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. By using this transition rule together with adaptive direction sampling (Gilks, Roberts, and George 1994), we propose a novel method for incorporating local optimization steps into a MCMC sampler in continuous state-space. Numerical studies show that the new method performs significantly better than the traditional Metropolis-Hastings (M-H) sampler. With minor tailoring in using the rule, the multiple-try method can also be exploited to achieve the effect of a griddy 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; Conjugate Gradient; Damped Sinusoidal; Gibbs Sampling; Griddy Gibbs Sampler; Hit-and-Run Algorithm; Markov Chain Monte Carlo; Metropolis Algorithm; Mixture Model; Orientational Bias Monte Carlo.

1 Introduction

Monte Carlo simulation methods have been increasingly recognized by scientists as indispensable tools for difficult computational problems. These methods have also been central to many recent developments in statistical modeling, computation, and general practice. A common feature of methods for simulating from a complex probability distribution, say $\pi(\boldsymbol{x})$, is their reliance on cumulative evolutions of small, albeit random, changes. A fundamental construction that enables such "local-search" methods is the Markov transition function proposed by Metropolis et al. (1953) and modified by Hastings (1970). Briefly, suppose that we currently have obtained $\boldsymbol{x}_t$, Metropolis-Hastings suggest that one can employ a rather arbitrary Markov transition function T to generate a proposal $\boldsymbol{y} \sim T(\boldsymbol{x}_t, \cdot)$, and then "thin it down" by letting

[1] Jun S. Liu is Assistant Professor, Department of Statistics, Stanford University, Stanford, CA 94305. Faming Liang is Visiting Fellow, Department of Statistics and Applied Probability, the National University of Singapore, Singapore 119260. Wing Hung Wong is Professor, Department of Statistics, University of California, Los Angeles, CA 90095. Liu's research is supported in part by NSF grant DMS-9803649. Wong's research is supported in part by NSF grant DMS-9703918. We thank James Berger, Persi Diaconis, Radford Neal, two referees and the associate editor for helpful suggestions.

the next state $\boldsymbol{x}_{t+1} = \boldsymbol{y}$ with probability

$$r = \min\left\{1,\ \frac{\pi(\boldsymbol{y})T(\boldsymbol{y},\boldsymbol{x}_t)}{\pi(\boldsymbol{x}_t)T(\boldsymbol{x}_t,\boldsymbol{y})}\right\}, \tag{1}$$

and letting $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$ with the remaining probability. We call r the M-H ratio.

The Metropolis-type local move allows one to break a complex task into a series of manageable pieces. On the other hand, however, these local moves often lead to slow-converging algorithms which are easily trapped in a local mode. A brute-force way of being "less local" is to directly increase the "searching region" of each Metropolis-Hastings step characterized by $T(\boldsymbol{x},\boldsymbol{y})$. But this approach often results in a very small M-H ratio and, therefore, an ineffective algorithm. The multiple-try Metropolis (MTM) algorithm described in this article tackles this "conflict of interest" by enabling one to propose multiple trial points from T and, therefore, allowing one to use those $T(\boldsymbol{x},\boldsymbol{y})$ corresponding to a very large searching region.

As an alternative to the basic Metropolis-Hastings construction, conditional sampling in some desirable subspaces or along certain directions as that in a Gibbs sampler (Geman and Geman 1984), in a multigrid Monte Carlo algorithm (Goodman and Sokal 1989), or in a hit-and-run algorithm (Chen and Schmeiser 1993) is often a powerful updating step in Markov chain Monte Carlo (MCMC). However, an inherent difficulty with this approach is that the required 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 as in the griddy Gibbs sampler of Ritter and Tanner (1992). But when approximations are used, there will no longer be any guarantees of the existence of an equilibrium distribution, let alone an assessment of discrepancy between the Monte Carlo output and the target distribution. A remedy is to incorporate the Metropolis-Hastings rule in using the approximate distribution. But a good approximation is usually too expensive to come by and such approximations get worse very fast as dimension increases. The MTM method provides an effective means to conduct directional sampling without having to suffer the shortcomings in those previous approaches.

Another actively pursued direction for improving a MCMC sampler is via adaptation; that is, roughly speaking, when one attempts to use information generated by the up-to-date iterations to guide for future simulations. A particularly interesting idea is the *adaptive direction sampling* (ADS) algorithm proposed by Gilks, et al. (1994), which suggests using multiple MCMC chains and adapting movements of one chain according to information from others. We describe a novel algorithm that combines the ADS framework with the MTM method. The main features of the new algorithm are (i) the local optimality information revealed by a deterministic local-search scheme is explicitly used for adaptation; and (ii) the companion MTM allows for a very large

step-size in searching along a "promising" direction. Our numerical examples show that the new sampler offers significant improvement over the traditional Metropolis sampler, especially in difficult problems.

This paper is arranged as follows. Section 2 introduces the basic MTM methodology and provides a simple proof for the correctness of the method; Section 3 discusses further extensions of the MTM and a few theoretical issues; Section 4 presents our new sampler, the *conjugate-gradient Monte Carlo*; Section 5 shows two other variations in using the MTM; Section 6 demonstrates the use of the method through several numerical examples; and Section 7 concludes with a brief discussion.

2 General Methodology of Multiple-Try Metropolis

As in a standard Metropolis-Hastings algorithm, we let $T(\boldsymbol{x}, \boldsymbol{y})$ be a proposal transition function which may or may not be symmetric. A modest requirement is that $T(\boldsymbol{x}, \boldsymbol{y}) > 0$ if and only if $T(\boldsymbol{y}, \boldsymbol{x}) > 0$. Furthermore, we define

$$w(\boldsymbol{x}, \boldsymbol{y}) = \pi(\boldsymbol{x}) T(\boldsymbol{x}, \boldsymbol{y}) \lambda(\boldsymbol{x}, \boldsymbol{y}), \tag{2}$$

where $\lambda(\boldsymbol{x}, \boldsymbol{y})$ is a nonnegative symmetric function in $\boldsymbol{x}$ and $\boldsymbol{y}$ that can be chosen by the user. The only requirement is that $\lambda(\boldsymbol{x}, \boldsymbol{y}) > 0$ whenever $T(\boldsymbol{x}, \boldsymbol{y}) > 0$. We present a few choices of $\lambda(\boldsymbol{x}, \boldsymbol{y})$ in the latter part of this section. Suppose the *current state* is $\boldsymbol{X}_t = \boldsymbol{x}$, then a MTM transition is defined as follows:

Multiple-Try Metropolis:

- Draw k i.i.d. trial proposals, $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$, from $T(\boldsymbol{x}, \cdot)$. Compute $w(\boldsymbol{y}_j, \boldsymbol{x})$ for $j = 1, \ldots, k$.
- Select $\mathbf{Y} = \boldsymbol{y}$ among the trial set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$ with probability proportional to $w(\boldsymbol{y}_j, \boldsymbol{x})$, $j = 1, \ldots, k$. Then draw $\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{k-1}^*$ from the distribution $T(\boldsymbol{y}, \cdot)$, and let $\boldsymbol{x}_k^* = \boldsymbol{x}$.
- Accept $\boldsymbol{y}$ with probability

$$r_g = \min\left\{1, \frac{w(\boldsymbol{y}_1, \boldsymbol{x}) + \cdots + w(\boldsymbol{y}_k, \boldsymbol{x})}{w(\boldsymbol{x}_1^*, \boldsymbol{y}) + \cdots + w(\boldsymbol{x}_k^*, \boldsymbol{y})}\right\} \tag{3}$$

  and reject it with probability $1 - r_g$. The quantity r_g is called the *generalized M-H ratio.*

Theorem 2.1 *The MTM transition rule described above satisfies the detailed balance condition and, hence, induces a reversible Markov chain with π as its invariant distribution.*

Proof: Let $A(\boldsymbol{x}, \boldsymbol{y})$ be the actual transition probability for moving from $\boldsymbol{x}$ to $\boldsymbol{y}$ in a MTM sampler. Suppose $\boldsymbol{x} \neq \boldsymbol{y}$ and let I indicate which of $\boldsymbol{y}_j$ has been selected. Since $w(\boldsymbol{y}, \boldsymbol{x}) =$

$\pi(\boldsymbol{y})T(\boldsymbol{y},\boldsymbol{x})\lambda(\boldsymbol{y},\boldsymbol{x})$ and the $\boldsymbol{y}_j$ are exchangeable, we have

$$\begin{aligned}
\pi(\boldsymbol{x})A(\boldsymbol{x},\boldsymbol{y}) &= \pi(\boldsymbol{x})P[\cup_{j=1}^{k}\{(\mathbf{Y}_j=\boldsymbol{y})\cap(I=j)\}\mid\boldsymbol{x}] \\
&= k\;\pi(\boldsymbol{x})P[(\mathbf{Y}_k=\boldsymbol{y})\cap(I=k)\mid\boldsymbol{x}] \\
&= k\;\pi(\boldsymbol{x})\int\cdots\int T(\boldsymbol{x},\boldsymbol{y})T(\boldsymbol{x},\boldsymbol{y}_1)\cdots T(\boldsymbol{x},\boldsymbol{y}_{k-1})\frac{w(\boldsymbol{y},\boldsymbol{x})}{w(\boldsymbol{y},\boldsymbol{x})+\sum_{j=1}^{k-1}w(\boldsymbol{y}_j,\boldsymbol{x})} \\
&\times \min\left\{1,\frac{w(\boldsymbol{y},\boldsymbol{x})+\sum_{j=1}^{k-1}w(\boldsymbol{y}_j,\boldsymbol{x})}{w(\boldsymbol{x},\boldsymbol{y})+\sum_{j=1}^{k-1}w(\boldsymbol{x}_j^*,\boldsymbol{y})}\right\}T(\boldsymbol{y},\boldsymbol{x}_1^*)\cdots T(\boldsymbol{y},\boldsymbol{x}_{k-1}^*)d\boldsymbol{y}_1\cdots d\boldsymbol{y}_{k-1}d\boldsymbol{x}_1^*\cdots d\boldsymbol{x}_{k-1}^* \\
&= k\;\frac{w(\boldsymbol{x},\boldsymbol{y})w(\boldsymbol{y},\boldsymbol{x})}{\lambda(\boldsymbol{y},\boldsymbol{x})}\int\cdots\int\min\left\{\frac{1}{w(\boldsymbol{y},\boldsymbol{x})+\sum_j w(\boldsymbol{y}_j,\boldsymbol{x})},\frac{1}{w(\boldsymbol{x},\boldsymbol{y})+\sum_j w(\boldsymbol{x}_j^*,\boldsymbol{y})}\right\} \\
&\times T(\boldsymbol{x},\boldsymbol{y}_1)\cdots T(\boldsymbol{x},\boldsymbol{y}_{k-1})T(\boldsymbol{y},\boldsymbol{x}_1^*)\cdots T(\boldsymbol{y},\boldsymbol{x}_{k-1}^*)d\boldsymbol{y}_1\ldots d\boldsymbol{y}_{k-1}d\boldsymbol{x}_1^*\cdots d\boldsymbol{x}_{k-1}^* \qquad (4)
\end{aligned}$$

The expression (4) is symmetric in $\boldsymbol{x}$ and $\boldsymbol{y}$ because $\lambda(\boldsymbol{x},\boldsymbol{y})=\lambda(\boldsymbol{y},\boldsymbol{x})$. Thus we proved that $\pi(\boldsymbol{x})A(\boldsymbol{x},\boldsymbol{y})=\pi(\boldsymbol{y})A(\boldsymbol{y},\boldsymbol{x})$, which is the detailed balance condition. □

The simplest choice of the symmetric function in the MTM sampler is $\lambda(\boldsymbol{x},\boldsymbol{y})\equiv 1$, and the resulting algorithm will be called MTM (I). Another choice is

$$\lambda(\boldsymbol{x},\boldsymbol{y})=\left(\frac{T(\boldsymbol{x},\boldsymbol{y})+T(\boldsymbol{y},\boldsymbol{x})}{2}\right)^{-1}. \qquad (5)$$

When $T(\boldsymbol{x},\boldsymbol{y})$ is symmetric, (5) leads to the following MTM (II) algorithm:

- Draw k trials $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k$ from a *symmetric* proposal distribution $T(\boldsymbol{x},\cdot)$.
- Select $\mathbf{Y}=\boldsymbol{y}$ among $\{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k\}$ with probability proportional to $\pi(\boldsymbol{y}_j)$, $j=1,\ldots,k$. Then draw $\boldsymbol{x}_1^*,\ldots,\boldsymbol{x}_{k-1}^*$ from the distribution $T(\boldsymbol{y},\boldsymbol{x}^*)$. Denote $\boldsymbol{x}_k^*=\boldsymbol{x}$.
- Accept $\boldsymbol{y}$ with probability

$$\min\left\{1,\frac{\pi(\boldsymbol{y}_1)+\cdots+\pi(\boldsymbol{y}_k)}{\pi(\boldsymbol{x}_1^*)+\cdots+\pi(\boldsymbol{x}_k^*)}\right\}$$

  and reject with the remaining probability.

MTM (II) corresponds to the method of "orientational-biased Monte Carlo" described in Frenkel and Smit (1996, Section 13.1), in which a specialized proof in the context of molecular structure simulation is also provided. The idea of utilizing multiple trial proposals to "bias" the local sampling in a Metropolis algorithm seems to first appear in Mezei (1980), whose method is in fact very similar to a local griddy-Gibbs. Cracknell, Nicholson, Parsonage, and Evans (1990) describe a more general version of Mezei's method and apply it to the simulation of fluids (e.g.,

water, benzene) and other structured particles. However, Mezei's method differs substantially from MTM (II) because of its approximate nature.

Another sensible choice of λ is

$$\lambda(\boldsymbol{x}, \boldsymbol{y}) = \{T(\boldsymbol{x}, \boldsymbol{y})T(\boldsymbol{y}, \boldsymbol{x})\}^{-\alpha}, \tag{6}$$

defined in the region $\{(\boldsymbol{x}, \boldsymbol{y}) : \ T(\boldsymbol{x}, \boldsymbol{y}) > 0\}$. When $\alpha = 1$, function $w(\boldsymbol{x}, \boldsymbol{y})$ corresponds to the importance weight of $\boldsymbol{x}$ as though the sampling distribution is $T(\boldsymbol{y}, \boldsymbol{x})$ and the target is $\pi(\boldsymbol{x})$. This seems to be an appealing feature. From our limited numerical experiences, we found that the performance of an MTM sampler is insensitive to the choice of $\lambda(\boldsymbol{x}, \boldsymbol{y})$, although the form (6) with α close to 1 worked the best in a few cases and MTM (II) was robust in all of our examples. It is still an open question whether and how we can find an optimal $\lambda(\boldsymbol{x}, \boldsymbol{y})$ for a particular problem.

The MTM transition allows one to explore more thoroughly in the "neighboring region" defined by $T(\boldsymbol{x}, \boldsymbol{y})$; this is especially useful when such a region (or direction) is obtained from a relatively expensive adaptation method.

3 Some Theoretical Discussions on the MTM

3.1 Further Generalizations

From the proof of Theorem 2.1, we observe that the k trial proposals $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$ need not be independent draws from $T(\boldsymbol{x}, \cdot)$. For example, if the transition function depends on another random variable $\boldsymbol{e}$ whose sampling distribution is $f_{\boldsymbol{x}}(\boldsymbol{e})$, i.e., $T(\boldsymbol{x}, \boldsymbol{y}) = \int T_{\boldsymbol{e}}(\boldsymbol{x}, \boldsymbol{y}) f_{\boldsymbol{x}}(\boldsymbol{e}) d\boldsymbol{e}$, we can first draw $\boldsymbol{e}$ and then draw the trial set $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$ from $T_{\boldsymbol{e}}(\boldsymbol{x}, \cdot)$. In this case, the $\boldsymbol{y}$'s are no longer independently distributed, but are still exchangeable. The MTM rule can be modified accordingly. More precisely, by defining that

$$w_{\boldsymbol{e}}(\boldsymbol{x}, \boldsymbol{y}) = \pi(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{e}) T_{\boldsymbol{e}}(\boldsymbol{x}, \boldsymbol{y}) \lambda_{\boldsymbol{e}}(\boldsymbol{x}, \boldsymbol{y}), \tag{7}$$

where, $\lambda_{\boldsymbol{e}}(\boldsymbol{x}, \boldsymbol{y})$ is a positive symmetric function in $\boldsymbol{x}$ and $\boldsymbol{y}$, we can implement a modified MTM as follows: (a) choose $\mathbf{Y} = \boldsymbol{y}$ among $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$ with probability proportional to $w_{\boldsymbol{e}}(\boldsymbol{x}, \boldsymbol{y}_j)$; (b) generate $\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{k-1}^*$ from $T_{\boldsymbol{e}}(\boldsymbol{y}, \cdot)$; and (c) accept $\boldsymbol{y}$ with the generalized Metropolis ratio

$$\min\left\{1, \frac{w_{\boldsymbol{e}}(\boldsymbol{y}, \boldsymbol{x}) + \sum_{j=1}^{k-1} w_{\boldsymbol{e}}(\boldsymbol{y}_j, \boldsymbol{x})}{w_{\boldsymbol{e}}(\boldsymbol{x}, \boldsymbol{y}) + \sum_{j=1}^{k-1} w_{\boldsymbol{e}}(\boldsymbol{x}_j^*, \boldsymbol{y})}\right\}. \tag{8}$$

One can easily show that this scheme satisfies the detailed balance. When the sampling distribution of $\boldsymbol{e}$ is independent of the location $\boldsymbol{x}$ or $\boldsymbol{y}$, the generalized Metropolis ratio (8) can

be further simplified by ignoring $f_{\boldsymbol{x}}(\boldsymbol{e})$ in the definition of $w_{\boldsymbol{e}}(\boldsymbol{x},\boldsymbol{y})$, i.e., by using $w_{\boldsymbol{e}}(\boldsymbol{x},\boldsymbol{y}) = \pi(\boldsymbol{x})T_{\boldsymbol{e}}(\boldsymbol{x},\boldsymbol{y})\lambda_{\boldsymbol{e}}(\boldsymbol{x},\boldsymbol{y})$.

Some other generalizations of the basic MTM algorithm are also possible. For example, the trial set $\{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k\}$ can be generated from a Markov chain (or even deterministically) in which case the balance set $\{\boldsymbol{x}_1^*,\ldots,\boldsymbol{x}_{k-1}^*\}$ has to be generated differently. The effectiveness of these algorithms is currently under investigation.

3.2 Limiting Behavior of the MTM

It is of interest to know the limiting behavior of the MTM sampler as the number of trial proposals k becomes large. Let

$$\rho(\boldsymbol{x},\boldsymbol{y}) = T(\boldsymbol{x},\boldsymbol{y})T(\boldsymbol{y},\boldsymbol{x})\lambda(\boldsymbol{x},\boldsymbol{y}), \tag{9}$$

and define $M(\boldsymbol{x}) = E_\pi[\rho(\boldsymbol{x},\mathbf{Y})]$. Then by the weak law of large number (WLLN),

$$\frac{1}{k}\{w(\boldsymbol{y}_1,\boldsymbol{x}) + \cdots + w(\boldsymbol{y}_k,\boldsymbol{x})\} \stackrel{k\to\infty}{\longrightarrow} M(\boldsymbol{x}) \text{ in probability}$$

if and only if $M(\boldsymbol{x}) < \infty$. Similarly, $\{w(\boldsymbol{y},\boldsymbol{x}_1^*) + \cdots + w(\boldsymbol{y},\boldsymbol{x}_k^*)\}/k \to M(\boldsymbol{y})$. To ensure a robust performance of MTM, we may want to impose the condition $\mathrm{var}_{y|x}\{w(\mathbf{Y},\boldsymbol{x})\} < \infty$ so that the strong law holds.

Under the limiting situation, the proposal distribution of $\boldsymbol{y}$ is, therefore,

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\pi(\boldsymbol{y})\rho(\boldsymbol{x},\boldsymbol{y})}{M(\boldsymbol{x})}.$$

Intuitively, the proposal transition function $T(\boldsymbol{x},\boldsymbol{y})$ is first "symmetrized" and then applied. If we choose $\lambda(\boldsymbol{x},\boldsymbol{y})$ as in (6) with $\alpha = 1$ in the region $A_{\boldsymbol{x}} = \{\boldsymbol{y} : T(\boldsymbol{x},\boldsymbol{y}) > 0\}$, we would have $\rho(\boldsymbol{x},\boldsymbol{y}) = 1$ in $A_{\boldsymbol{x}}$, $M(\boldsymbol{x}) = \pi(A_{\boldsymbol{x}})$, and $p(\boldsymbol{y} \mid \boldsymbol{x}) \propto \pi(\boldsymbol{y})$ with $\boldsymbol{y} \in A_{\boldsymbol{x}}$. If $T(\boldsymbol{x},\cdot)$ is restricted to be uniform in a region A regardless of $\boldsymbol{x}$, each MTM step converges to an exact draw from $\pi(\boldsymbol{x})$ conditioned on that $\boldsymbol{x} \in A$. This is essentially a Gibbs sampling update.

Realistically, the efficiency of a MTM sampler relies on the calibration between the proposal step-size, the number of multiple trials k, and the landscape of the target density π. A further theoretical understanding on these issues is desirable.

3.3 MTM and the General Formulation of Reversible Markov Chains

A general way of producing a reversible Markov chain with π as its invariant distribution is to let the actual transition function from $\boldsymbol{x}$ to $\boldsymbol{y}$ (for $\boldsymbol{y} \neq \boldsymbol{x}$) be of the form

$$A(\boldsymbol{x},\boldsymbol{y}) = \pi(\boldsymbol{y})\delta(\boldsymbol{x},\boldsymbol{y}), \tag{10}$$

where $\delta(\boldsymbol{x}, \boldsymbol{y})$ is any suitable (nonnegative) symmetric function in $\boldsymbol{x}$ and $\boldsymbol{y}$ (C. Stein, personal communication). One can easily verify that $A(\boldsymbol{x}, \boldsymbol{y})$ given by (10) satisfies the detailed balance condition. However, the choice of $\delta(\boldsymbol{x}, \boldsymbol{y})$ is far from arbitrary because of the constraint that

$$\int_{\boldsymbol{y} \neq \boldsymbol{x}} A(\boldsymbol{x}, \boldsymbol{y}) dy \leq 1, \quad \text{for almost all } \boldsymbol{x}.$$

In contrast, the choice of $\lambda(\boldsymbol{x}, \boldsymbol{y})$ in a MTM sampler is nearly arbitrary. Clearly, function $A(\boldsymbol{x}, \boldsymbol{y})$ in a Metropolis-Hastings algorithm is of the form $\pi(\boldsymbol{y})\delta_{\mathrm{MH}}(\boldsymbol{x}, \boldsymbol{y})$ with

$$\delta_{\mathrm{MH}}(\boldsymbol{x}, \boldsymbol{y}) = \min\left\{\frac{T(\boldsymbol{x}, \boldsymbol{y})}{\pi(\boldsymbol{y})}, \frac{T(\boldsymbol{y}, \boldsymbol{x})}{\pi(\boldsymbol{x})}\right\},$$

which is apparently symmetric in $\boldsymbol{x}$ and $\boldsymbol{y}$. In contrast, the limiting transition function of MTM corresponds to

$$\delta_{\mathrm{MTM}}(\boldsymbol{x}, \boldsymbol{y}) = \rho(\boldsymbol{x}, \boldsymbol{y}) \min\{M^{-1}(\boldsymbol{x}), M^{-1}(\boldsymbol{y})\},$$

with ρ defined in (9). This form might be useful in comparing different choices of $\lambda(\boldsymbol{x}, \boldsymbol{y})$ in MTM. Since $M(\boldsymbol{x})$ is obtained by averaging π over a "neighborhood" of $\boldsymbol{x}$, the resulting transition makes it more likely to move away from $\boldsymbol{x}$. Because of the construction of MTM, the asymptotic convergence property of the sampler is similar to that of a Metropolis-Hastings sampler. See Roberts and Tweedie (1996) for methods of analyzing such a chain.

4 Using Local Optimization for Adaptation in MCMC

In this section we show how to combine the MTM with the ADS method of Gilks et al. (1994) to produce a better sampler. The 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.

4.1 Adaptive Direction Sampling

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

Gilks et al. (1994) and Roberts and Gilks (1994) show that $f(r)$ should be of the form

$$f(r) \propto |r|^{d-1}\pi(\boldsymbol{X}_t^{(a)} + r\boldsymbol{e}_t),$$

where d is the dimensionality of the state space. 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 $\boldsymbol{e}_t$. 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.

4.2 Local Optimization-Based MTM

We follow the ADS approach of evolving a population of samples, say, $\mathcal{S}_t = \{\boldsymbol{X}_t^{(1)}, \ldots, \boldsymbol{X}_t^{(m)}\}$, at each iteration. To update one of the samples, say, $\boldsymbol{X}_t^{(c)}$, we use the other samples to construct a good reference point $\mathbf{Y}_t$, and then update $\boldsymbol{X}_t^{(c)}$ by an MTM transition along the direction defined by $\boldsymbol{X}_t^{(c)}$ and $\mathbf{Y}_t$. With the theory established by Roberts and Gilks (1994), one can see that essentially *any* way of choosing the reference point $\mathbf{Y}_t$ is appropriate provided that $\mathbf{Y}_t$ is independent of $\boldsymbol{X}_t^{(c)}$ 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 $\mathcal{S}_t = \{\boldsymbol{X}_t^{(1)}, \ldots, \boldsymbol{X}_t^{(m)}\}$. At time $t+1$,

Conjugate-Gradient Monte Carlo (CGMC):

1. Choose $\boldsymbol{X}_t^{(r)} \in \mathcal{S}_t$ at random. Obtain either the gradient of $\log \pi(\boldsymbol{x})$ at $\boldsymbol{X}_t^{(r)}$ or the conjugate gradient of the function by using both the gradient at $\boldsymbol{X}_t^{(r)}$ and direction $\boldsymbol{X}_t^{(r)} - \boldsymbol{X}_{t-1}^{(r)}$. Denote this direction as $\boldsymbol{u}_t$ and conduct a *deterministic* search to find the mode (anchor point), $\boldsymbol{A}_t$, of π along $\boldsymbol{X}_t^{(r)} + r\boldsymbol{u}_t$.

2. Choose another member $\boldsymbol{X}_t^{(c)} \neq \boldsymbol{X}_t^{(r)}$ from $\mathcal{S}_t$ at random.

3. Let $\boldsymbol{e}_t = (\boldsymbol{A}_t - \boldsymbol{X}_t^{(c)})/\|\boldsymbol{A}_t - \boldsymbol{X}_t^{(c)}\|$, and sample along the line $\boldsymbol{A}_t + r\boldsymbol{e}_t$ by the MTM method, with the target distribution for $r \in (-\infty, \infty)$ being

$$f(r) \propto |r|^{d-1}\pi(\boldsymbol{A}_t + r\boldsymbol{e}_t), \tag{11}$$

   where d is the dimensionality of the state space.

The gradient/conjugate gradient procedure in step 1 can be iterated for any number of times and can also be replaced by *any* effective local optimization method, such as the iterative conditional maximization 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), page 418. In steps 2 and 3 of CGMC, one can choose any number of chains in $\mathcal{S}_t \setminus \{\boldsymbol{X}_t^{(r)}\}$ to move towards anchor point A_t simultaneously. This modification sometimes can improve the performance of CGMC.

The population size m needs not be too large. In fact we found that having $m = 2$ to 5 was good enough for all of our examples. 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 4.1 *If we have population $\mathcal{S}_t = \{\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(m)}\}$ in a CGMC setting, then the invariant distribution of $\mathcal{S}_t$ under the CGMC move is $\pi(\boldsymbol{x}_t^{(1)}) \times \cdots \times \pi(\boldsymbol{x}_t^{(m)})$.*

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

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

Without loss of generality, we assume that in the next step $\boldsymbol{x}_t^{(1)}$ is chosen to be updated, and $\boldsymbol{a}_t$ is the anchor point which is obtained by a local optimization step started from $\boldsymbol{x}_t^{(m)}$, say. Because the local optimization step can be viewed as a function of $\boldsymbol{x}_t^{(m)}$, we can write $\boldsymbol{a}_t = g(\boldsymbol{x}_t^{(m)})$. Since $\boldsymbol{x}_t^{(m)}$ follows π and is independent of $\boldsymbol{x}_t^{(1)}$, the anchor point $\boldsymbol{a}_t$ must also be independent of $\boldsymbol{x}_t^{(1)}$. 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 4.1 *Suppose $\boldsymbol{x} \sim \pi$ and $\boldsymbol{y}$ is any fixed point in a d-dimensional space. Let $\boldsymbol{e} = (\boldsymbol{x} - \boldsymbol{y})/\|\boldsymbol{x} - \boldsymbol{y}\|$ be a unit vector. If r is drawn from distribution $f(r) \propto |r|^{d-1}\pi(\boldsymbol{y} + r\boldsymbol{e})$, then $\boldsymbol{x}' = \boldsymbol{y} + r\boldsymbol{e}$ follows distribution π. If $\boldsymbol{y}$ is generated from a distribution $D(\boldsymbol{y})$ independent of $\boldsymbol{x}$, then $\boldsymbol{x}'$ is independent of $\boldsymbol{y}$ and has density $\pi(\boldsymbol{x}')$.*

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

$$E\{h(\boldsymbol{x}')\} = E[E\{h(\boldsymbol{x}') \mid \boldsymbol{x}\}] = \int\int h(r\boldsymbol{x}/\|\boldsymbol{x}\|)\frac{|r|^{d-1}\pi(r\boldsymbol{x}/\|\boldsymbol{x}\|)}{\int |r'|^{d-1}\pi(r'\boldsymbol{x}/\|\boldsymbol{x}\|)dr'}\pi(\boldsymbol{x})drd\boldsymbol{x}.$$

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

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

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

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

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

5 Other Variations of the Multiple-Try Metropolis

In this section, we describe two other ways of using the MTM in MCMC sampling. The first method is closely related to the hit-and-run algorithm of Chen and Schmeiser(1993); and the second one, to the griddy Gibbs sampler (Ritter and Tanner 1992).

5.1 Random-Ray Monte Carlo

Hit-and-Run (HR) Algorithm. For a given current sample $\boldsymbol{X}_t$ one does the following: (a) uniformly select a random direction $\boldsymbol{e}_t$; (b) sample a scalar r_t from density $f(r) \propto \pi(\boldsymbol{X}_t + r\boldsymbol{e}_t)$; and (c) update $\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + r_t\boldsymbol{e}_t$. This algorithm behaves like a random-direction Gibbs sampler and it tends to be very helpful if the probability landscape of π consists of distinctive modes along non-coordinate directions.

A main difficulty in implementing the HR algorithm, however, is that sampling from $f(r)$ is rarely feasible in practice; one often ends up using only a single step of a Metropolis-type update (Chen and Schmeiser 1993). The MTM method combats this limitation by using a relatively flat and long-tailed proposal transition $T(r_1, r_2)$ (along direction $\boldsymbol{e}_t$) that corresponds to a very large searching range. As discussed in Section 3.2, as $k \to \infty$ and when T diverges to a uniform distribution, the resulting algorithm is exactly an HR sampler. Suppose that the current state is $\boldsymbol{X}_t = \boldsymbol{x}$, the random-ray algorithm executes the following update:

- Randomly generates a direction (a unit vector) $\boldsymbol{e}$.

- Draws $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$ from the proposal transition $T_{\boldsymbol{e}}(\boldsymbol{x}, \cdot)$ along the direction $\boldsymbol{e}$. A generic choice is to draw i.i.d. samples $r_1, \ldots, r_k$ from $N(0, \sigma^2)$, where σ can be chosen rather big, and set $\boldsymbol{y}_j = \boldsymbol{x} + r_j \boldsymbol{e}$. Another possibility is to draw $r_j \sim \text{Unif}[-\sigma, \sigma]$.

- Conducts the MTM, as described in Section 3.1.

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

5.2 Griddy-Gibbs MTM

The Gibbs sampler differs from the Metropolis algorithm in its emphasis on the use of conditional distributions. When sampling from a certain conditional distribution is not analytically achievable, Ritter and Tanner (1992) propose an approximate method, the griddy Gibbs sampler, as an alternative. The method has been successfully applied to several statistical problems (Barnard, *et al.* 1998), but its approximate nature still prevents it from being widely used. Using the MTM method, we can design a sampler, the MTM-Gibbs, that resembles the griddy Gibbs both in form and in performance but is *exact* in the sense that π is invariant under the sampler's transition. 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 it is of interest to draw from $\pi(\boldsymbol{x})$. With the current state $\boldsymbol{X}_t = \boldsymbol{x}$, the transition rule of the MTM-Gibbs is as follows:

- Pick any component, say $x(i)$. Draw $y_1, \ldots, y_k$ i.i.d. from a transition $T(x(i), \cdot)$ along the axis of $x(i)$, and evaluate

$$w(y_j, x(i)) = \pi(x(1), \ldots, x(i-1), y_j, x(i+1), \ldots, x(d)) T(y_j, x(i)) \lambda(y_j, x(i)),$$

  for $j = 1, 2, \ldots, k$. The transition T is *allowed* to depend on $\boldsymbol{x}[-i]$. The simplest such choice is $T(x(i), y) \equiv c$, i.e., uniform, in the range of $x(i)$, when the range is finite. Alternatively, one often chooses $t = x(i) + \epsilon$ with $\epsilon \sim N(0, \sigma^2)$.

- Choose $y{=}y_j$ with probability proportional to $w(y_j, x(i))$. Draw $k-1$ i.i.d. samples from $T(y, \cdot)$, say $s_1, \cdots, s_{k-1}$. Name $s_k = x(i)$.

- Compute the generalized Metropolis ratio r_g as in equation (3); accept y with probability r_g and reject with $1 - r_g$.

We applied the above algorithm to an example of Ritter and Tanner (1992, §6.1) in which they fitted a nonlinear model

$$y = \theta_1 (1 - \exp(-\theta_2 x)) + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

to a real dataset. The prior of σ^2 was chosen to be flat on $(0, \infty)$; and the prior of (θ_1, θ_2) was uniform in the region $[-20, 50] \times [-2, 6]$. Note that the posterior would have been improper had a flat improper prior for (θ_1, θ_2) been chosen.

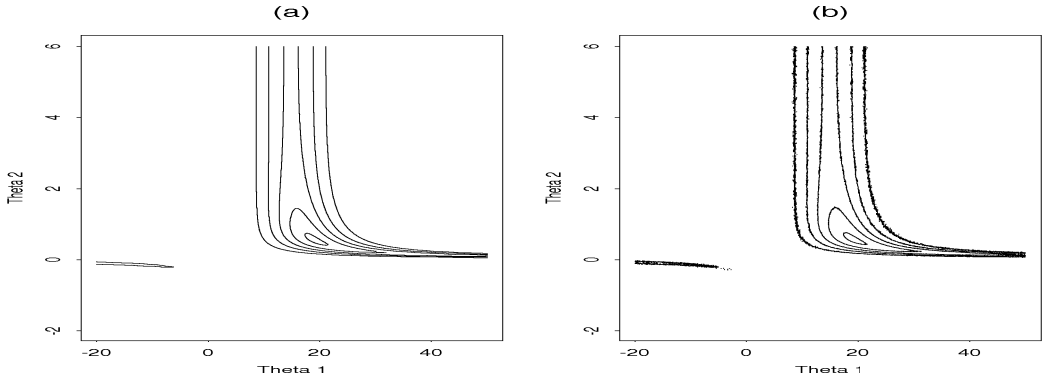

Figure 1: A comparison of (a) the exact contour plots with (b) the one estimated by the MTM-Gibbs samples (b).

With $k = 10$, we let the proposal of a MTM-Gibbs be $N(\theta, 25^2)$ in the θ_1 direction and $N(\theta, 3^2)$ in the θ_2 direction. The starting point was randomly drawn from the unit square. We ran a total of 4.0e+7 MTM-Gibbs iterations and used the resulting MC samples to estimate the contour lines corresponding to 85%, 90%, 95%, 99% and 99.9% of the mass, respectively. Figure 1(a) shows the true contour lines (from a theoretical evaluation) and Figure 1(b) shows the estimated ones. Even though the mass of the second mode is very small (approximately 0.32%), the sampler gave us a fairly accurate estimate, 0.31%. The CGMC algorithm was also applied to the example with the following setting: the number of streams m=3 and the number of multiple tries k=10. The proposal function for the line search was $N(x, 25^2)$, and that for the local Metropolis was Unif [-4,4] along a randomly chosen direction. We observed that the CGMC produced about 50% more effective samples than the griddy-Gibbs in one unit of computing time.

6 Numerical Examples

6.1 A Simple Unimodal Example for Comparisons

We first applied the MTM algorithm to sample from a simple t-distribution with 5 degrees of freedom. The main purpose of this exercise is to compare the performance of the MTM sampler with that of a Metropolis sampler and a Langevin-Euler sampler for a smooth, but

long-tailed, target distribution. The Langevin-Euler scheme can be understood as a special Metropolis-Hastings algorithm. Identical to the setting of Stramer and Tweedie (1998), we used the proposal function $T(x, \cdot) = N(x, 1)$ for the Metropolis algorithm, and used the proposal function

$$T(x, \cdot) = N(x + \frac{1}{2}h\nabla \log \pi(x),\ h) \tag{12}$$

with $h = 1$ for the Langevin-Euler scheme. A scheme like this tends to perform well when the target distribution is smooth, but the type of move is far too "local" when the target distribution has a rough landscape.

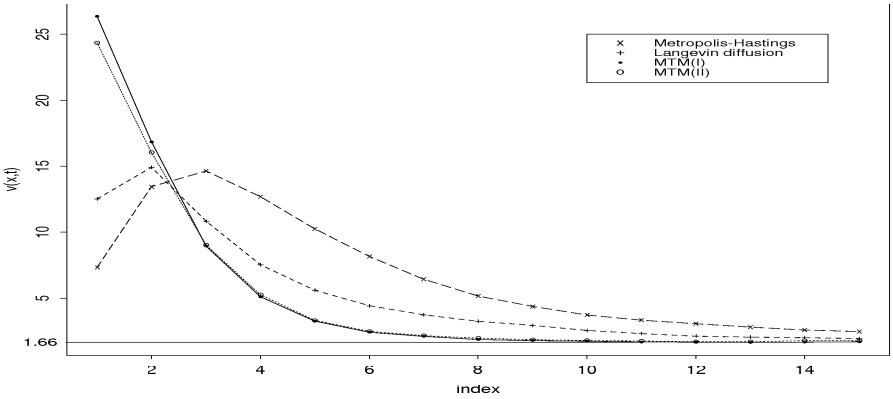

Figure 2: A comparison of the MTM scheme with the Metropolis and Langevin-Euler schemes. The curves have been adjusted for computational costs. That is, those points with the same index value correspond to the same computing time (instead of iterations).

Following Stramer and Tweedie, we let all the competing samplers start from $x = 9$ and monitored $v(x, t)$ (computed by averaging over 5000 replications), the mean estimation of the variance of π by using the first t consecutive MCMC samples (see Stramer and Tweedie for more details). As $t \to \infty$, $v(x, t)$ should converge to 1.66. We applied both MTM (I) and MTM (II) with $k = 5$ to this problem. For MTM (I), the proposal function is $T(x, \cdot) = N(x, \sigma_x^2)$, where $\sigma_x = \max\{10, |x|\}$. For MTM (II), the symmetric proposal is $N(x, 10^2)$. Figure 2 shows that the two MTM schemes performed rather similarly, and both outperformed the Metropolis and the Langevin-Euler scheme (after taking into consideration the extra computational cost). More importantly, we note that the MTM strategy can be applied in combination with all the advanced Metropolis-Hastings type algorithms (including the Langevin-type and the self-targeting type algorithms) to achieve better performance.

6.2 A Multimodal Example

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

$$.34 \times N_2(\mathbf{0}, I_2) + .33 \times N_2\left\{\begin{pmatrix} -6 \\ -6 \end{pmatrix}, \begin{pmatrix} 1 & .9 \\ .9 & 1 \end{pmatrix}\right\} + .33 \times N_2\left\{\begin{pmatrix} 4 \\ 4 \end{pmatrix}, \begin{pmatrix} 1 & -.9 \\ -.9 & 1 \end{pmatrix}\right\}.$$

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

We started two independent Metropolis samplers with starting points drawn from Unif$[-.5, .5]^2$. A spherical proposal function was employed: a direction was generated uniformly and then the radius drawn from Unif$[0, a]$, where a(=4 in our case) is calibrated so that the Metropolis sampler had an acceptance rate of about 0.23 (Gelman, Roberts and Gilks, 1996). 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 3, we plotted the histograms and autocorrelations for one of the variables (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 were very poorly estimated.

The CGMC method was applied to this problem with m=2 streams and 20,000 iterations for each. Each iteration consists of two Metropolis steps and one adaptation step. So a total of 100,000 random draws from π, which took about 27 seconds of CPU time from the same computer, were produced as the program ended. 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 CGMC, 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 3, we plotted the histograms and autocorrelations for one of the variables in one stream (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

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

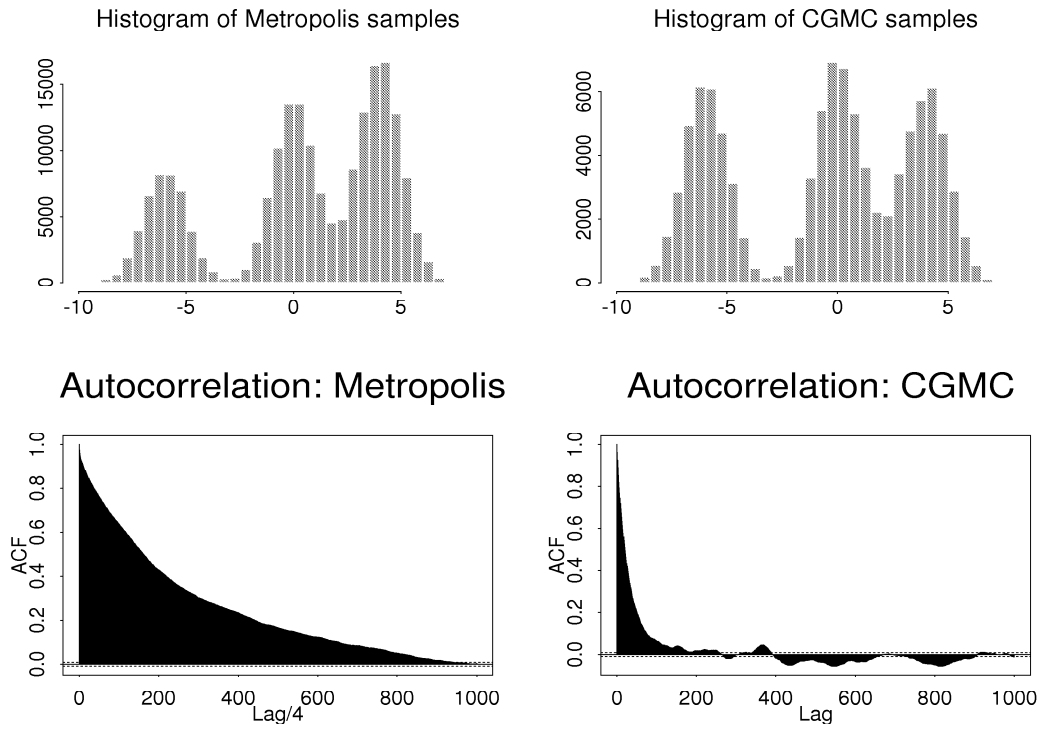

Figure 3: A comparison of results obtained by the Metropolis sampler and by the CGMC. The autocorrelation plot of the Metropolis samples has taken the computational cost into account.

where $\mathbf{0} = (0, \ldots, 0)$ and $\mathbf{5} = (5, \ldots, 5)$ (the distance between the two modes is $5\sqrt{5} = 11.2$). We applied both the Metropolis algorithm (with proposal $N(x, 1.5^2)$) and the Langevin-Euler scheme (12) to this distribution and they performed similarly. In all of the 70 million iterations, the Langevin-Euler scheme with $h = 2.8$ (acceptance rate 0.247) was not able to escape from the mode in which it was started.

We started the CGMC with $m = 2$ streams and the initial values drawn from Unif$[-.5, .5]^5$ (e.g., both streams were started from the first mode). Each iteration of the CGMC algorithm consists 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 uses 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 were insensitive to the choice of step-size in this range. Two general guidelines are: (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); (b) the proposal step size for the Metropolis should give an acceptance rate in the range of (0.35,0.5).

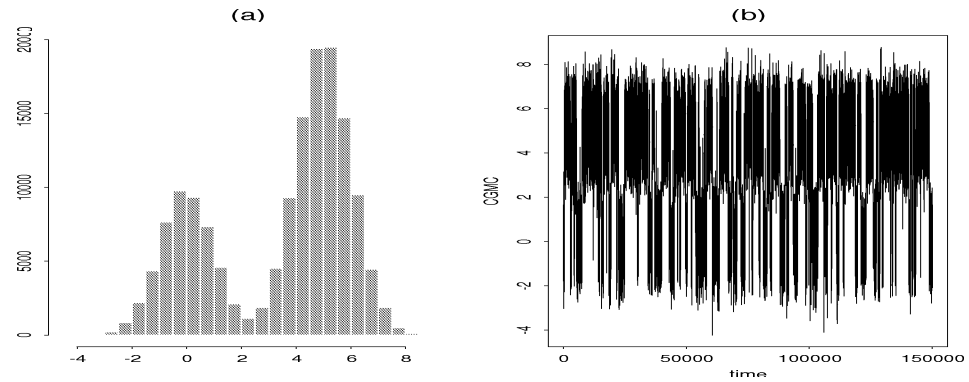

Figure 4: The mixing result of the CGMC samples: (a)a marginal histogram of the sample; (b) time series plot of one stream of the CGMC.

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 4, we plotted the histogram and the time 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 working is the effectiveness of line-sampling.

6.3 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 $a_j = 0$, $j = 1, 2, 3$, 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

$$L(\boldsymbol{\theta} \mid y_1, \ldots, y_n) \propto \frac{1}{\sigma^n} \exp\left\{ -\frac{\sum_{i=1}^{n} (y_i - \sum_{j=1}^{3} e^{-a_j - b_j x_i} \cos(c_j x_i + d_j))^2}{2\sigma^2} \right\}, \tag{13}$$

which can also be treated as a posterior distribution of $\boldsymbol{\theta}$ with a flat prior. To avoid improper posterior distribution, we restricted b_j, c_j and d_j in a bounded region $[-1, 1] \times [-3, 3] \times [-3, 3]$.

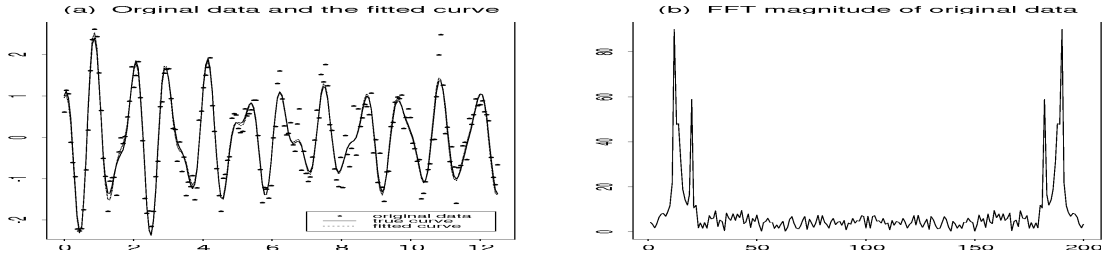

Figure 5: (a) The simulated sinusoidal data and the fitted curve. (b) The Fourier transform of the data.

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 5 (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 for nonlinear least square fitting, failed in this problem. The best fit of the data is shown by dotted lines in Figure 5(a). Posteriors of the unknown quantities, based on histograms obtained from one CGMC run with m =5 chains, are shown in Figure 6.

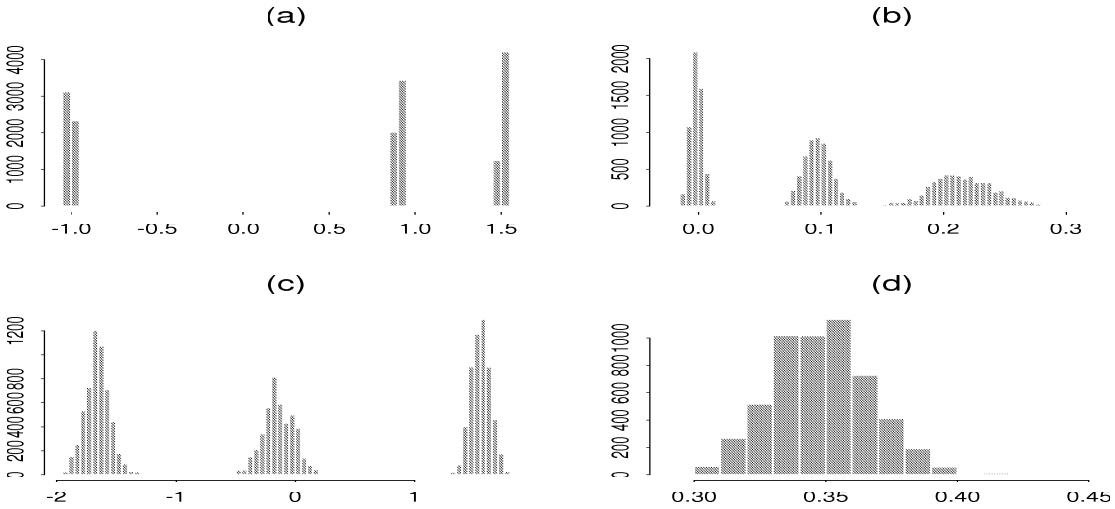

Figure 6: Using Monte Carlo samples produced by the CGMC, we obtained histograms of (a) three frequencies (true values c_j= .9, 1, and 1.5); (b) three damping coefficients (true values b_j=0, .1, and .2); (c) three phases (true values d_j=-1.57, 0, and 1.57); (d) the error standard deviation (true value σ=.3).

This Bayesian curve-fitting procedure was applied to analyze the recording of bird's chirps contained in the standard MATLAB 4.2 demo folder. The whole data file consists of 13,129 digital signals, as plotted in Figure 7 (a). We took a segment of length 100 (segment [301: 400]) from it for analysis (plotted in Figure 7 (b)). Two related Fourier transforms (of two segments) are shown in in Figure 7 (c) and (d). As one can see, the fast Fourier transform (FFT) of the data segment [201:400] revealed two distinct frequency peaks, whereas only one frequency showed up when FFT was applied to segment [301:400]. Even with extensive zero padding, we still could not see the two distinct frequencies.

In contract, our Bayesian curve fitting procedure (with $J = 3$) found the two major frequency components at 4.22 and 2.14, respectively, which correspond well to the two components revealed by the Fourier analysis of the longer data segment [201:400]. We also found another minor frequency at 1.95 with a large damping coefficient. The three components are plotted in Figure 8. The fitted value (solid line) agreed very well with the original data.

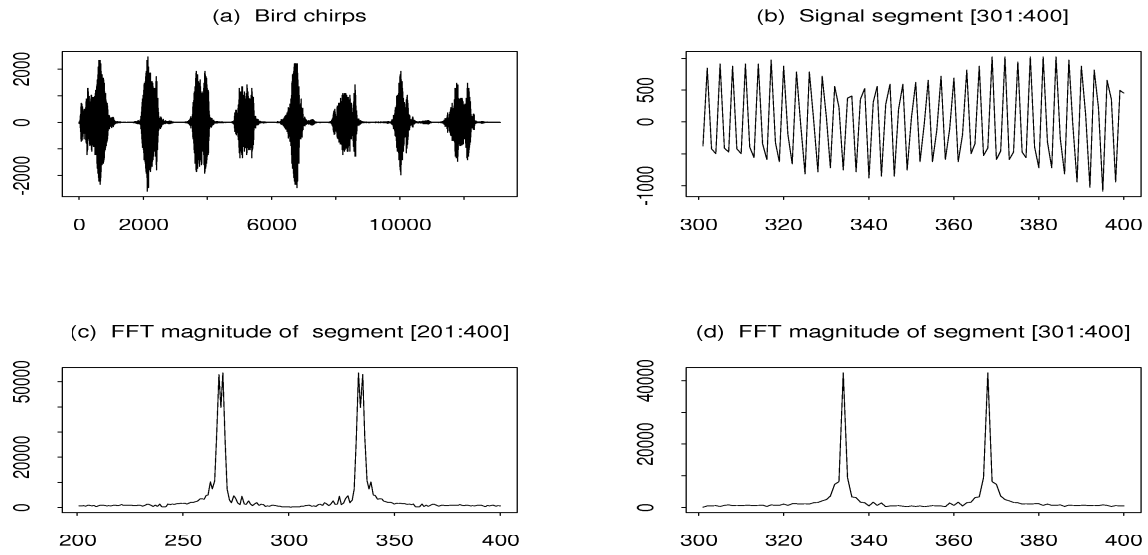

Figure 7: (a) Plot of the bird chirp sequence; (b) the plot of a segment ([301:400]) of the original sound signal sequence; (c) the Fast-Fourier transform (FFT) of data segment [201:400]; (d) the FFT of segment [301:400].

6.4 Fitting Mixture Models

Suppose $y_1, \ldots, y_n$ are i.i.d. 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. For technical reasons, we performed a parameter transformation $u_1 = \log p_1$ and $u_2 = \log p_2$, and let our sampler operate on $\boldsymbol{\theta} = (u_1, u_2; \mu_i, \log \sigma_i, i = 1, 2, 3)$. With a flat prior on $(\log \sigma_i, \mu_i;\ i = 1, 2, 3)$ in a bounded region $[-4, 4]^3 \times [-10, 10]^3$, and a prior $p(u_1, u_2) \propto (1 - e^{u_1} - e^{u_2})_+$ with $(u_1, u_2) \in [-10, 0] \times [-10, 0]$, we obtained an expression for the posterior distribution of these parameters, up to a normalizing constant. To cope with the non identifiability problem, we imposed the constraint that $\mu_1 \geq \mu_2 \geq \mu_3$.

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 9(a). A Gibbs sampler can be designed

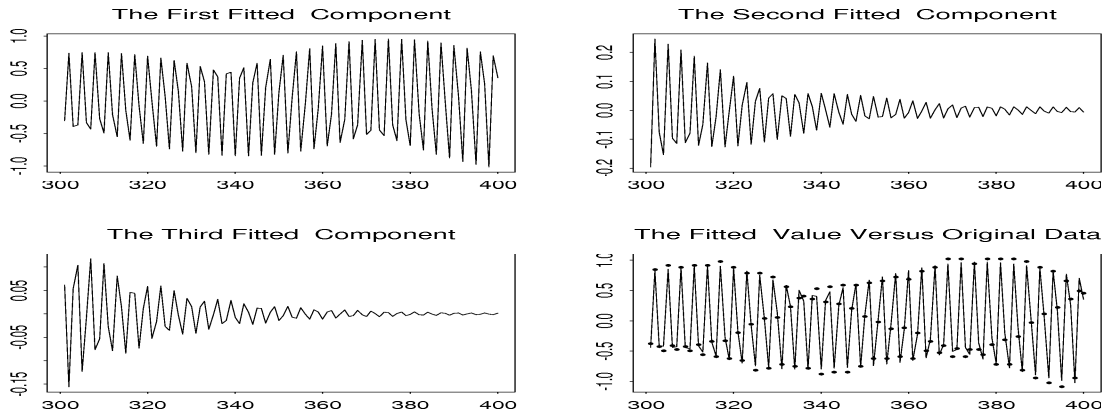

Figure 8: Results from the CGMC fitting. The dots in lower right figure correspond to the original signals, plotted in Figure 7 (b), after rescaling.

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 used a generic proposal: uniformly choosing a direction in the $\boldsymbol{\theta}$ space, and then sampling a distance from Unif[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 was apparently unfavorable for the simulation because it put 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 remainder 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 were inserted. Thus, "one-iteration" in this algorithm consisted 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 9 (d) shows the time series plot of the log-likelihood for the 4 streams of a randomly chosen CGMC run.

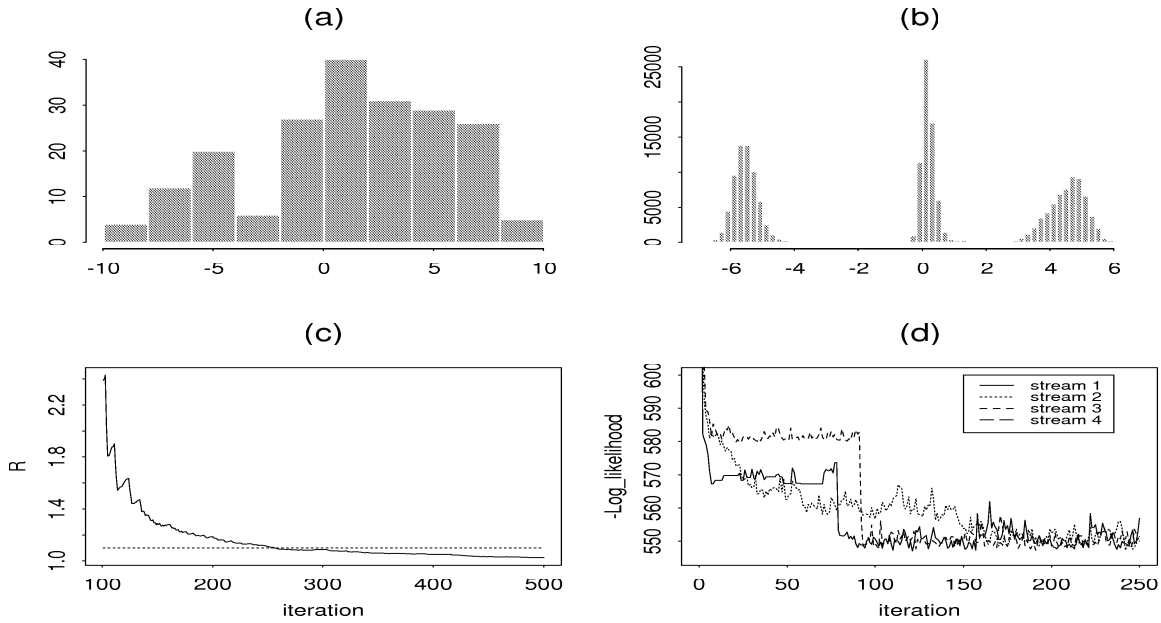

Figure 9: (a) Histogram of the data; (b) the posterior distribution of the μ_i, with the samples obtained from the CGMC; (c) Gelman-Rubin convergence criterion $\hat{R}$; (d) time series plot of the 4 streams in the CGMC. Every iteration unit in both (c) and (d) corresponds to 1 step of line-sampling and 20 steps of Metropolis in simulation.

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 9(d)) together with the Gelman-Rubin statistic $\hat{R}$ (Figure 9(c)) served convergence diagnosis purposes very well: an $\hat{R} < 1.2$ computed with m=4 chains always indicated that the stationarity was reached. Although a large $\hat{R}$ does indicate non-convergence, caution is needed when claiming convergence: because the multiple streams used in the CGMC are not completely independent before they reach equilibrium, the resulting $\hat{R}$ is not as reliable an indicator as that from independent runs. A quick remedy is to execute a small number (2 to 5) of independent CGMC runs for computing the G-R statistic.

The foregoing mixture model fitting procedure was applied to the galaxy dataset analyzed

by Roeder and Wasserman (1997) The original data were first rescaled by 5000 and then fitted by a 4-component Gaussian mixture model. Figure 10(a) shows the overlay plot of 20 sampled predictive densities for the galaxy dataset and Figure 10(b) shows the overlay plot of the maximum *a posteriori* predictive density.

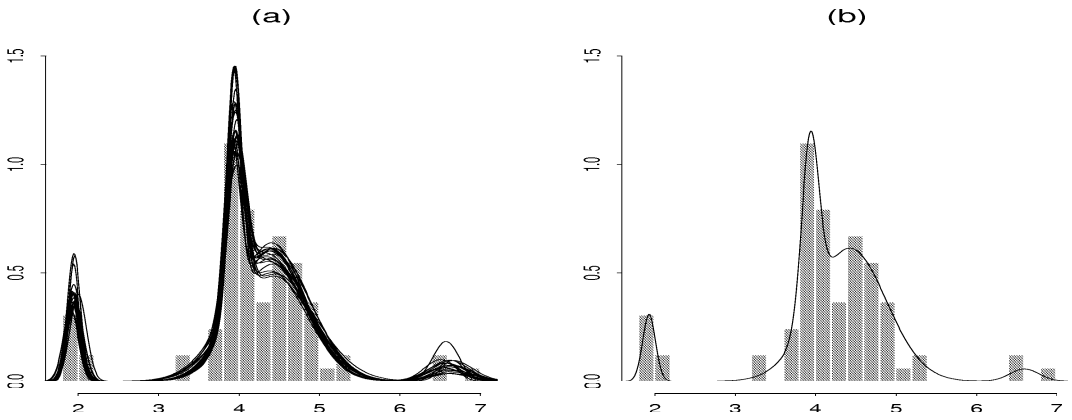

Figure 10: (a) Random draws of the posterior predictive densities for the galaxy data in Roeder and Wasserman (1997); (b) The maximum *a posteriori* estimate of the predictive density.

7 Discussion

We have proposed two novel ideas and their variations 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 these ideas, and their usefulness has been carefully examined through several numerical examples. Generally speaking, the multiple-try Metropolis enables us to make large step-size transitions in a MCMC sampler. It is particularly useful when one is able to identify certain directions of interest but then has difficulty to implement a Gibbs-sampling type move because of unfavorable conditional distributions.

It has been suggested by some researchers that Monte Carlo efficiency might 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 mode-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," *Journal of the Royal Statistical Society,* Ser. B, 58, 397-415.

Chen, M.-H. and Schmeiser, B.W. (1993), "Performances of the Gibbs, hit-and-run, and Metropolis samplers," *Journal of Computational and Graphical Statistics*, 2, 251-272.

Cracknell, R.F., Nicholson, D., Parsonage, N.G., and Evans, H. (1990), "Rotational Insertion Bias: a Novel Method for Simulating Dense Phases of Structured Particles, With Particular Application to Water," *Molecular Physics*, 71, 931-943.

Diebolt, J. and Robert, C.P. (1994), "Estimation of Finite Mixture Distribution Through Bayesian Sampling," *Journal of the Royal Statistical Society,* 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 Transactions on Pattern Analysis and Machine Intelligence*, 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*, 41, 337-348.

Gilks, W.R., Roberts, R.O., and Sahu, S.K. (1998), "Adaptive Markov Chain Monte Carlo Through Regeneration," *Journal of the American Statistical Association,* 93, 1045-1054.

Goodman, J. and Sokal, A.D. (1989), "Multigrid Monte Carlo Method. Conceptual Foundations," *Physical Review D*, 40 2035-2071.

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, 1087-1092.

Mezei, M. (1980), "A Cavity-Biased (T, V, μ) Monte Carlo Method for the Computer Simulation of Fluids," *Molecular Physics,* 40, 901-906.

Ritter, C. and Tanner, M.A. (1992), "Facilitating the Gibbs Sampler: the Gibbs Stopper and the Griddy-Gibbs Sampler," *Journal of the American Statistical Association,* 87, 861-868.

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

Roberts, G.O. and Tweedie, R. (1996), "Geometric Convergence and Central Limit Theorems for Multidimensional Hastings and Metropolis Algorithms," *Biometrika*, 83, 95-110.

Roeder, K. and Wasserman, L. (1997), "Practical Bayesian Density Estimation Using Mixtures of Normals," *Journal of the American Statistical Association,* 92, 894-902.

Stramer, O. and Tweedie, R. (1998), "Self-Targeting Candidates for Hastings-Metropolis Algorithms," *Technical Report,* University of Iowa.