}{\int |s'|^{d-1}\pi(s'x)ds'}\pi(x)dsdx.$$

Let $g(x) = \int |s'|^{d-1}\pi(s'x)ds'$. Then $g(x)$ has the property that $g(tx) = |t|^{-d}g(x)$. Let $z = sx$, we obtain that

$$\begin{aligned} E\{h(x')\} &= \int\int h(z)\pi(z)|s|^{-1}\pi(s^{-1}z)/g(s^{-1}z)dsdz \\ &= \int h(z)\pi(z)/g(z)\int |s|^{-d-1}\pi(s^{-1}z)dsdz \\ &= \int h(z)\pi(z)dz = E_\pi\{h(x)\} \end{aligned}$$

The second to the last equality follows because $\int |s|^{-d-1}\pi(s^{-1}z)ds = \int |u|^{d-1}\pi(uz)du = g(z)$. Thus, the updated sample x' follows distribution π. Since the expectation $E\{h(x')\}$ does not depend on particular value of y, the independence between x' and y is apparent. □

4 Other Variations of the Multiple-Try Metropolis

In this section, we describe two other ways of using the MTM in Markov chain Monte Carlo sampling. The first method is closely related to the hit-and-run algorithm of Chen and

Schmeiser(1993), and the second one is related to the griddy Gibbs sampler (Ritter and Tanner, 1992).

4.1 Random-Ray Monte Carlo

Hit-and-Run (HR) 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: it allows the exploration of a wide range along a randomly chosen direction.

The HR tends to be helpful if there are separate *regions* with relatively high probabilities. 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 update of Metropolis (Chen and Schmeiser 1993). The MTM method provides a general way of using "directional" information as implied by $T(x, y)$ (note that all the proposals are made from T). Thus, we can prescribe a good $T(x, y)$ for a particular problem in order to let MTM exert its power. The following random-ray Monte Carlo scheme is a way of using MTM to achieve the hit-and-run effect. Suppose that the current state is $X_t = x^*$, our new algorithm is:

The Algorithm:

- Randomly generate a direction (a unit vector) $\boldsymbol{e}$.
- Propose to draw $y_1, \ldots, y_k$ from a distribution $T_e(x^*, y)$ along the direction $\boldsymbol{e}$. A generic choice is to draw iid samples $r_1, \ldots, r_k$ from $N(0, \sigma^2)$, where σ can be chosen rather big, and set $y_j = x + r_j\boldsymbol{e}$. Another possibility is to draw $r_j \sim \text{Unif}(-\sigma, \sigma)$.
- Conduct the multiple-try step. That is, we choose $Y = y^*$ from $y_1, \ldots y_k$ with probability proportional to $\pi(y_j)$; and then draw $x'_1, \ldots, x'_{k-1}$ iid from $T_e(y^*, x)$. Let $x^* = x'_k$. Then compute the generalized Metropolis ratio

$$r = \min\left\{1,\ \frac{\sum_{j=1}^{k} \pi(y_j) T_e(y_j, x^*)}{\sum_{j=1}^{k} \pi(x'_j) T_e(x'_j, y^*)}\right\}$$

In our experience, a much larger σ can be used compared to that in an HR with single Metropolis update, resulting in a higher acceptance rate for the same computational time.

4.2 Griddy-Gibbs-Like MTM

The Gibbs sampler differs from a typical Metropolis algorithm in its emphasis on the use of conditional distributions. When sampling from certain conditional distributions is not achievable analytically, Ritter and Tanner (1992) propose an effective alternative, the Griddy Gibbs sampler. The method has been applied successfully in several statistical modeling problems (Bernard, et. al. 1998), but its approximate nature still prevent it from being widely used. Using the MTM method, we can design an exact sampler that resembles the griddy Gibbs both in form and performance but is *exact* in the sense that its transition always has the target distribution π as the invariant distribution. The computation effort required by the new sampler is at most twice that of the original griddy Gibbs. Suppose $\boldsymbol{x} = (x_1, \ldots, x_d)$, and of interest is to draw from $\pi(\boldsymbol{x})$. Our algorithm is as follows:

- Pick any component, say x_j. Draw $t_1, \ldots, t_k$ iid from a *symmetric transition*, $T(x_j, t)$, and evaluate $w_l = \pi(x_j = t_l \mid \boldsymbol{x}_{[-j]})$, for $l = 1, 2, \ldots, k$. The transition T is *allowed* to depend on $\boldsymbol{x}_{[-j]}$. A simplest such choice is $T(x_j, t) \equiv c$, i.e., uniform, in the range of x_j, when the range is finite. Alternatively, we often choose $t = x_j + \epsilon$ with $\epsilon \sim N(0, \sigma^2)$.

- Choose $y{=}t_l$ with probability proportional to w_l. Draw $k-1$ iid samples from $T(y, \cdot)$, say $s_1, \cdots, s_{k-1}$. Name $s_k = x_j$.

- Compute the generalized Metropolis ratio
$$r = \min\left\{1,\ \frac{\sum_{l=1}^{k} \pi(t_l, \boldsymbol{x}_{[-j]})}{\sum_{l=1}^{k} \pi(s_l, \boldsymbol{x}_{[-j]})}\right\}$$
Accept y with probability r and reject with $1-r$.

5 Numerical Examples

5.1 A multimodal problem

Consider simulating from a 2-dimensional mixture Gaussian distribution $\pi(x)$

$$0.34 \times N_2(\mathbf{0}, I_2) + 0.33 \times N_2\left\{\begin{pmatrix} -6 \\ -6 \end{pmatrix}, \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}\right\} + 0.33 \times N_2\left\{\begin{pmatrix} 4 \\ 4 \end{pmatrix}, \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}\right\}$$

Here the covariance matrices in the three mixture components are identical to those in Gilks et al. (1998), but the mean vectors are separated by a larger distance in each dimension (twice as large).

We started two independent Metropolis sampler with starting points drawn from unif $[-.5,.5]^2$. A spherical proposal function was used: it uniformly selects a direction and then draw the radius from Uniform $[0,a]$, where a(=4 in our case) is calibrated so that the Metropolis sampler had an acceptance rate of about 0.23 (Gelman et al., 1993). A total of 200,000 iterations of the Metropolis step was conducted for each sampler, which took about 28 seconds of CPU time from a Sun Ultra 2 workstation. In Figure 1, we plotted the histograms and autocorrelations for one of the variable (left panels). It is seen that the Metropolis sampler moves very slowly due to the low-probability barriers between the modes, and the mixture proportions are very poorly estimated.

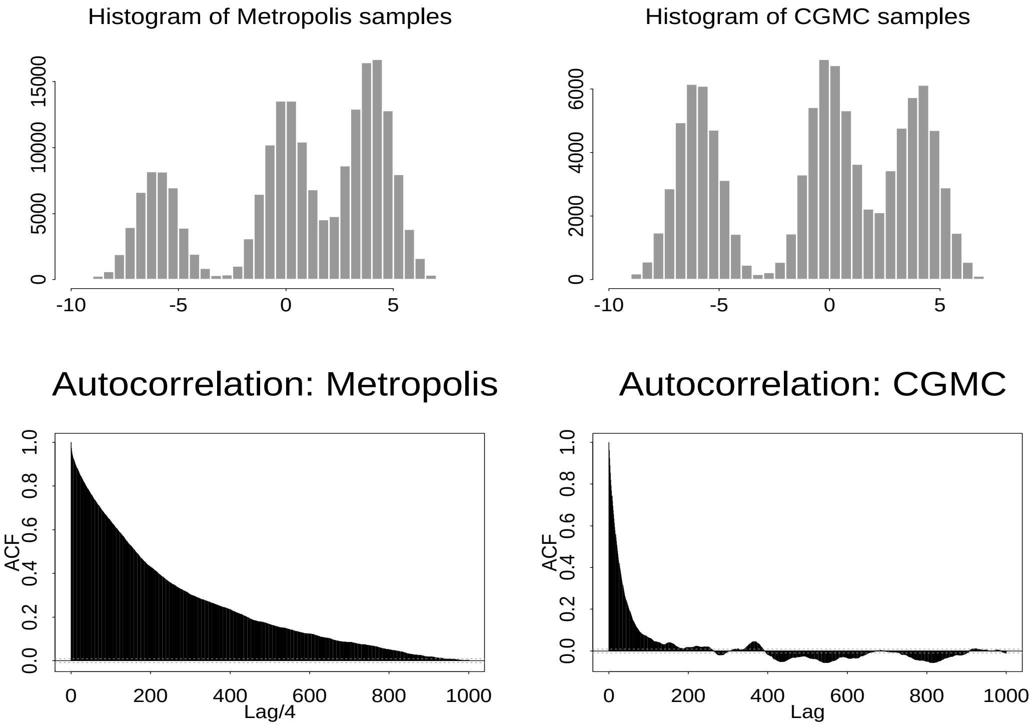

Figure 1: A comparison of the results obtained by the Metropolis sampler and that by the CGMC. The autocorrelation plot of the Metropolis samples has taken the computational cost into account.

The CGMC method is applied to this problem with m=2 streams and 20,000 iterations for each. Each iteration consists of 2 Metropolis steps and one adaptation step. So a total of 100,000 random draws from π was produced as the program ended, which took about 27 seconds of CPU time from the same computer. The proposal function for the Metropolis step was the same spherical distribution as in the previous case but with a narrower range for the radius: $[0, 2.5]$ (corresponding to an acceptance rate of 0.37). For the GCMC, a small Metropolis step is beneficial for the purpose of exploring local features. The line sampling proposal was a univariate Gaussian with variance $= 10^2$ and the number of tries $k = 5$. This corresponds to an acceptance rate of .47. Our experience shows that an acceptance rate between 0.4 and 0.5 for the multiple-try step is appropriate. In Figure 1, we plotted the histograms and autocorrelations for one of the variable in one streams (right panels).

Using the heuristic of *integrated autocorrelation time (IAT)*, which equals to the sum of all-lag autocorrelations, we can estimate that with the *same* amount of CPU time, the IAT for the Metropolis algorithm is about 249 after adjusting for the computational cost (4 to 1 ratio), whereas for each stream of the CGMC the integrated autocorrelation time is about 34. This translates to a 7-fold improvement.

To push the limit, we also tested the CGMC on a 5-dimensional mixture Gaussian

$$\pi(\boldsymbol{x}) = \frac{1}{3}N_5(\mathbf{0}, I_5) + \frac{2}{3}N_5(\mathbf{5}, I_5),$$

where $\mathbf{0} = (0, \ldots, 0)$ and $\mathbf{5} = (5, \ldots, 5)$. Thus the distance between the two modes is $5\sqrt{5} = 11.2$, which posts a great challenge to Metropolis samplers.

We started the CGMC with $m = 2$ steams and the initial values drawn from Unif$[-.5, .5]^5$ (e.g., both streams were started from the first mode). Similar to the previous example, each iteration of the CGMC algorithm consisted of two Metropolis steps and one gradient line-sampling step. The line-sampling step uses a Gaussian proposal with std= 20, and $k = 10$ multiple tries (the resulting acceptance rate was about 0.44). The Metropolis step used a spherical distribution which is uniform in the polar coordinates with radius $\in (0, 1.5)$ (acceptance rate=0.36). We tested a large number of different proposal step sizes, ranging from 5 to 25. The results is insensitive to the choice of step-size in this range. There are two useful guidelines: (a) the step size for the line-sampling should be reasonably large, i.e., with a resulting acceptance rate in the range of (.35, .55), which gives one a huge range of proposal step size; (b) the proposal step size for the Metropolis should give an acceptance rate in the range of (0.35,0.5).

With 100,000 iterations, the CGMC algorithm produced a total of 300,000 random draws from π in about 300 seconds of CPU time. The estimates of mixing proportion, marginal means, variances, and even cdfs based on these 300,000 samples are rather accurate (i.e., differ from the true values in the second decimal place). In Figure 2, we plotted the histogram and the time

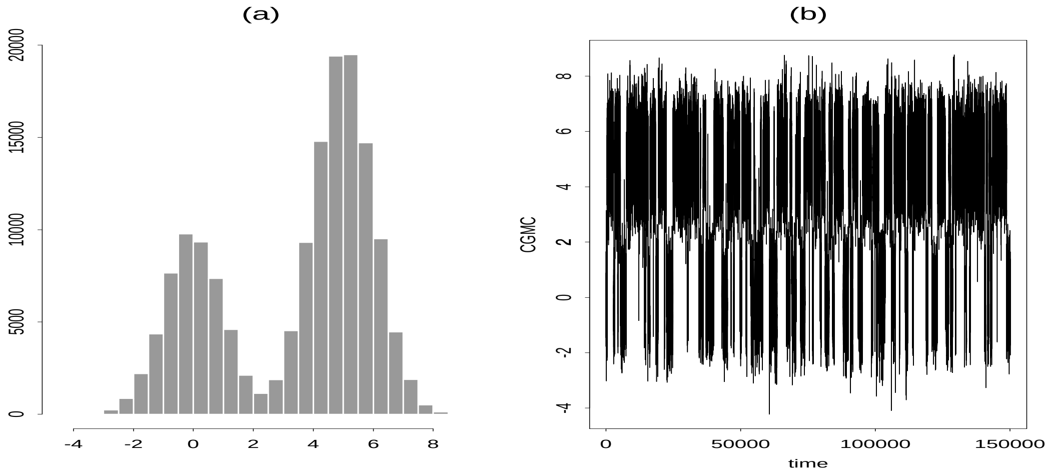

Figure 2: The mixing result of the CGMC samples: (a)a marginal histogram of the sample; (b) time series plot of one stream of the CGMC.

series of the first coordinate of $\boldsymbol{x}$ in one of the two streams (the size was 150,1347). The IAT for this time series is in the range between 400 to 550, which leaves us an effective sample size of about 600 with two streams.

A Random-Ray sampler (Section 4.1) was also applied to the problem with $k = 8$ and the proposal std=12. It performed rather well for this problem. With the same CPU time, it can produce about 300 effective sample, slightly worse than the CGMC. This similarity in performance between the two methods is understandable: because the two modes are essentially unconnected, the gradient information in this example provides little help for the CGMC. The reason for both methods to work is the effectiveness of line-sampling.

For comparisons, we also tested with an optimally tuned Metropolis algorithm (the proposal distribution is spherical Gaussian with variance $= 1.5^2$, corresponding to an acceptance rate of 0.22). With similar computing time (300 seconds), we can generate 7,000,000 draws. But in all of 10 runs we tried, the sampler was never able to get out of the mode it started with.

5.2 Damped Sinusoidal Fitting

In this experiment, we simulated $n = 200$ observations from the model

$$y_i = \sum_{j=1}^{J} e^{-a_j - b_j x_i} cos(c_j x_i + d_j) + \epsilon_i,$$

where $\epsilon_i \sim N(0, \sigma^2)$. The true signal in this model has three sinusoidal components ($J = 3$) where each component is characterized by its weights e^{-a_j}, damping constant b_j, angular frequency c_j, and phase d_j. In our simulation, the weight factor $a_j = 0$, $j = 1, \ldots, 3$, is assumed known. Thus, there are a total of 10 unknown parameters and they can be summarized as $\boldsymbol{\theta} = (\sigma;\ b_j, c_j, d_j,\ j = 1, \ldots, 3)$, with their true values being (.3, 0, .9, 1.57, .2, 1, 0, .1, 1.5, -1.57). Note that the three different frequencies (c_j) are 0.9, 1.0 and 1.5. The likelihood function of $\boldsymbol{\theta}$ can be written as

$$L(\boldsymbol{\theta} \mid y_1, \ldots, y_n) \propto \frac{1}{\sigma^n} \exp\left\{ -\frac{\sum_{i=1}^{n} (y_i - \sum_{j=1}^{3} e^{-a_j - b_j x_i} cos(c_j x_i + d_j))^2}{2\sigma^2} \right\}, \tag{5}$$

which can also be treated as a posterior distribution of $\boldsymbol{\theta}$ with a flat prior.

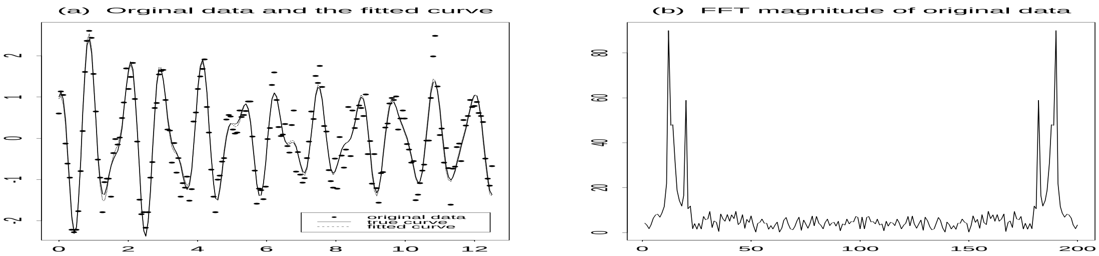

Figure 3: (a) The simulated sinusoidal data and the fitted curve. (b) The Fourier transform of the data.

A fast Fourier transform was first applied to the data and two major frequencies were easily recognized. But the third frequency component was not detectable. Also, Fourier analysis has difficulties in resolving the damping coefficients (Figure 3 (b)). These suggest that, although it is computationally more demanding, a brute-force curve fitting method can sometimes help gain more information on the finer structure of a sinusoidal signal than a fast Fourier analysis.

We applied both the Metropolis and the CGMC method to find the maximum likelihood fit of the data. The CGMC with m=5 streams was run for 14.5 minutes, which produced 1200 iterations (each iteration consists of one step of line sampling and 20 steps of Metropolis). The Metropolis was run for the same amount of time (5 independent streams, each with 2000 × 20 Metropolis steps in each iteration). We monitored the change of likelihood as the iterations proceeded. The Metropolis found the modal region (the best out of 5 streams) 6 times out of 20 repeated runs (about 30%). In contrast, the CGMC found the modal region 14 times in 20 trials (about 70%). Interestingly, the deterministic conjugate gradient method, which is routinely used

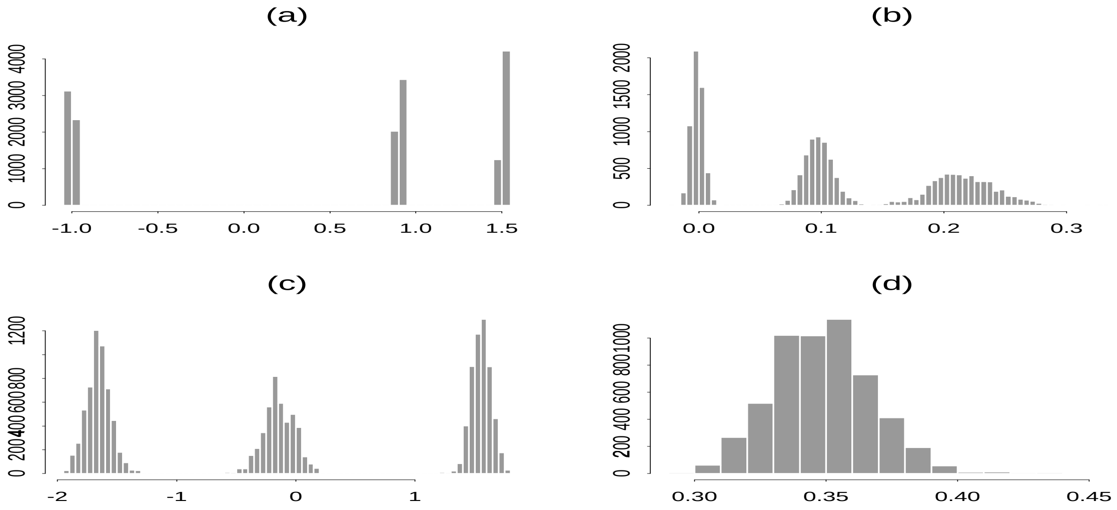

Figure 4: Using Monte Carlo samples produced by the CGMC, we obtained histograms of (a) three frequencies (true values c_j= .9, 1, and 1.5); (b) three damping coefficients (true values b_j=0, .1, and .2); (c) three phases (true values d_j=-1.57, 0, and 1.57); (d) the error standard deviation (true value σ=.3).

for nonlinear least square fitting, failed in this problem. The best fit of the data is shown by dotted lines in Figure 3(a). Posteriors of the unknown quantities, based on histograms obtained from one CGMC run with m =5 chains, are shown in Figure 4.

5.3 Fitting Mixture Models

Suppose $y_1, \ldots, y_n$ are iid samples from a mixture distribution with 3 Gaussian components with unknown means, variances, and proportions. It is easy to write down the likelihood as

$$L(\boldsymbol{\theta} \mid y_1, \ldots, y_n) = \prod_{i=1}^{n} \left\{ p_1 \phi\left(\frac{y_i - \mu_1}{\sigma_1}\right) + p_2 \phi\left(\frac{y_i - \mu_2}{\sigma_2}\right) + p_3 \phi\left(\frac{y_i - \mu_3}{\sigma_3}\right) \right\}$$

where $p_3 = 1 - p_1 - p_2$ and $\phi(\cdot)$ is the standard Gaussian density. With a Dirichlet (0,0,0) as the prior for (p_1, p_2, p_3) and flat prior on $(\mu_i, \log\sigma_i, i = 1, 2, 3)$, and a restriction $\sigma_{[\text{min}]} \leq \sigma_i \leq \sigma_{[\text{max}]}$ and $p_i > \epsilon$, we can easily get the expression for the posterior distribution of these parameters. To cope with the unidentifiability problem, we impose the restriction that $\mu_1 \geq \mu_2 \geq \mu_3$. We also did a parameter transformation $u_1 = \log p_1$ and $u_2 = \log p_2$, and let our sampler operate on $\boldsymbol{\theta} = (u_1, u_2; \mu_i, \log\sigma_i, i = 1, 2, 3)$.

We simulated a data set with $n = 200$; p_1=.2, p_2=.3; μ_1=−5, μ_2=0, μ_3=5; and σ_1=2, σ_2=1, σ_3=2. The histogram of this dataset is shown in Figure 7(a). A Gibbs sampler can be designed for this problem with the introduction of a set of latent component-indicator variables (Diebolt and Robert 1994; Chen and Liu 1996). Instead of the latent variable approach, we can also attempt to simulate from the joint posterior distribution of $\boldsymbol{\theta}$ directly by a Metropolis sampler.

Clearly, efficiency of a Metropolis sampler running on the space of $\boldsymbol{\theta}$ strongly depends on its proposal distribution. In the test example, we have chosen a generic proposal: randomly choose a direction in the $\boldsymbol{\theta}$ space, and then sample a distance from uniform [0, a], where a is adjusted so that the overall acceptance rate is about .25. For our example, a=.3 was suitable. This choice of the proposal is apparently unfavorable for the simulation because it puts all the parameters in a common scale (e.g., it treats $\log p_i$, μ_j, and $\log \sigma_k$ indiscriminately). Although we understand that a more sophisticated choice of the proposal distribution can increase efficiency of the sampler, we believe that it would be a more convincing illustration to use a generic proposal without entertaining any special property of the problem. The CGMC will also use a similar proposal — by which we hope to show that a brute force approach is also a viable choice.

We independently started 100 Metropolis chains with random starting points and monitored their log-likelihood values. With 21,000 Metropolis steps, we observed that 33 chains out of the 100 were stuck in local modes, whereas the rest of them successfully settled down in the modal region (i.e., log-likelihood ≥ -555).

We applied the CGMC to the same problem, with one-step conjugate-gradient local optimization, a radius of 8 for the line-sampling, and k=20 multiple tries. The total number of streams was kept at m=4. Between every line-sampling, 20 Metropolis steps are inserted. Thus, "one-iteration" in this algorithm consists of a line-sampling and 20 Metropolis steps. The computational cost of one step of line-sampling is roughly the same as 10 steps of the standard Metropolis moves. We did 10 independent runs of the CGMC on this problem, which produced a total of 40 chains. We found that all but 3 chains settled in the modal region within 500 iterations (equivalent of 7,500 Metropolis steps). Figure 5 presents the time series plot of the log-likelihood for the 4 streams of a randomly chosen CGMC run.

A by-product of the CGMC sampler is that a convergence-diagnostic statistic can be produced based on the multiple streams of a single CGMC run. In this example, the time series plot (Figure 5(d)) together with the Gelman-Rubin statistic $\hat{R}$ (Figure 5(c)) served convergence diagnosis purpose very well: an $\hat{R} < 1.2$ computed with m=4 chains always indicated that the stationarity was reached. But some caution needs to be taken here: since the multiple streams used in the CGMC are not completely independent, the resulting $\hat{R}$ is not as reliable as that from independent runs. A quick remedy is to run 2 independent CGMC runs to compute the G-R statistic.

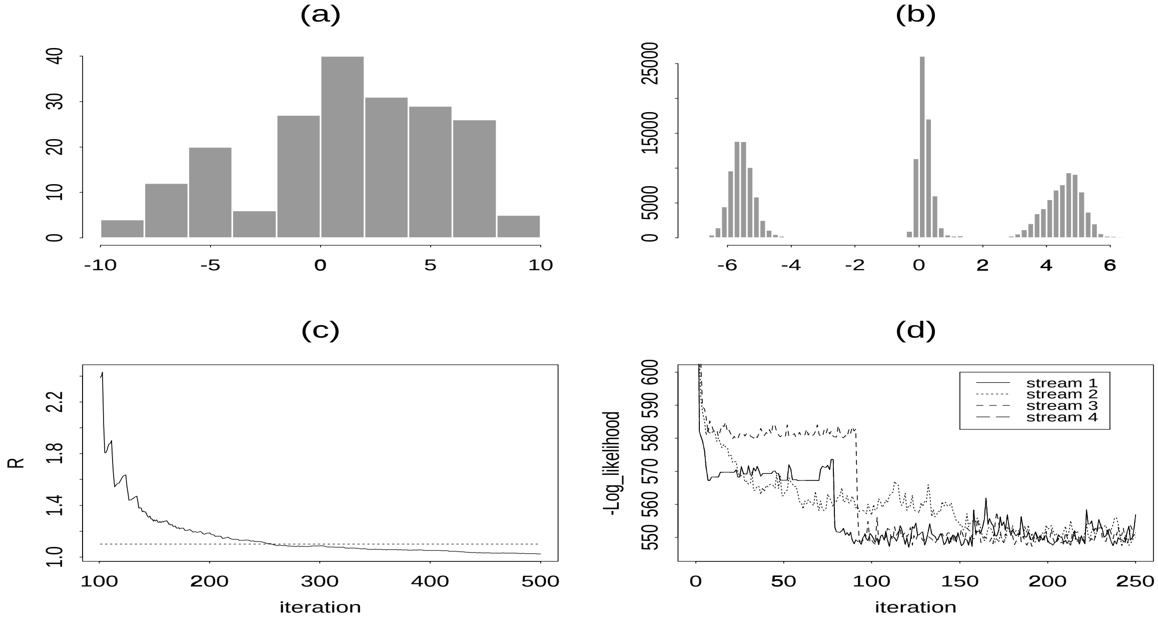

Figure 5: (a) Histogram of the data; (b) the posterior distribution of the μ_i, with the samples obtained from the CGMC; (c) Gelman-Rubin convergence criterion $\hat{R}$; (d) time series plot of the 4 streams in the CGMC. Every iteration unit in both (c) and (d) corresponds to 1 step of line-sampling and 20 steps of Metropolis in simulation.

6 Discussion

We have proposed two novel ideas in this article: a new transition rule applicable to all Markov chain Monte Carlo algorithms, and a way to utilize local optimization in Monte Carlo simulation. Necessary theoretical foundations are provided to justify the correctness of the methods. The usefulness of the two ideas have been carefully examined through several numerical examples, and some other variations mentioned.

Generally speaking, the multiple-try Metropolis enables us to make large step-size transitions in a MCMC sampler. 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.

It has long been proposed by researchers that Monte Carlo efficiency can be improved by first doing mode-finding and then adjusting the proposal function accordingly (Gelman and Rubin 1992). But to our best knowledge, there is no effective and general-purpose means to put mode-finding steps into a proper Markov chain Monte Carlo framework. Lemma 3.1 shows that any *anchor point* that is independent of the current state can be effectively used in a MCMC sampler to direct future draws. Thus, one can either apply deterministic local-finders in advance to locate some modes as anchor points or apply these deterministic procedures adaptively as in our CGMC sampler. Further analysis along this direction is of interest.

REFERENCES

Barnard, J., McCulloch, R., and Meng, X.L. (1997). Modeling covariance matrices in terms of standard deviations and correlations with application to shrinkage. *Technical Report*, Department of Statistics, Harvard University.

Chen, R. and Liu, J.S. (1996). Predictive updating methods with applications in Bayesian classification. *J. R. Statist. Soc.* Ser. B, **58**, 397-415.

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.

Diebolt, J. and Robert, C.P. (1994) Estimation of finite mixture distribution through Bayesian sampling. *J. Roy. Statist. Soc.* Ser. B, **56**, 363-375.

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

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

Gelman, A., Roberts, R.O., and Gilks, W.R. (1996). Efficient Metropolis jumping rules. In *Bayesian Statistics* 5, J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith (eds). New York: Oxford University Press.

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

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

Gilks, W.R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*,

Gilks, W.R., Roberts, R.O., and Sahu, S.K. (1998). Adaptive Markov chain Monte Carlo through regeneration. *Technical Report.*

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

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (1996). *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge: University Press.

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

Roberts, G.O. and Gilks, W.R. (1994). Convergence of Adaptive Direction Sampling. *Journal of Multivariate Analysis* **49**, 287-298.