UNIVERSITY of CALIFORNIA
SANTA CRUZ

MIRROR-JUMP SAMPLING: A STRATEGY FOR MCMC ACCELERATION

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Shufeng Liu

May 2003

The thesis of Shufeng Liu is approved:

Professor David Draper, Chair

Professor Manfred K. Warmuth

Vice Provost and Dean of Graduate Studies

Acknowledgements

I wish to thank Professor David Draper for being a wonderful advisor. His superb mentoring, inspiration, encouragement, help, and patience altogether have made my gradate study here a valuable and unforgetful experience.

I want to thank Professor Manfred Warmuth for being my committee member. His help and suggestions are greatly appreciated.

Abstract

Mirror-Jump Sampling: A Strategy for MCMC Acceleration

by

Shufeng Liu

This project explores the effectiveness of "mirror-jump sampling", a special type of Metropolis-Hastings algorithm, as a strategy to enhance accuracy of posterior estimates and accelerate learning process in MCMC simulations by reducing autocorrelations of MCMC output. Mirror-jump sampling makes use of rough estimates of location and scale parameters of the posterior available from maximum likelihood methods or preliminary MCMC output. Our systematic simulation studies show that this strategy is indeed very effective in unimodal posterior settings.

Contents

Chapter 1

Introduction

The main difference between the classical approach and Bayesian approach in statistical analysis lies in the interpretation of parameters of a model. Bayesian approach considers the parameters to be stochastic, while classical approach assumes them to be fixed values. Bayesian statistics is both old and new. The earliest Bayesian work dates back to the 18th century by Bayes and Laplace [T.63]. However, it was limited to very simple problems, because for a relatively complicated model, Bayesian inference is often computationally prohibitive. During the past decade, however, due to the increasing capability of fast computers, there has been a revolution in Bayesian computation, and Bayesian statistics is becoming more and more successful and popular in many scientific disciplines. Nevertheless, how to construct an efficient Bayesian computation model is often not easy and is an active ongoing research topic.

1.1 Bayesian Inference

A typical statistical problem is stated as follows. Some random quantity Y has a sampling distribution $p(y|\theta)$ where θ is unknown model parameter(s). A data set y is generated from this stochastic model. We want to infer θ from this observed data y. The classical statistical inference assumes that θ is some unknown fixed value, and the statistician's job is to find a good estimate of θ. As a Bayesian, one thinks of θ as a random variable, just like Y. So the Bayesian inference will ask for the distribution of θ given the observed data y, denoted as $p(\theta|y)$. Such a conditional distribution is called the posterior density of θ.

The Bayes theorem:

$$p(\theta|y) = \frac{p(y,\theta)}{p(y)} = \frac{p(y|\theta)p(\theta)}{p(y)}$$

where

$$p(y) = \int p(y,\theta)d\theta = \int p(y|\theta)p(\theta)d\theta.$$

Or simply

$$p(\theta|y) \propto p(y|\theta)p(\theta).$$

since $p(y)$ is "just" a scaling factor.

$p(\theta)$ is called the prior distribution of θ. It represents subjective belief on θ before seeing the data. According to Bayes theorem, the posterior density of θ is proportional to the product of the likelihood $p(y|\theta)$ and the prior $p(\theta)$. So the can be regarded as the updated belief on θ after seeing the data.

The controversy on Bayesian statistics mainly originates from the issue of the prior. When we do have prior information about θ, we can incorporate it into the model. But when such information is unavailable, people generally prefer the use of non-informative or diffuse priors, i.e., priors that do not introduce too much subjectivity and do not affect the inference too much.

The framework of Bayesian modeling is clean and clear. But note that in order to get the exact posterior distribution $p(\theta|y)$, we have to compute the integral for $p(y)$, which could often be a daunting task. Often we are interested in estimating a function of θ, say $h(\theta)$:

$$E[h(\theta)] = \int h(\theta)p(\theta|y)d\theta$$

and this too involves evaluation of an integral which might not be analytically solvable.

1.2 Monte Carlo Simulations

The general solution to the above integration problems are Monte Carlo simulations. If we can draw a sufficiently large sample from the posterior distribution $p(\theta|y)$, then we can get approximate $E[h(\theta)]$ by $\frac{1}{N}\sum_{i=1}^{N} h(\theta_i)$ with a high degree of accuracy. All Monte Carlo simulations make use of random sample generation from the $Unif[0,1]$ distribution. The sampling methods described in this section bear the common feature that individual draws are (at least approximately) independent and they are like IID sampling from the target distribution.

1.2.1 Transformation of Variables

Sometimes it is difficult to draw samples directly from a distribution of a variable, but easy to draw samples from a transformation of the variable. One such method is called the inverse CDF method. If the density of a random variable is $f(x)$, we define $U \equiv F(X) = \int_{-\infty}^{X} f(t)dt$, the distribution function of X, then $U \sim Unif[0,1]$. So if we generate a draw u from $Unif[0,1]$, then $F^{-1}(u)$ would be a draw from $f(x)$. When $F(X)$ does not have an analytical expression, one has to approximate it by constructing grids for the density and the distribution. In any case, this method is a direct sampling method and is limited to cases where the density $f(x)$ is known.

1.2.2 Rejection Sampling

There are cases where the target density $p(\theta|y)$ is difficult to sample from but you can find an envelop proposal density $g(\theta)$ such that $p(\theta|y) \leq cg(\theta)$ for all θ, and $g(\theta)$ is easy to sample from. This rejection sampling algorithm works as follows:

1. Sample θ^* from $g(\theta)$;
2. Sample u from $Unif[0,1]$;
3. If $u \leq \frac{p(\theta|y)}{cg(\theta)}$ then $\theta = \theta^*$ else go to step 1.

If the proposed sample from step 1 is rejected, then it is thrown away. To achieve a higher acceptance rate and thus higher efficiency, the constant c should be as small as possible: $c = sup_\theta \frac{p(\theta|y)}{g(\theta)}$.

1.2.3 Importance Sampling

Unlike rejection sampling where a draw is either accepted or rejected, i.e. the weight of a draw is either 1 or 0, in importance sampling [J.89] the weights can possibly take any value representing the relative importance of the draws. Suppose we want to estimate a function of θ, $h(\theta)$. If $p(\theta|y)$ is difficult to sample from but $g(\theta)$ is a candidate density easy to sample from, then:

$$E(h(\theta)|y) = \int \frac{h(\theta)}{p(\theta|y)} d\theta = \int h(\theta) \frac{p(\theta|y)}{g(\theta)} g(\theta) d\theta$$

If we draw a sufficiently large sample from $g(\theta)$, then we can approximate $E(h(\theta)|y)$ with $\frac{1}{N}\sum_{i=1}^{N} h(\theta_i)w(\theta_i)$ where $w(\theta_i) = \frac{p(\theta_i|y)}{g(\theta_i)}$ is called the importance weight (ratio). One potential problem is that we could possibly miss some importance weights that are extremely rare but extremely large, and consequently result in bad estimate. To avoid such situation, the candidate density $g(\theta)$ should have fatter tails than the target density $p(\theta|y)$. Another drawback is that there are no methods which can provide reliable accuracy assessment of an importance sampling estimate.

1.3 Markov Chain Monte Carlo (MCMC) Simulations

1.3.1 Basic Markov Chain Theory

For more details on Markov chain theory, please refer to [Gam97], [J.L53], and [Par62].

A (discrete-time) *Markov Chain* is a stochastic process $\{\theta_t, t \in T, \theta_t \in S\}$ where $T = \{0, 1, ...\}$ and S is the state space, that satisfies the Markovian property:

$$P(\theta_{t+1} \in A|\theta_t, \theta_{t-1}, ..., \theta_0) = P(\theta_{t+1} \in A|\theta_t), A \subseteq S$$

In simple terms, this says that the next state of a Markov chain only depends on the current state, but not the previous states.

Definitions:

• The probability of the chain starting from state x hitting state y at the n^{th} step $\equiv P^n_{yy}$, called the *transition probabilities* of the chain. When $n = 1, P^1_{yy} \equiv P_{yy}$.

• The probability of the chain starting from state x hitting state y at any posterior step $\equiv \rho_{xy} = P_x(T_y < \infty)$.

• A state y is *positive recurrent* iff $\rho_{yy} = 1 \& E(T_y) < \infty$. A chain is positive recurrent iff every state y is positive recurrent.

• A state y is *aperiodic* iff $lcd\{n \geq 1 : P^n_{yy} > 0\} = 1$ where lcd means largest common divisor. A chain is aperiodic iff every state y is aperiodic.

• A state y is *ergodic* iff y is positive recurrent and aperiodic. A chain is ergodic iff every state y is ergodic.

• A chain is *irreducible* iff $\rho_{xy} > 0$ for all $x, y \in S$.

• A distribution π is a *stationary (invariant) distribution* of a chain iff $\sum_{x \in S} \pi(x) P_{xy} = \pi(y)$, for all $y \in S$. When π is unique it is called the *equilibrium (limiting) distribution*.

• A chain with transition matrix P and stationary distribution π is *reversible* iff $\pi(x)P_{xy} = \pi(y)P_{yx}$ for all $x, y \in S$.

Theorems:

• A chain is irreducible and ergodic $\Leftrightarrow$ it has a limiting distribution π.

• A chain is reversible, irreducible, and aperiodic $\Leftrightarrow$ it has a limiting distribution π.

It turns out that most Markov chains used nowadays for simulation are irreducible and aperiodic [L.94], so constructing a chain with a limiting distribution π reduces to finding transition probabilities satisfying the reversibility conditon.

Now back to the integration problem of $p(\theta|y)$. Metropolis (1953) invented an ingenious sampling engine by combining the essence in rejection sampling and the concepts in Markov chain theory, thus opening the door to a new world of Bayesian computation - Markov Chain Monte Carlo (MCMC) simulations. The underlying rationale is: if we can construct a Markov Chain which has $p(\theta|y)$ as its limiting distribution and is easy to simulate from, then after we run the chain for a sufficiently long time (burn-in period), it will have converged to the limiting distribution, and the iterates afterwards will form a random sample from our target posterior density $p(\theta|y)$, and we can thus extract any features of the posterior as we wish.

1.3.2 The Metropolis-Hastings Algorithm

The most general method used in MCMC is the Metropolis-Hastings (M-H) algorithm. It was first introduced by chemical physicists Metropolis et al. (1953) [NAM$^+$53] and later generalized by a statistician Hastings (1970) [WK70].

Let $p(\theta|y)$ be the target density. Given the current state θ_t, We choose a proposal distribution (jumping distribution, transition matrix) $q(\theta|\theta_t)$ that is easy to sample from. The *M-H algorithm* works as follows:

- Assign a feasible initial value to θ_0;
- Do for $t = 0, 1, 2, \ldots, N$:
  - Draw a random sample θ^* from $q(\theta|\theta_t)$;
  - Draw a random sample u from $Unif[0, 1]$;
  - Compute acceptance rate $\alpha(\theta_t, \theta^*) = min\left(1, \frac{p(\theta^*|y)q(\theta_t|\theta^*)}{p(\theta_t|y)q(\theta^*|\theta_t)}\right)$;
  - If $u \leq \alpha(\theta_t, \theta^*)$ then $\theta_{t+1} \leftarrow \theta^*$ else $\theta_{t+1} \leftarrow \theta_t$

If the proposal distribution is symmetric, i.e. $q(\theta_a|\theta_b) = q(\theta_b|\theta_a)$ for all θ_a, θ_b, then $\alpha(\theta_t, \theta^*) = min\left(1, \frac{p(\theta^*|y)}{p(\theta_t|y)}\right)$. This is the *Metropolis algorithm*. When $q(\theta|\theta_t)$ only depends on $(\theta - \theta_t)$, the chain is called a *random walk*. A typical choice of proposal density for a random walk is $N(\theta|\theta_t, s^2)$ or $t_\nu(\theta|\theta_t, s^2)$.

It can be shown that [L.94] the Markov chain constructed by the M-H algorithm satisfies the reversibility condition and it indeed has $p(\theta|y)$ as its limiting distribution as desired.

1.3.3 Gibbs Sampling

A special MCMC algorithm with a acceptance rate of 1 that is very useful in many multidimensional problems is the *Gibbs sampler* [SG84]. When we have a d-dimensional vector of parameters $\theta = (\theta_1, \theta_2, ..., \theta_d)$, it is often difficult to work with $p(\theta|y)$ but sometimes much easier to simulate from the conditional distribution of one component given all the others. At each iteration t of the Gibbs sampler, the full conditional distribution of θ_i give all the other components is $p(\theta_i|\theta^t_{-i}, y)$ where $\theta^t_{-i} = (\theta^{t+1}_1, ..., \theta^{t+1}_{i-1}, \theta^t_i, ..., \theta^t_d)$. Each iteration t has d steps, and each step draws one component from its full conditional given the current values of all other components. The Gibbs sampler works as follows:

- Assign feasible initial values to $\theta^0 = (\theta_1^0, \theta_2^0, ..., \theta_d^0)$;
- Do for $t = 0, 1, 2, ..., N$:
    - Do for $i = 1, 2, ..., d$:
        Draw a random sample θ_i^{t+1} from $p(\theta_i|\theta_{-i}^t, y)$

An obvious requirement for implementation of Gibbs sampling is that all the full conditionals are easy to compute and simulate from.

1.3.4 The Multimodality Problem

In cases where the target distribution has multiple modes, the Markov chain is very likely to get stuck around one particular mode (depending on the initial value of the chain) and never visit the other parts of the state space. This is a very difficult problem to solve and is still an active area of research in Bayesian computation. My work on MCMC acceleration only deals with unimodal posteriors, and it is by no means the goal of this project to solve the multimodality problem.

1.3.5 MCMC Diagnostics

There are two key issues in MCMC implementation. The first one is how long should we wait for the chain to reach stationarity (equilibrium, convergence), and the second one is after reaching equilibrium, how long should we monitor the chain in order to get desired accuracy of posterior estimates.

Convergence Diagnostics

People invented a number of convergence diagnostics to help with the convergence issue, for example, the Geweke diagnostic [J.92], the Gelman-Rubin diagnostic [AD92], the Raftery-Lewis diagnostic [AS92], and the Heidelberger-Welch diagnostic [PP83]. But theoretically speaking we can never be completely sure of reaching convergence, and none of these diagnostic methods is completely safe all the time. This is currently an active area of research in Bayesian computation.

Accuracy Assessment

The realizations of a Markov chain can be viewed as a time series, so we can apply basic time series analysis techniques to Markov chain simulations. Assume we monitor N iterates after convergence is reached. Denote the time series as $\{X_t, t \in T, X_t \in S\}$ where $T = \{0, 1, ...\}$ and S is the state space. By definition, the autocovariance function of a (weakly) stationary time series at lag k is:

$$\gamma_k = E[(X_t - \mu)(X_{t+k} - \mu)]$$

The autocorrelation function at lag k is:

$$\rho_k = \frac{\gamma_k}{\gamma_k}$$

The sample autocovariance function at lag k is:

$$\hat{\gamma_k} = \frac{1}{N} \sum_{t=1}^{N-k} (x_t - \bar{x})(x_{t+k} - \bar{x})$$

The sample autocorrelation function at lag k is:

$$\hat{\rho_k} = \frac{\hat{\gamma_k}}{\hat{\gamma_k}}$$

The *ergodic theorem*: (Markov chain equivalent of the *law of large numbers*):

> If a Markov chain $\{X_t\}$ is ergodic with limiting distribution π and $h(X)$ is a function of X such that $E_\pi[h(X)] < \infty$, then with probability 1:
>
> $$\bar{h}_N = \frac{1}{N} \sum_{t=1}^{N} h(x_t) \to E_\pi[h(X)] \quad as \quad N \to \infty$$

So $\bar{h}_N$ is an asymptotically unbiased estimator of $E_\pi[h(X)]$.

Markov chain equivalent of the *central limit theorem*:

> If a Markov chain $\{X_t\}$ is uniformly geometrically ergodic with limiting distribution π and $h^2(X)$ is integrable wrt. π, then:
>
> $$\frac{\bar{h}_N - E_\pi[h(X)]}{Var_\pi[h(X)]} \to N(0,1) \quad as \quad N \to \infty$$

In general let ρ_k be the autocorrelation function of the series $\{h_t = h(X_t)\}$. Denote the variance of $h(X)$ under the limiting distribution π as $Var_\pi[h(X)] = \sigma_h^2$. It can be shown that in repeated sampling the sampling variance of $\bar{h}_N$ is:

$$Var_\pi(\bar{h}_N) = \frac{\sigma_h^2}{N}\left(1 + 2\sum_{k=1}^{N-1}(1 - \frac{k}{N})\rho_k\right) = \frac{\sigma_h^2}{N}VIF$$

It can also be shown that:

$$Var_\pi(\bar{h}_N) \to \frac{\sigma_h^2}{N}(1 + 2\sum_{k=1}^{\infty}\rho_k) \quad as \quad N \to \infty$$

We call $\sqrt{Var_\pi(\bar{h}_N)}$ the Monte Carlo Standard Error (MCSE) of $\bar{h}_N$. It provides a measure of the accuracy (precision) of $\bar{h}_N$ as a Monte Carlo estimate of the truth $E_\pi[h(X)]$. A smaller MCSE means higher precision.

This provides the grounds and inspiration for our work. Next (chapter 2) we propose and explore a method to effectively achieve higher precision of posterior estimates without having to resort to longer monitoring of the Markov chain.

Chapter 2

Mirror-Jump Sampling

2.1 Motivation

Assume the Markov chain $\{\theta_t, t \in \{0, 1, ...\}\}$ has limiting distribution $p(\theta|y)$, which is our posterior (target) distribution of interest. After the chain reaches equilibrium, we take N realizations from the chain as a random sample from the posterior. One important aspect of our interest in extracting features of $p(\theta|y)$ is to estimate the posterior mean $E_\pi(\theta)$ with the Monte Carlo estimate $\bar{\theta}_N$. The Monte Carlo Standard Error (MCSE) of this estimate can be computed as follows (from chapter 1):

$$MCSE^2(\bar{\theta}_N) = Var_\pi(\bar{\theta}_N) = \frac{\sigma_\theta^2}{N} VIF = \frac{\sigma_\theta^2}{N}\left(1 + 2\sum_{k=1}^{N-1}(1 - \frac{k}{N})\rho_k\right)$$

where $Var_\pi(\theta) = \sigma_\theta^2$ and $\{\rho_k\}$ are the autocorrelation coefficients of the series $\{\theta_t\}$. In repeated sampling, say we have M chains each of size N, we can estimate $MCSE(\bar{\theta}_N)$ with Mean Squared Error, given by:

$$MSE(\bar{\theta}_N) = \frac{1}{M}\sum_{j=1}^{M}(\bar{\theta}_N^j - \bar{\theta})^2, \quad where \quad \bar{\theta} = \frac{1}{M}\sum_{j=1}^{M}\bar{\theta}_N^j$$

Markov chains generated from commonly used MCMC algorithms, such as random walk sampling or Gibbs sampling, almost always have considerable positive autocorrelations. The reason is obvious: for example for random walk type of Markov chains, the proposed move for the next state is always relative to the current state, which of course would induce positive serial correlations of adjacent states. The consequence of having positive autocorrelations is that it makes VIF (variance inflation factor) significantly larger than 1, with 1 being what we would get if IID sampling were available. In other words, positive autocorrelations slow down our learning process about the target distributions. If we could design the Markov chain in such a way that it has smaller autocorrelations, then we could reduce VIF and MCSE, thus increase the precision of posterior estimates, effectively accelerate our learning process about the posterior. Better still, if we could achieve negative autocorrelations, then VIF would be smaller than 1 and we could do even better than IID sampling! This may sound too good to be true, but we indeed achieved this in some cases of our simulation studies!

In the physics and statistics literature, there has been some work devoted to reducing the strong dependencies between variables in Gibbs sampling, most noticeably, "overrelaxed" Gibbs sampling by Adler in 1981 [Adl81], and "ordered overrelaxation" Gibbs sampling by Radford in 1995 [Nea95]. But overrelaxed Gibbs sampling is applicable only to problems where all the full conditional distributions are Gaussian, which is very restrictive. Ordered overrelaxation is applicable to problems where the full conditional distributions are such that their cumulative distributions functions (CDFs) and inverse CDFs can be efficiently computed, which can be quite restrictive, too. In addition, many times the full conditional distributions are not available and thus Gibbs sampling is not feasible in the first place. Here we explore, within the more general framework of Metropolis algorithm, a sampling method termed “mirror-jump sampling”, to effectively reduce dependencies of successive states of a Markov chain.

2.2 Description of Mirror-Jump Sampling

All simulations were performed in a statistical analysis software called “R”.

Here we propose a strategy for MCMC acceleration for unimodal posteriors - “mirror-jump sampling”. It belongs to the category of Metropolis-Hastings algorithm, but its proposal distribution has a special nice feature. With the aim to reduce the positive serial correlations and even induce negative correlations, our proposal for the next move is based on not the current position, but instead the “mirror-point” of the current position. The “mirror-point” of a position is a loose term and can be any point on the other side of the mode. The hard-core of “mirror-jump” sampling is how to select the optimal “mirror-point” which in turn determines the form of the optimal proposal distribution.

Denote the K-dimensional parameter vector as $\theta = (\theta_1, ..., \theta_K)$. The “mirror-jump” sampling algorithm has the following three stages:

1. By means of Maximum Likelihood Estimations (MLEs) or preliminary MCMC output, we can have some crude estimates of the location and scale parameters of the posterior. We are going to take advantage of these information to improve subsequent simulations. Denote the rough estimates of posterior mode (or mean) as μ_0 (a K-d vector) and posterior covariance matrix as Σ_0 (a $K \times K$ matrix).

2. Given the current state θ_t, define the proposal distribution as:

$$q(\theta|\theta_t) = N(\theta_t^{(m)}, \Sigma)$$

$$\theta_t^{(m)} = \mu_0 + C \cdot (\mu_0 - \theta_t)$$

$$\Sigma = (DD^T) \otimes \Sigma_0$$

where: $\otimes$ denotes element-wise matrix multiplication, $\theta_t^{(m)}$ is the mirror-point of θ_t, and both C and D are adjustable vectors of coefficients of length K. Empirically the best range for C and D are $C_j \in [0, 1]$ and $D_j \in [0.5, 1.5]$, $j = 1, .., K$.

We then seek for optimal values of C and D as follows: we run trial MCMC simulations for grids of pairs of C and D values, compute their corresponding VIFs, and pick the C and D pair with the minimal VIF as the optimal choice of C and D.

3. The optimal proposal distribution is the one with the optimal C and D. We then carry out our MCMC simulations.

2.3 Design of Simulation Studies on Mirror-Jump Sampling

We compared the performances of three MCMC simulation schemes: mirror-jump sampling, random-walk sampling and IID sampling when available (for comparison purpose). Both mirror-jump sampling and random-walk sampling belong to the broad category of Metropolis-hastings algorithm. The difference between mirror-jump and random-walk lies in the forms of their proposal distributions. Mirror-jump is described in previous section. Random-walk has a proposal of this form: $q(\theta|\theta_t) = N(\theta_t, \Sigma)$, where $\Sigma = (DD^T) \otimes \Sigma_0$. So in random-walk the next move is relative to the current state. Only D is an adjustable vector of coefficients.

We did simulation studies on 8 different target distribution scenarios. They are all unimodal, but have different shape features representing essentially all possible shapes. For each scenario we first seek for the optimal C and D values for mirror-jump, and the optimal D value for random-walk. Then we carry out MCMC simulations using their optimal proposal distributions.

In order to carry out the comparisons, we selected those posteriors whose true mean and true standard deviation are known, denoted as μ, σ, respectively. To quantify the comparisons, we did repeated sampling, i.e., for each scenario we simulated M chains each of size N (after convergence). For all 8 examples studied here, $M = 100$. For the χ^2_5 example $N = 100000$, for the NB10 example $N = 50000$, and for all others $N = 10000$.

Let θ_{ij} denote the j-th iterate in the i-th chain. Let $\bar{\theta}_{i.}$, $\bar{s}_{i.}$, $\rho_i^{(1)}$, $\rho_i^{(2)}$, and VIF_i, with $i = 1, 2, ..., M$, be the (sample) mean, SD, ρ_1,ρ_2, and VIF, respectively. For each scenario we give a table of comparisons, with the meaning of each term defined in table 2.0. In addition, we also compare the acceptance rate r between mirror-jump sampling and random-walk sampling in each example.

For each scenario, we also compared posterior density plots of mirror-jump and random-walk sampling methods, and true posterior density if available (as dotted lines in figures).

quantity	estimated value	give or take
mean	mean[$\bar{\theta}_{i.}$]	SD[$\bar{\theta}_{i.}$]/$\sqrt{M}$
bias(mean)	mean[$\bar{\theta}_{i.} - \mu$]	SD[$\bar{\theta}_{i.} - \mu$]/$\sqrt{M}$
MSE(mean)	mean[$(\bar{\theta}_{i.} - \mu)^2$]	SD[$(\bar{\theta}_{i.} - \mu)^2$]/$\sqrt{M}$
SD	mean[$\bar{s}_{i.}$]	SD[$\bar{s}_{i.}$]/$\sqrt{M}$
bias(SD)	mean[$\bar{s}_{i.} - \sigma$]	SD[$\bar{s}_{i.} - \sigma$]/$\sqrt{M}$
MSE(SD)	mean[$(\bar{s}_{i.} - \sigma)^2$]	SD[$(\bar{s}_{i.} - \sigma)^2$]/$\sqrt{M}$
ρ_1	mean[$\rho_i^{(1)}$]	SD[$\rho_i^{(1)}$]/$\sqrt{M}$
ρ_2	mean[$\rho_i^{(2)}$]	SD[$\rho_i^{(2)}$]/$\sqrt{M}$
VIF	mean[VIF_i]	SD[VIF_i]/$\sqrt{M}$

Table 2.0: Definitions of terms used in comparison tables of simulation results

2.4 Results and Discussions

2.4.1 Univariate Case

(1). $N(0,1)$ target distribution

We started off with a simple toy example: the standard normal density. Simulation results are summarized in table 2.1 and figure 2.1.

- In mirror-jump sampling, VIF is as small as 0.1! This is about 40 times smaller than that in random-walk sampling (VIF=4.3), and 10 times smaller than that in iid sampling (VIF=1).

- Consistently with VIF, MSE(mean) in mirror-jump is 50 times smaller than that in random-walk, and 10 times smaller than that in iid.

- MSE(SD) in mirror-jump is comparable to that in random-walk, both twice bigger than that in iid. IID sampling (when available!) has the best precision in SD estimate, which is not surprising.

- The density plot of mirror-jump sampling is very close to the true density (dotted line), and is of much better quality than that of random-walk sampling.

- The acceptance rate is r=1 in mirror-jump sampling, 2.3 times higher than that (r=0.44) in random-walk sampling.

(2) a logit hierarchical model

• sampling distribution:

$$(y|p) \sim Bin(n=5,p),$$

• prior:

$$\theta = logit(p) = log(\frac{p}{1-p}) \sim N(\mu = 0, \sigma^2 = 0.5)$$

• posterior:

$$p(\theta|y) \propto \frac{e^{y\theta - \theta^2}}{(1 + e^{\theta})^n}$$

• posterior with y = 5:

$$p(\theta|y) \propto \frac{e^{5\theta - \theta^2}}{(1 + e^{\theta})^5}$$

Simulation results are summarized in table 2.2 and figure 2.2.

- In mirror-jump sampling, VIF is as small as 0.15! This is about 30 times smaller than that in random-walk sampling (VIF=4.4), and 6 times smaller than that in iid sampling (were it available).

- Consistently with VIF, MSE(mean) in mirror-jump is 32 times smaller than that in random-walk.

- MSE(SD) in mirror-jump is twice bigger than that in random-walk.

- The density plot of mirror-jump sampling is very close to the truth and is of much better quality than that of random-walk sampling.

- The acceptance rate is r=0.96 in mirror-jump sampling, 2.5 times higher than that (r=0.38) in random-walk sampling.

(3). $t_4(0,1)$ target distribution

Simulation results are summarized in table 2.3 and figure 2.3.

- In mirror-jump sampling, VIF=0.4. This is about 14 times smaller than that in random-walk sampling (VIF=5.7), and 2.5 times smaller than that in iid sampling (were it available).

- Consistently with VIF, MSE(mean) in mirror-jump is 15 times smaller than that in random-walk.

- MSE(SD) in mirror-jump is twice bigger than that in random-walk.

- The density plot of mirror-jump sampling is very close to the truth and is of much better quality than that of random-walk sampling.

- The acceptance rate is r=0.76 in mirror-jump sampling, 2.3 times higher than that (r=0.33) in random-walk sampling.

(4). log-Inv-χ^2 target distribution

$$x \sim Inv - \chi^2(\nu = 9, s^2 = 1)$$

$$\theta = log(x)$$

$$p(\theta|\nu, s^2) \propto e^{-\frac{\nu}{2}(\theta + s^2/e^\theta)}$$

Simulation results are summarized in table 2.4 and figure 2.4.

- In mirror-jump sampling, VIF=1. This is about 4.6 times smaller than that in random-walk sampling (VIF=4.6), and the same as that in iid sampling.

- Consistently with VIF, MSE(mean) in mirror-jump is 3.8 times smaller than that in random-walk.

- MSE(SD) in mirror-jump is 3 times smaller than that in random-walk.

- The density plot of mirror-jump sampling is very close to the truth and is of much better quality than that of random-walk sampling.

- The acceptance rate is r=0.82 in mirror-jump sampling, 2 times higher than that (r=0.42) in random-walk sampling.

(5). χ^2_5 target distribution

Simulation results are summarized in table 2.5 and figure 2.5.

- In mirror-jump sampling, VIF=2.2. This is 3 times smaller than that in random-walk sampling (VIF=6.7).

- Consistently with VIF, MSE(mean) in mirror-jump is 3 times smaller than that in random-walk.

- MSE(SD) in mirror-jump is twice smaller than that in random-walk.

- The density plot of mirror-jump sampling is of better quality than that of random-walk sampling.

- The acceptance rate is r=0.63 in mirror-jump sampling, 1.4 times higher than that (r=0.44) in random-walk sampling.

2.4.2 Bivariate Case

(6). bivariate normal target distribution

$$\theta = (\theta_1, \theta_2) \sim \mathrm{N}(\mu, \Sigma) \text{ where } \mu = (0,0) \text{ and } \Sigma = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

Since θ_1 and θ_2 are symmetric and have the same marginal distribution, so only the simulation results for θ_1 are presented in table 2.6 and figure 2.6.

- In mirror-jump sampling, VIF=0.12. This is 68 times smaller than that in random-walk sampling (VIF=8.18), and 8 times smaller than that in iid sampling (VIF=1).

- Consistently with VIF, MSE(mean) in mirror-jump is 100 times smaller than that in random-walk, and 10 times smaller than that in iid.

- MSE(SD) in mirror-jump is comparable to that in random-walk, both 6 times bigger than that in iid.

- The density plot of mirror-jump sampling is very close to the truth and is of much better quality than that of random-walk sampling.

- The acceptance rate is r=1 in mirror-jump sampling, 4.3 times higher than that (r=0.23) in random-walk sampling.

A comparison to univariate normal case:

Mirror-jump in bivariate case performs almost equally well as in univariate case, but random-walk performs much worse, evidenced by the inferior accuracy of mean estimates and the poor shape of the density plot.

(7). a normal hierarchical model

• sampling distribution:

$$y_i|\mu, \sigma^2 \sim N(\mu, \sigma^2)$$

• prior:

$$p(\mu, \sigma^2) \propto \frac{1}{\sigma^2}$$

• It can be shown that the marginal posteriors are:

$$\mu|y \sim t_{n-1}(\bar{y}, s^2/n)$$

$$\sigma^2|y \sim Inv - \chi^2(n-1, s^2)$$

where $\bar{y}$ and s^2 are sampling mean and variance, respectively.

• variable transformation:

$$\theta = (\theta_1, \theta_2), \qquad \theta_1 = \mu, \qquad \theta_2 = log(\sigma^2)$$

• joint posterior of θ:

$$p(\theta_1, \theta_2|y) \propto e^{-\frac{n}{2}\theta_2 - \frac{1}{2e^{\theta_2}}(n(\theta_1-\bar{y})^2+(n-1)s^2)}$$

• in this study: $n = 10,\ \bar{y} = -0.4325064,\ s^2 = 1.038623$

Simulation results for θ_1 are summarized in table 2.7 and figure 2.7.

- In mirror-jump sampling, VIF=2. This is 4.5 times smaller than that in random-walk sampling (VIF=9).

- Consistently with VIF, MSE(mean) in mirror-jump is 2.3 times smaller than that in random-walk.

- MSE(SD) in mirror-jump is 4 times bigger than that in random-walk.

- The density plot of mirror-jump sampling is very close to the truth and is of much better quality than that of random-walk sampling.

Simulation results for θ_2 are summarized in table 2.8 and figure 2.8.

- In mirror-jump sampling, VIF=4. This is 2.6 times smaller than that in random-walk sampling (VIF=10.5).

- Consistently with VIF, MSE(mean) in mirror-jump is 1.7 times smaller than that in random-walk.

- MSE(SD) in mirror-jump is comparable to that in random-walk, both 8 times bigger than that in iid.

- The density plot of mirror-jump sampling is very close to the truth and is of much better quality than that of random-walk sampling.

The acceptance rate is r=0.74 in mirror-jump sampling, 3.7 times higher than that (r=0.2) in random-walk sampling.

2.4.3 Trivariate Case

(8). NB10 example [Dra97]

NB10 is a type of metal block whose weight is supposed to be 10g.

• The actual weights of $n = 100$ NB10's are $y_i, i = 1, ..., 100$, as listed in table 2.12.

• sampling distribution:

$$y_i|\mu, \sigma, \nu \sim t_\nu(\mu, \sigma^2)$$

• priors:

$$\mu \sim N(\mu_0 = 0, \sigma_0^2 = 10^6)$$

$$\frac{1}{\sigma^2} \sim Gamma(\alpha = 10^{-3}, \beta = 10^{-3})$$

$$\nu \sim Unif(a = 2, b = 12)$$

• joint posterior:

$$p(\mu, \sigma, \nu | y) \propto \prod_{i=1}^{n} \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\nu\pi}\sigma}(1 + \frac{1}{\nu}(\frac{y_i - \mu}{\sigma})^2)^{-\frac{\nu+1}{2}} \cdot e^{-\frac{1}{2}(\frac{\mu-\mu_0}{\sigma_0})^2} \cdot \sigma^{-2\alpha-1}e^{-\frac{\beta}{\sigma^2}}$$

• let $\theta = (\theta_1, \theta_2, \theta_3), \quad \theta_1 = \mu, \quad \theta_2 = \sigma, \quad \theta_3 = \nu$

Simulation results for θ_1 are summarized in table 2.9 and figure 2.9.

- In mirror-jump sampling, VIF is 2.3. This is about 6.8 times smaller than that in random-walk sampling (VIF=15.7).

- Consistently with VIF, MSE(mean) in mirror-jump is 5.3 times smaller than that in random-walk.

- MSE(SD) in mirror-jump is comparable to that in random-walk.

- The density plot of mirror-jump sampling is of much better quality than that of random-walk sampling.

Simulation results for θ_2 are summarized in table 2.10 and figure 2.10.

- In mirror-jump sampling, VIF is 4.5. This is about 3.8 times smaller than that in random-walk sampling (VIF=17).

- Consistently with VIF, MSE(mean) in mirror-jump is 2.7 times smaller than that in random-walk.

- MSE(SD) in mirror-jump is comparable to that in random-walk.

- The density plot of mirror-jump sampling is of much better quality than that of random-walk sampling.

Simulation results for θ_3 are summarized in table 2.11 and figure 2.11.

- In mirror-jump sampling, VIF is 7.7. This is about 3 times smaller than that in random-walk sampling (VIF=23.9).
- Consistently with VIF, MSE(mean) in mirror-jump is 1.7 times smaller than that in random-walk.
- MSE(SD) in mirror-jump is comparable to that in random-walk.
- The density plot of mirror-jump sampling is of much better quality than that of random-walk sampling.

The acceptance rate is r=0.48 in mirror-jump sampling, 4 times higher than that (r=0.12) in random-walk sampling.

2.5 Summary and Conclusion

Of the 8 examples that we studied, 5 are univariate distributions, 2 are bivariate distributions, and 1 is trivariate distribution. Of the 11 marginal distributions in these examples, 8 are symmetric or roughly symmetric, 3 are lightly-skewed, and 2 are heavily-skewed. Also some have thinner tails while others have fatter tails. So we consider our examples to be representative of unimodal distributions.

In all these examples, mirror-jump sampling achieves much higher acceptance rates, yields much smaller VIFs and MSEs, and produces much better posterior density plots, than random-walk sampling. In other words, mirror-jump sampling is more effective in learning about posteriors than random-walk sampling, and thus can serve as a strategy for MCMC acceleration for unimodal posteriors. The highest improvement by mirror-jump is achieved when the target is roughly symmetric, but even when the target is heavily skewed, mirror-jump clearly outperforms random-walk.

An important factor in mirror-jump sampling is to choose suitable values of C and D, the adjustable vectors of coefficients. Based on our simulation studies, an empirical pattern of optimal values can be summarized as follows:

optimal value	symmetric target	skewed target
C	0.8-1	0.1-0.3
D	0.5-0.6	1.2-1.5

The above empirical pattern can serve as a general guide for routine use, and finding precisely optimal values might not be crucial. But if one wants to find the optimal value for a specific target, he can do the optimization on grids of values in trial runs.

In conclusion, mirror-jump sampling successfully enhances the accuracy of MCMC posterior estimates by reducing the positive autocorrelations of the Markov chain time series, thus is an effective strategy for MCMC acceleration in unimodal posterior settings.

2.6 Tables and Figures

MCMC approach	quantity	estimated value	give or take
iid	mean	8.554078e-04	1.054851e-03
	bias(mean)	8.554078e-04	1.054851e-03
	MSE(mean)	1.108901e-04	1.505994e-05
	SD	1.000218e+00	7.088864e-04
	bias(SD)	2.176655e-04	7.088864e-04
	MSE(SD)	4.979685e-05	7.961896e-06
	ρ_1	8.612204e-04	1.027168e-03
	ρ_2	-1.187528e-03	1.020740e-03
	VIF	1.000228e+00	1.093808e-03
random-walk D=2.4 r=0.44	mean	0.0014325209	2.293341e-03
	bias(mean)	0.0014325209	2.293341e-03
	MSE(mean)	0.0005227342	7.477420e-05
	SD	1.0005437915	1.497277e-03
	bias(SD)	0.0005437915	1.497277e-03
	MSE(SD)	0.0002222376	3.320356e-05
	ρ_1	0.6277240861	1.252537e-03
	ρ_2	0.0017124024	1.142851e-03
	VIF	4.3067727570	2.941036e-02
mirror-jump C=0.8 D=0.6 r=1	mean	4.323198e-04	3.195804e-04
	bias(mean)	4.323198e-04	3.195804e-04
	MSE(mean)	1.029793e-05	1.295362e-06
	SD	9.999628e-01	1.563248e-03
	bias(SD)	-3.719190e-05	1.563248e-03
	MSE(SD)	2.419320e-04	3.216148e-05
	ρ_1	-7.996474e-01	6.196107e-04
	ρ_2	-1.270918e-03	1.000725e-03
	VIF	1.115392e-01	4.184727e-03

Table 2.1: Comparisons of simulation results in $N(0,1)$ example

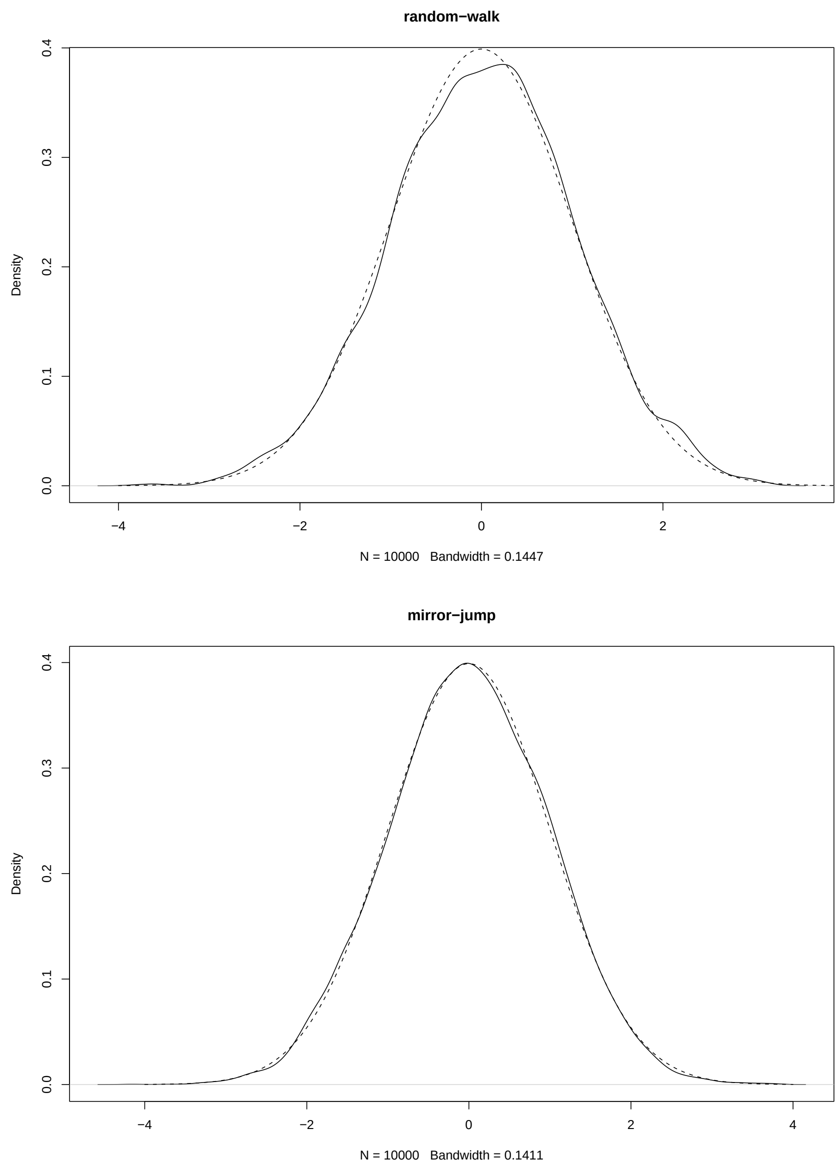

Figure 2.1: Density plots of simulations in $N(0, 1)$ example

MCMC approach	quantity	estimated value	give or take
random-walk D=2.9 r=0.38	mean	8.045333e-01	1.290523e-03
	bias(mean)	1.062937e-03	1.290523e-03
	MSE(mean)	1.660095e-04	2.253454e-05
	SD	5.773350e-01	8.831705e-04
	bias(SD)	1.352739e-03	8.831705e-04
	MSE(SD)	7.904893e-05	1.117144e-05
	ρ_1	6.346572e-01	1.177947e-03
	ρ_2	3.244813e-03	1.215001e-03
	VIF	4.414545e+00	2.610780e-02
mirror-jump C=0.9 D=0.5 r=0.96	mean	8.033672e-01	2.284637e-04
	bias(mean)	-1.030931e-04	2.284637e-04
	MSE(mean)	5.177999e-06	1.218180e-06
	SD	5.747720e-01	1.229427e-03
	bias(SD)	-1.210217e-03	1.229427e-03
	MSE(SD)	1.511022e-04	1.852850e-05
	ρ_1	-7.686650e-01	1.020488e-03
	ρ_2	1.281027e-02	1.729139e-03
	VIF	1.484356e-01	4.431210e-03

Table 2.2: Comparisons of simulation results in logit hierarchical example

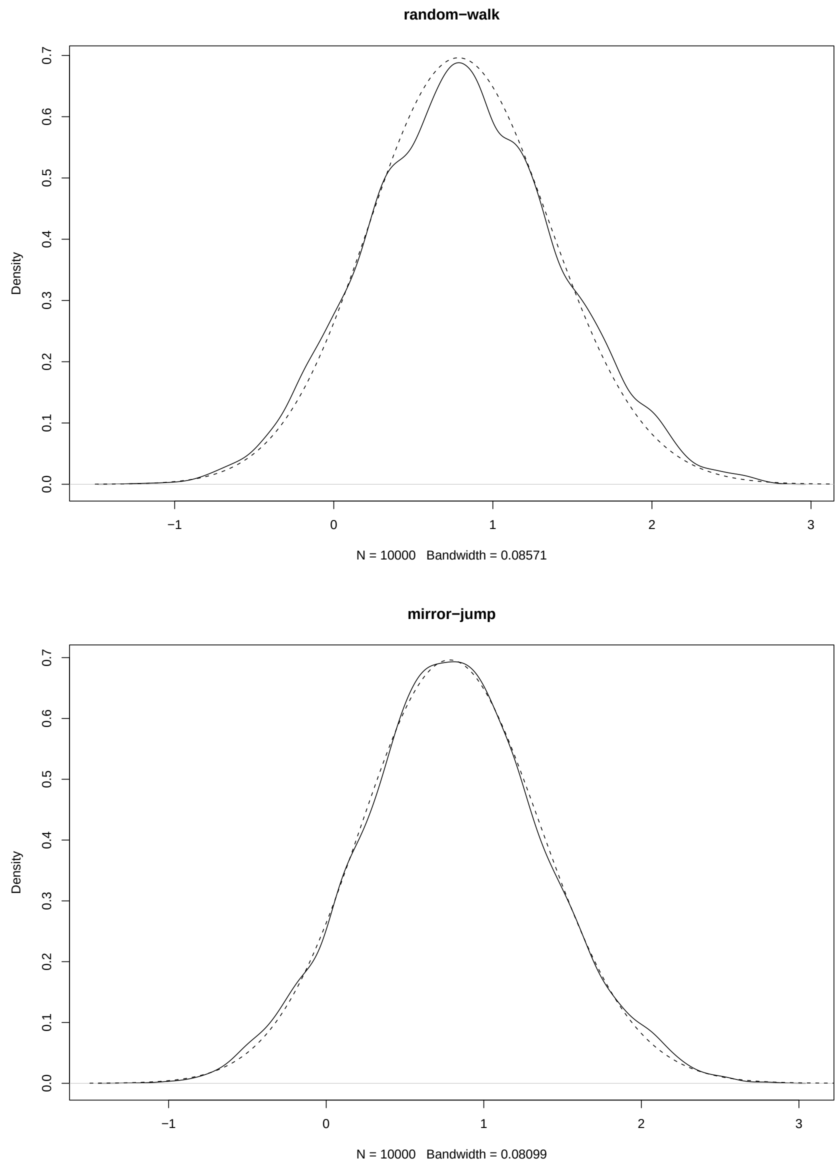

Figure 2.2: Density plots of simulations in logit hierarchical example

MCMC approach	quantity	estimated value	give or take
iid	mean	3.965573e-05	1.339549e-03
	bias(mean)	3.965573e-05	1.339549e-03
	MSE(mean)	1.776464e-04	2.153971e-05
	SD	1.416123e+00	6.060630e-03
	bias(SD)	1.909657e-03	6.060630e-03
	MSE(SD)	3.640039e-03	3.014827e-03
	ρ_1	-7.763759e-04	9.831798e-04
	ρ_2	1.149700e-04	8.762795e-04
	VIF	9.989622e-01	1.319235e-03
random-walk D=2.9 r=0.33	mean	-0.0002299677	0.0033663024
	bias(mean)	-0.0002299677	0.0033663024
	MSE(mean)	0.0011219201	0.0001961808
	SD	1.4087884549	0.0099283679
	bias(SD)	-0.0054251075	0.0099283679
	MSE(SD)	0.0097881082	0.0041073723
	ρ_1	0.6811269436	0.0032609569
	ρ_2	0.0097749450	0.0018411005
	VIF	5.6716204092	0.1802417027
mirror-jump C=1.0 D=0.6 r=0.76	mean	-0.0002130267	8.559561e-04
	bias(mean)	-0.0002130267	8.559561e-04
	MSE(mean)	0.0000725788	1.101928e-05
	SD	1.3967631964	1.369421e-02
	bias(SD)	-0.0174503660	1.369421e-02
	MSE(SD)	0.0188701108	4.604951e-03
	ρ_1	-0.4555809153	4.541883e-03
	ρ_2	0.0187301856	2.080024e-03
	VIF	0.3854887801	5.800125e-03

Table 2.3: Comparisons of simulation results in $t_4(0,1)$ example

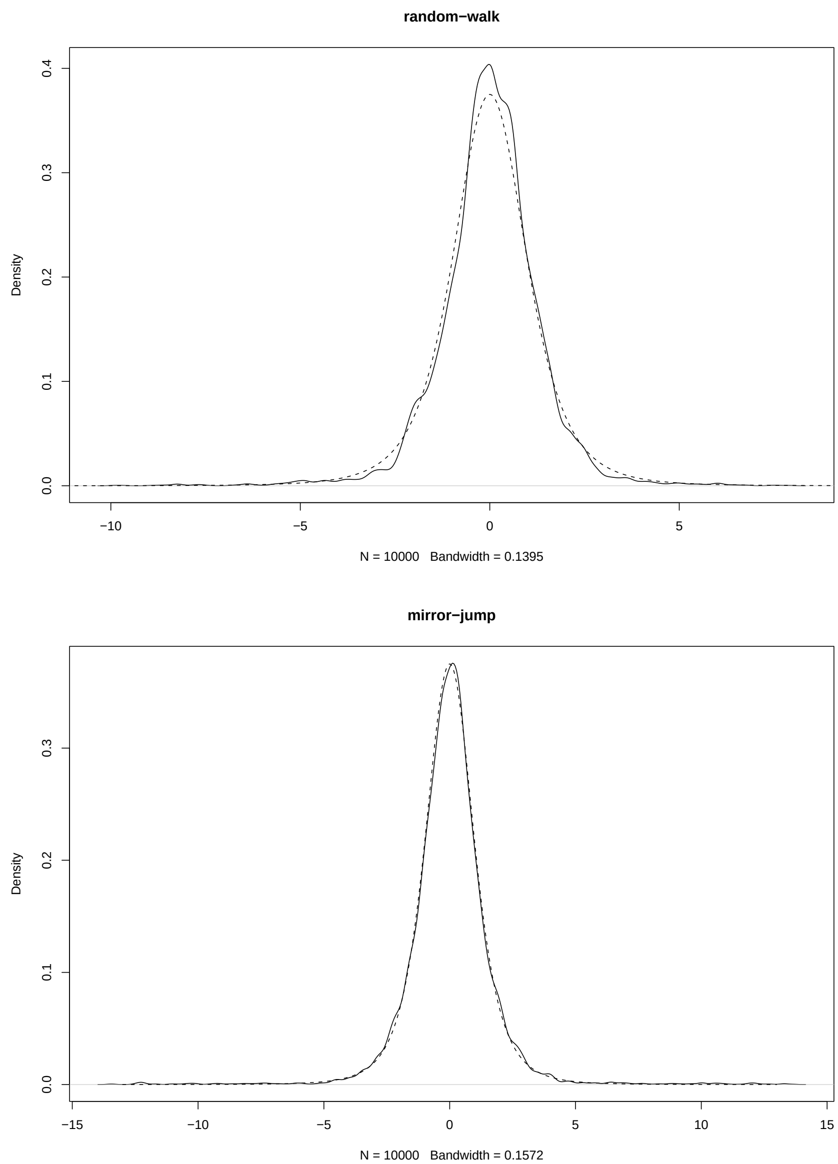

Figure 2.3: Density plots of simulations in $t_4(0, 1)$ example

MCMC approach	quantity	estimated value	give or take
iid	mean	1.147928e-01	5.522820e-04
	bias(mean)	-4.136916e-04	5.522820e-04
	MSE(mean)	3.036767e-05	3.961238e-06
	SD	4.984012e-01	4.241090e-04
	bias(SD)	-3.222230e-04	4.241090e-04
	MSE(SD)	1.791081e-05	2.258526e-06
	ρ_1	6.458457e-04	1.083683e-03
	ρ_2	-2.916296e-04	1.045298e-03
	VIF	9.998732e-01	1.374067e-03
random-walk D=2.5 r=0.42	mean	1.153939e-01	1.123602e-03
	bias(mean)	1.873827e-04	1.123602e-03
	MSE(mean)	1.250207e-04	1.382176e-05
	SD	4.990709e-01	9.892717e-04
	bias(SD)	3.474457e-04	9.892717e-04
	MSE(SD)	9.700791e-05	1.204232e-05
	ρ_1	6.445532e-01	1.318092e-03
	ρ_2	3.067704e-03	1.008426e-03
	VIF	4.569989e+00	3.077507e-02
mirror-jump C=0.3 D=1.2 r=0.82	mean	1.159082e-01	5.700396e-04
	bias(mean)	7.017524e-04	5.700396e-04
	MSE(mean)	3.266203e-05	4.771274e-06
	SD	4.989691e-01	5.705290e-04
	bias(SD)	2.456336e-04	5.705290e-04
	MSE(SD)	3.228516e-05	4.464146e-06
	ρ_1	6.231940e-03	1.763539e-03
	ρ_2	2.129702e-02	1.536596e-03
	VIF	1.042110e+00	1.952252e-02

Table 2.4: Comparisons of simulation results in log-Inv-χ^2 example

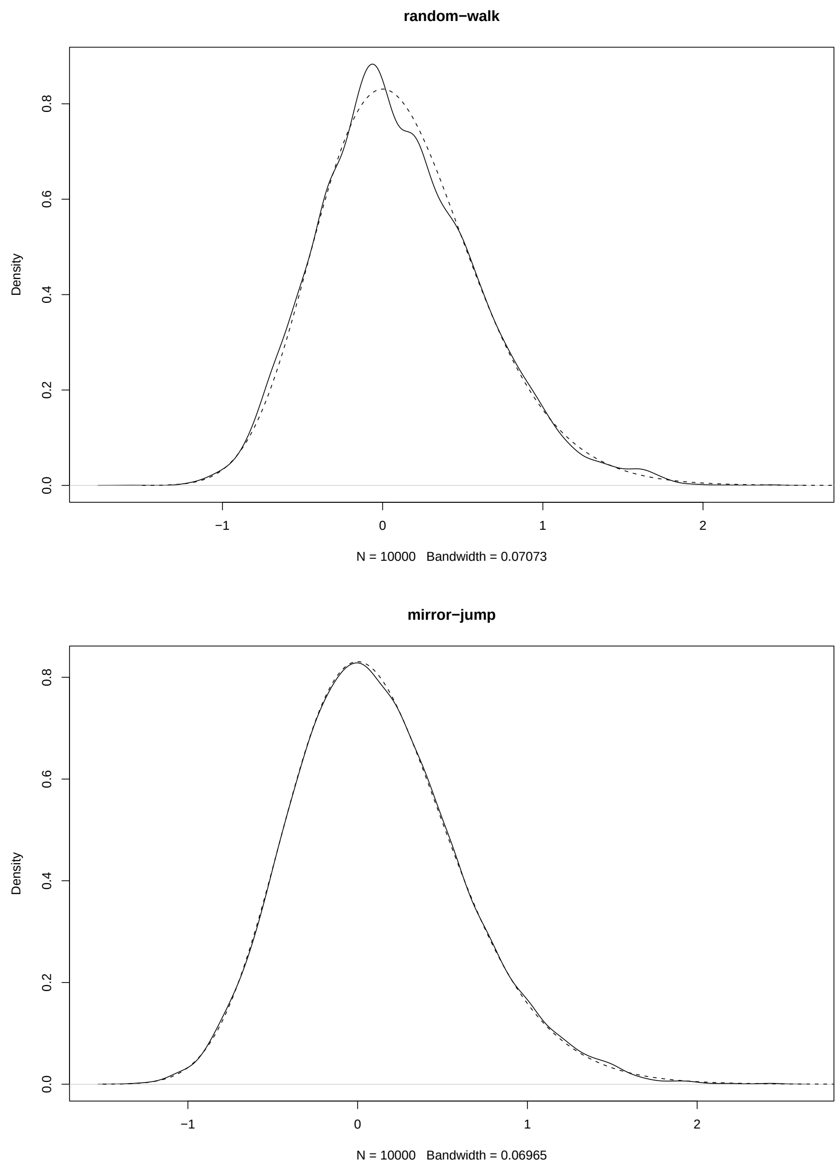

Figure 2.4: Density plots of simulations in log-Inv-χ^2 example

MCMC approach	quantity	estimated value	give or take
iid	mean	4.999948e+00	9.642185e-04
	bias(mean)	-5.234470e-05	9.642185e-04
	MSE(mean)	9.204475e-05	1.336530e-05
	SD	3.162806e+00	1.039038e-03
	bias(SD)	5.286118e-04	1.039038e-03
	MSE(SD)	1.071598e-04	1.495109e-05
	ρ_1	2.225566e-04	2.865213e-04
	ρ_2	-2.195448e-04	3.093434e-04
	VIF	1.000149e+00	1.489555e-04
random-walk D=2.0 r=0.44	mean	5.0038763245	0.0025211170
	bias(mean)	0.0038763245	0.0025211170
	MSE(mean)	0.0006442730	0.0001026431
	SD	3.1647312372	0.0029887489
	bias(SD)	0.0024535771	0.0029887489
	MSE(SD)	0.0008903494	0.0001270227
	ρ_1	0.7281834137	0.0004279591
	ρ_2	0.0090306537	0.0003707872
	VIF	6.6863594641	0.0280472723
mirror-jump C=0.3 D=1.4 r=0.63	mean	5.0027338392	1.494960e-03
	bias(mean)	0.0027338392	1.494960e-03
	MSE(mean)	0.0002287294	3.156601e-05
	SD	3.1655338967	2.140169e-03
	bias(SD)	0.0032562366	2.140169e-03
	MSE(SD)	0.0004640552	6.709491e-05
	ρ_1	0.3174548070	7.575200e-04
	ρ_2	0.0242047243	5.955489e-04
	VIF	2.2170131828	4.985935e-02

Table 2.5: Comparisons of simulation results in χ_5^2 example

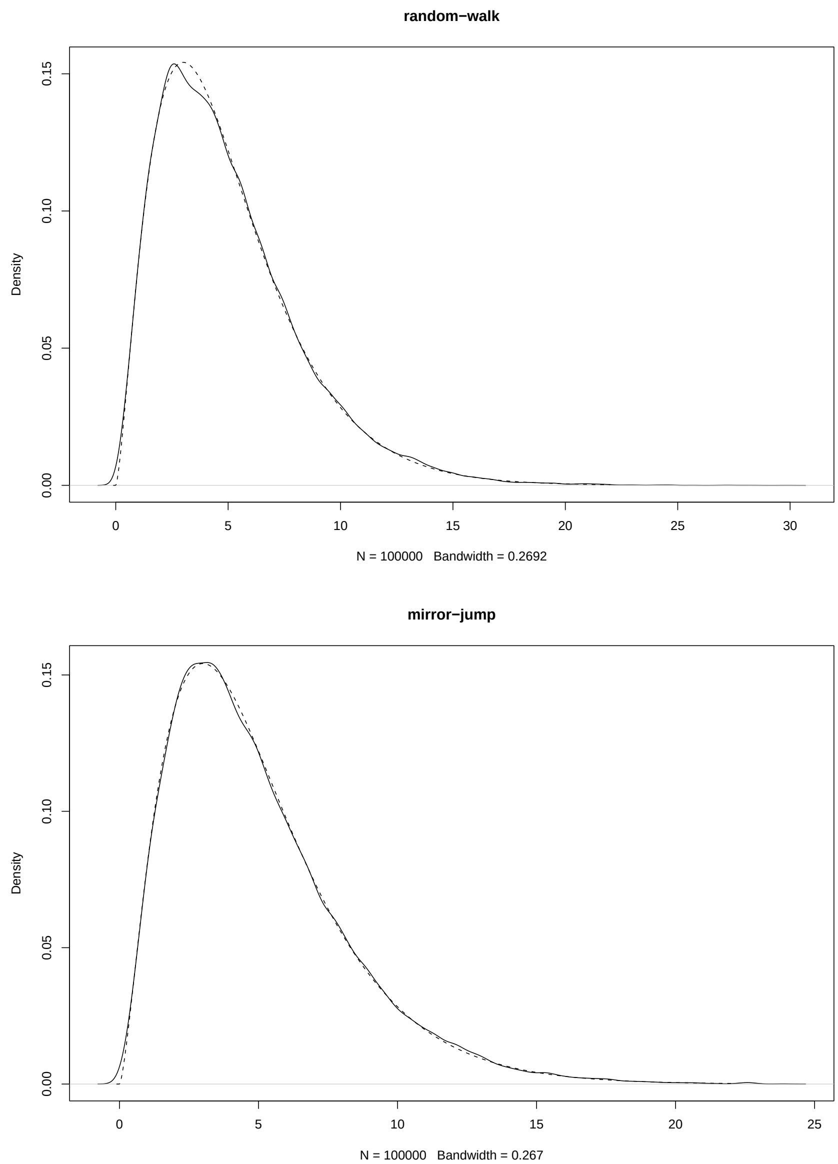

Figure 2.5: Density plots of simulations in χ^2_5 example

MCMC approach	quantity	estimated value	give or take
iid θ_1	mean	4.462195e-04	9.634991e-04
	bias(mean)	4.462195e-04	9.634991e-04
	MSE(mean)	9.210384e-05	1.155279e-05
	SD	9.992275e-01	6.531846e-04
	bias(SD)	-7.725333e-04	6.531846e-04
	MSE(SD)	4.283517e-05	5.965962e-06
	ρ_1	-4.030041e-04	1.076115e-03
	ρ_2	1.136734e-03	9.954501e-04
	VIF	9.996055e-01	1.082528e-03
random-walk θ_1 D=(2.4,2.4) r=0.23	mean	-0.0009113475	2.894526e-03
	bias(mean)	-0.0009113475	2.894526e-03
	MSE(mean)	0.0008302804	1.238292e-04
	SD	1.0024130674	1.837733e-03
	bias(SD)	0.0024130674	1.837733e-03
	MSE(SD)	0.0003401717	4.986272e-05
	ρ_1	0.7852362522	8.751035e-04
	ρ_2	0.0006861273	1.179157e-03
	VIF	8.1794693591	6.006317e-02
mirror-jump θ_1 C=(0.8,0.8) D=(0.6,0.6) r=1	mean	-2.503184e-04	2.883617e-04
	bias(mean)	-2.503184e-04	2.883617e-04
	MSE(mean)	8.294751e-06	1.307423e-06
	SD	1.001694e+00	1.651303e-03
	bias(SD)	1.693573e-03	1.651303e-03
	MSE(SD)	2.728216e-04	4.394595e-05
	ρ_1	-8.004676e-01	6.211636e-04
	ρ_2	1.136164e-03	1.181965e-03
	VIF	1.204849e-01	4.219793e-03

Table 2.6: Comparisons of simulation results for θ_1 in bivariate normal example

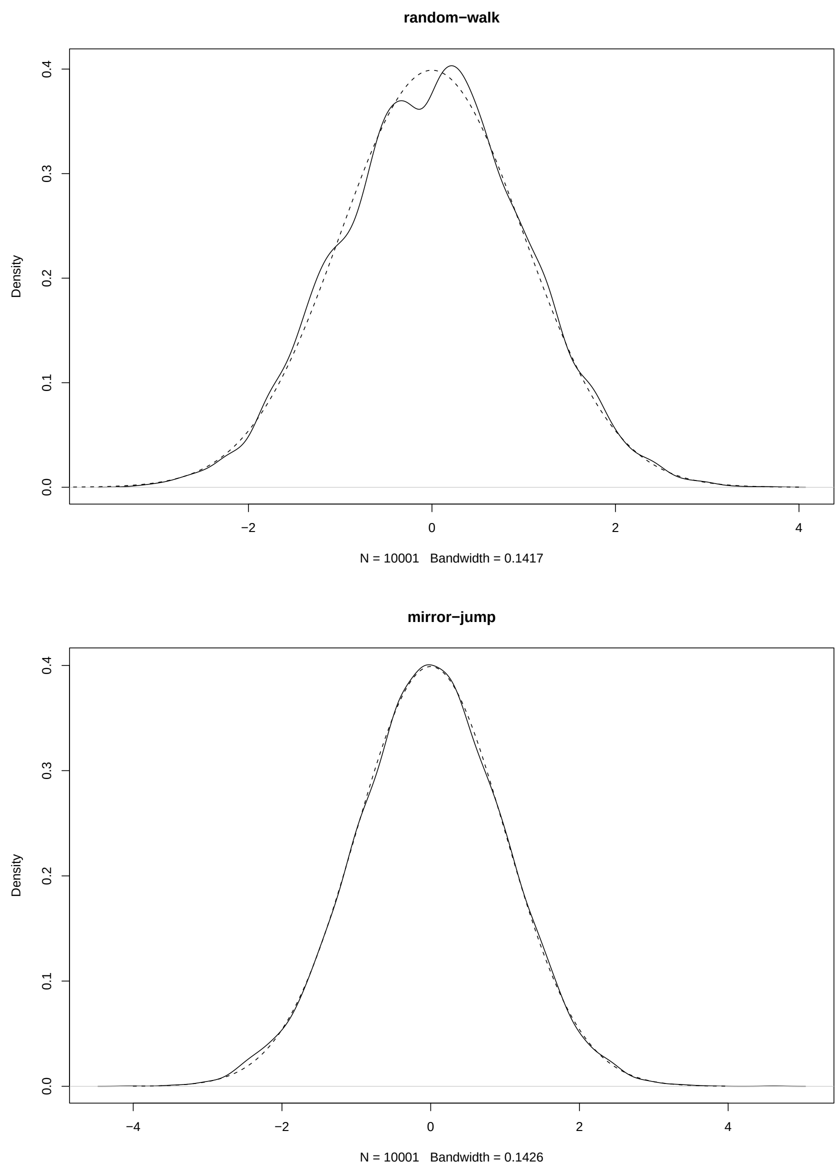

Figure 2.6: Density plots of simulations for θ_1 in bivariate normal example

MCMC approach	quantity	estimated value	give or take
iid θ_1	mean	-4.320503e-01	3.736863e-04
	bias(mean)	4.561211e-04	3.736863e-04
	MSE(mean)	1.403255e-05	2.108264e-06
	SD	3.655657e-01	3.136801e-04
	bias(SD)	1.382466e-04	3.136801e-04
	MSE(SD)	9.760240e-06	1.250467e-06
	ρ_1	-1.331796e-03	1.049818e-03
	ρ_2	-1.520480e-04	9.646113e-04
	VIF	1.000390e+00	1.263876e-03
random-walk θ_1 D=(2.5,2.5) r=0.2	mean	-0.4316500055	1.045987e-03
	bias(mean)	0.0008563945	1.045987e-03
	MSE(mean)	0.0001090482	1.289781e-05
	SD	0.3641175067	1.003741e-03
	bias(SD)	-0.0013099730	1.003741e-03
	MSE(SD)	0.0001014582	1.360322e-05
	ρ_1	0.8051734848	9.899560e-04
	ρ_2	0.0012193667	1.323203e-03
	VIF	9.1464056199	8.526489e-02
mirror-jump θ_1 C=(0.9,0.3) D=(0.5,1.2) r=0.74	mean	-4.323790e-01	6.945218e-04
	bias(mean)	1.274125e-04	6.945218e-04
	MSE(mean)	4.776993e-05	1.648946e-05
	SD	3.634888e-01	2.024940e-03
	bias(SD)	-1.938721e-03	2.024940e-03
	MSE(SD)	4.096963e-04	1.377912e-04
	ρ_1	-2.318094e-01	8.941884e-03
	ρ_2	1.510417e-01	7.489618e-03
	VIF	2.036863e+00	4.557477e-01

Table 2.7: Comparisons of simulation results for θ_1 in normal hierarchical example

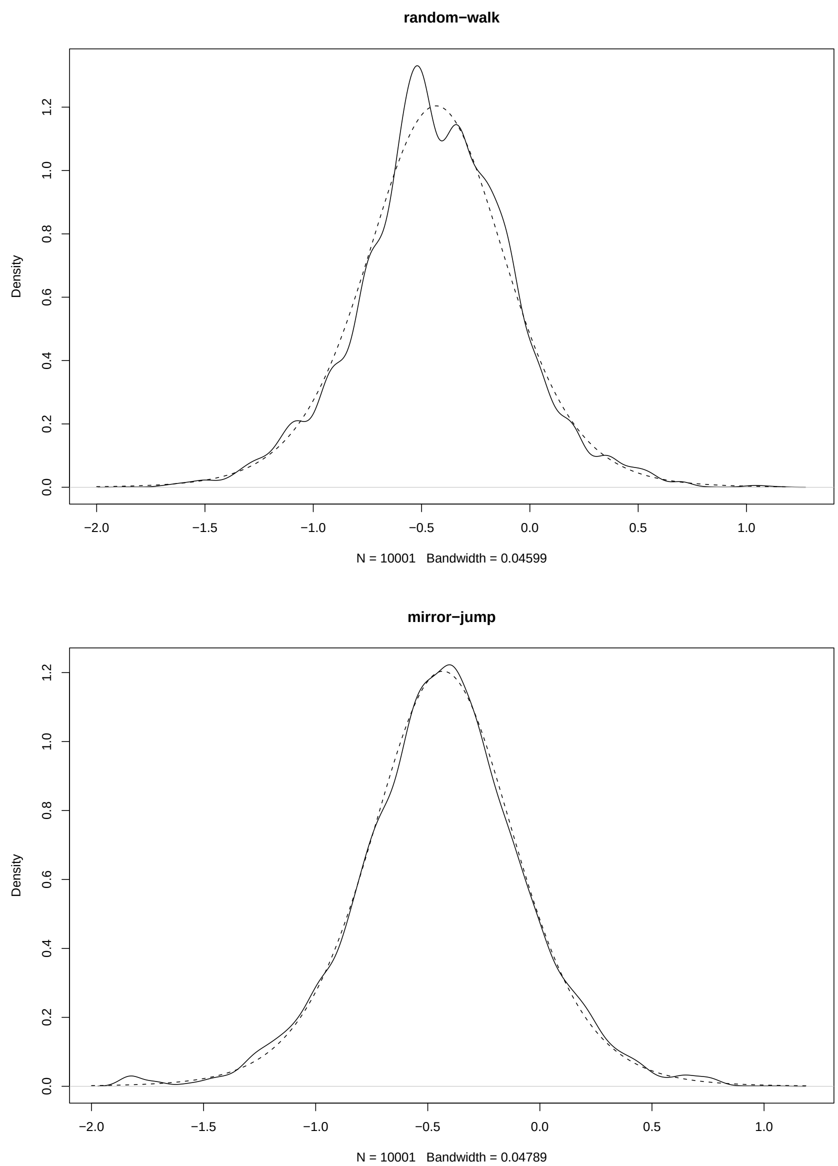

Figure 2.7: Density plots of simulations for θ_1 in normal hierarchical example

MCMC approach	quantity	estimated value	give or take
iid θ_2	mean	1.532318e-01	5.438054e-04
	bias(mean)	1.295117e-04	5.438054e-04
	MSE(mean)	2.929348e-05	3.518265e-06
	SD	4.985235e-01	4.391422e-04
	bias(SD)	-1.999919e-04	4.391422e-04
	MSE(SD)	1.913174e-05	2.409296e-06
	ρ_1	-7.785184e-04	8.569851e-04
	ρ_2	-9.864591e-04	1.140988e-03
	VIF	1.000414e+00	4.136749e-04
random-walk θ_2 D=(2.5,2.5) r=0.2	mean	0.1490482933	1.421540e-03
	bias(mean)	-0.0040539694	1.421540e-03
	MSE(mean)	0.0002164914	2.894667e-05
	SD	0.4967360883	1.219100e-03
	bias(SD)	-0.0019873786	1.219100e-03
	MSE(SD)	0.0001510839	2.069170e-05
	ρ_1	0.8195737143	8.655034e-04
	ρ_2	0.0082670088	1.220458e-03
	VIF	10.5066503504	9.508587e-02
mirror-jump θ_2 C=(0.9,0.3) D=(0.5,1.2) r=0.74	mean	0.1525447498	1.125050e-03
	bias(mean)	-0.0005575128	1.125050e-03
	MSE(mean)	0.0001256188	2.925314e-05
	SD	0.4968737911	1.241818e-03
	bias(SD)	-0.0018496758	1.241818e-03
	MSE(SD)	0.0001560903	4.700829e-05
	ρ_1	0.1375195029	3.967851e-03
	ρ_2	0.0996332979	3.025819e-03
	VIF	4.0399885594	7.396661e-01

Table 2.8: Comparisons of simulation results for θ_2 in normal hierarchical example

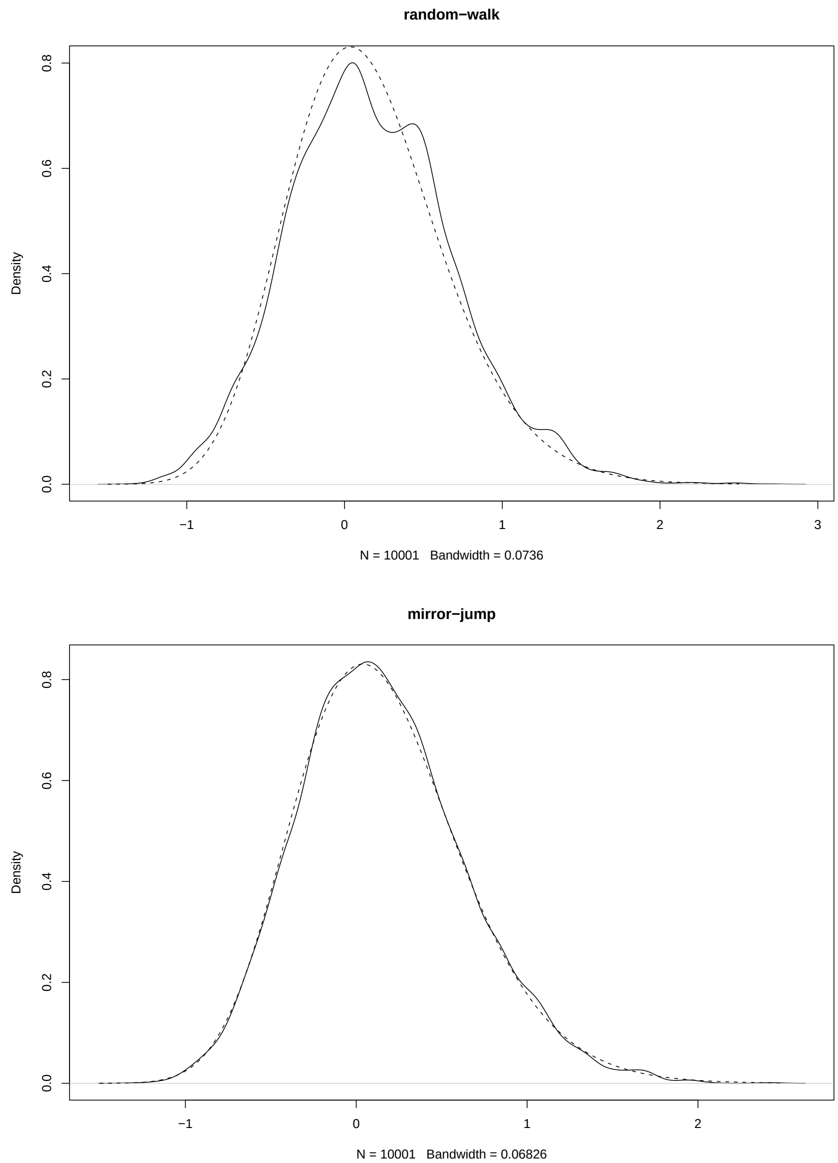

Figure 2.8: Density plots of simulations for θ_2 in normal hierarchical example

MCMC approach	quantity	estimated value	give or take
random-walk θ_1 D=(2.5,2.5,1.9) r=0.12	mean	4.042951e+02	8.461925e-04
	bias(mean)	-2.237356e-03	8.461925e-04
	MSE(mean)	7.589390e-05	1.058318e-05
	SD	4.705677e-01	5.925134e-04
	bias(SD)	2.445208e-04	5.925134e-04
	MSE(SD)	3.481593e-05	4.499988e-06
	ρ_1	8.798517e-01	2.858465e-04
	ρ_2	6.441885e-03	6.443480e-04
	VIF	1.565537e+01	3.940309e-02
mirror-jump θ_1 C=(0.85,0.85,0.1) D=(0.6,0.6,1.5) r=0.48	mean	4.042958e+02	3.466374e-04
	bias(mean)	-1.524960e-03	3.466374e-04
	MSE(mean)	1.422110e-05	2.119174e-06
	SD	4.718563e-01	5.770738e-04
	bias(SD)	1.533100e-03	5.770738e-04
	MSE(SD)	3.531880e-05	5.974636e-06
	ρ_1	9.287808e-02	1.407557e-03
	ρ_2	7.544614e-02	1.130357e-03
	VIF	2.293383e+00	5.089140e-01

Table 2.9: Comparisons of simulation results for θ_1 in NB10 example

MCMC approach	quantity	estimated value	give or take
random-walk θ_2 D=(2.5,2.5,1.9) r=0.12	mean	3.846994e+00	8.503532e-04
	bias(mean)	-1.626056e-03	8.503532e-04
	MSE(mean)	7.423101e-05	1.023882e-05
	SD	4.453373e-01	6.304163e-04
	bias(SD)	1.219363e-02	6.304163e-04
	MSE(SD)	1.880297e-04	1.621968e-05
	ρ_1	8.890258e-01	2.546241e-04
	ρ_2	1.082462e-02	6.515734e-04
	VIF	1.703165e+01	4.161197e-02
mirror-jump θ_2 C=(0.85,0.85,0.1) D=(0.6,0.6,1.5) r=0.48	mean	3.846000e+00	4.623756e-04
	bias(mean)	-2.619909e-03	4.623756e-04
	MSE(mean)	2.802925e-05	4.343847e-06
	SD	4.453822e-01	8.080820e-04
	bias(SD)	1.223858e-02	8.080820e-04
	MSE(SD)	2.144294e-04	3.027864e-05
	ρ_1	2.598496e-01	2.117838e-03
	ρ_2	1.051381e-01	1.475137e-03
	VIF	4.487855e+00	6.656248e-01

Table 2.10: Comparisons of simulation results for θ_2 in NB10 example

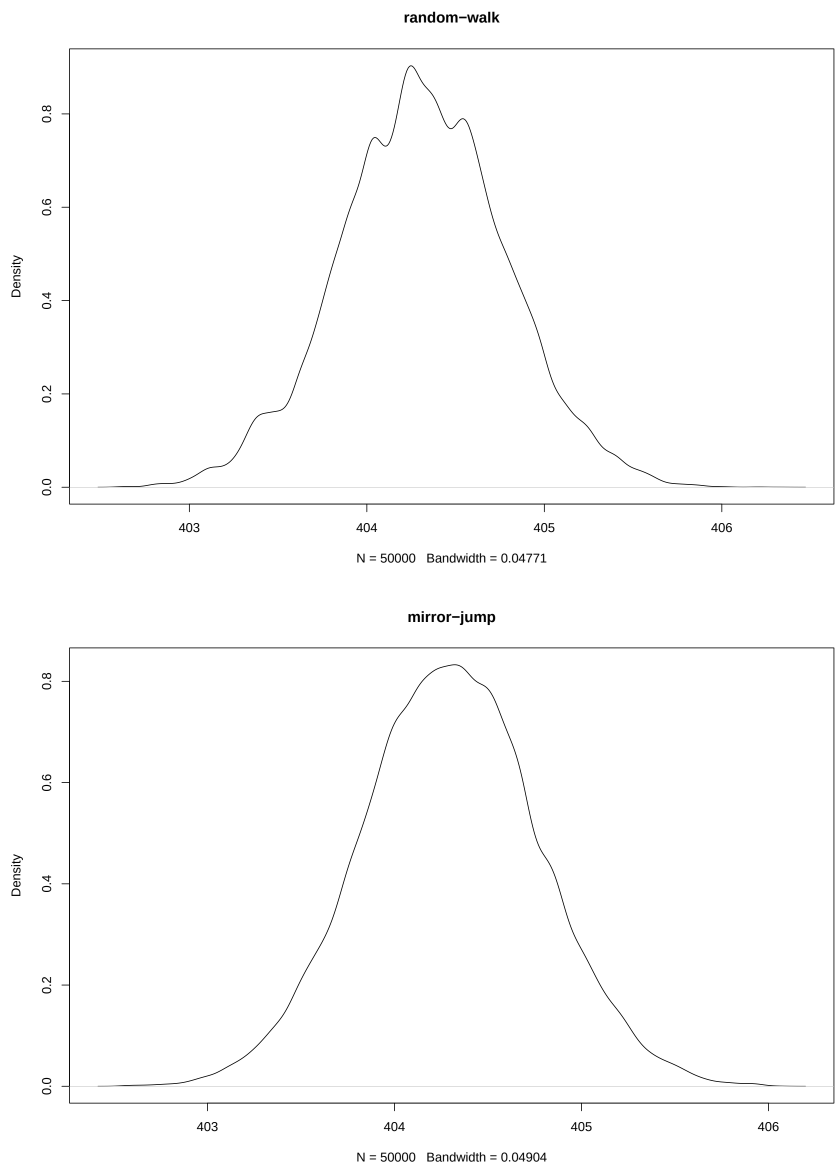

Figure 2.9: Density plots of simulations for θ_1 in NB10 example

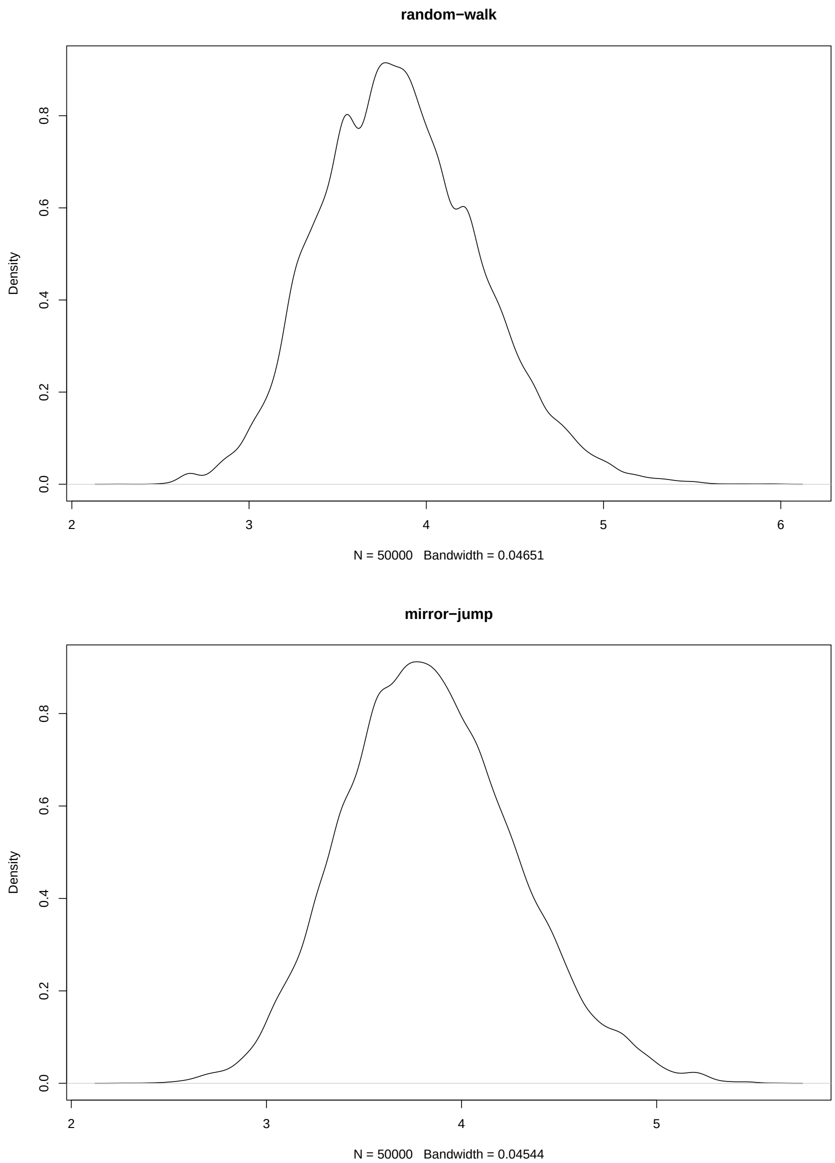

Figure 2.10: Density plots of simulations for θ_2 in NB10 example

MCMC approach	quantity	estimated value	give or take
random-walk θ_3 C=(2.5,2.5,1.9) r=0.12	mean	3.582675096	0.0029942593
	bias(mean)	-0.008020276	0.0029942593
	MSE(mean)	0.000951918	0.0001071656
	SD	1.228234392	0.0046995313
	bias(SD)	0.065272993	0.0046995313
	MSE(SD)	0.006447037	0.0007932370
	ρ_1	0.919436622	0.0004913180
	ρ_2	0.008107402	0.0005954981
	VIF	23.920226644	0.1578924199
mirror-jump θ_3 C=(0.85,0.85,0.1) D=(0.6,0.6,1.5) r=0.48	mean	3.5723597680	1.506736e-03
	bias(mean)	-0.0183356035	1.506736e-03
	MSE(mean)	0.0005609495	5.526925e-05
	SD	1.1978611060	3.239205e-03
	bias(SD)	0.0348997072	3.239205e-03
	MSE(SD)	0.0022567421	3.795677e-04
	ρ_1	0.4449081683	2.772287e-03
	ρ_2	0.0687624450	1.910930e-03
	VIF	7.7185631568	1.004869e+00

Table 2.11: Comparisons of simulation results for θ_3 in NB10 example

Value	375	392	393	397	398	399	400	401
Frequency	1	1	1	1	2	7	4	12
Value	402	403	404	405	406	407	408	409
Frequency	8	6	9	5	12	8	5	5
Value	410	411	412	413	415	418	423	437
Frequency	4	1	3	1	1	1	1	1

Table 2.12: Frequency distribution of NB10 weights [Dra97]

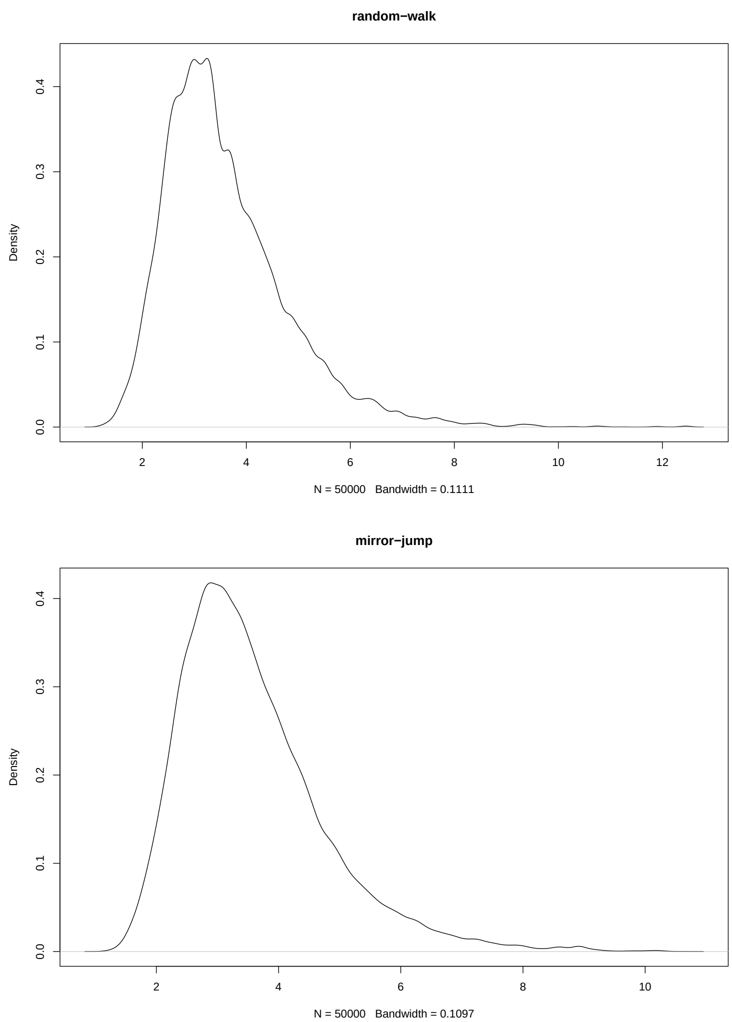

Figure 2.11: Density plots of simulations for θ_3 in NB10 example

List of Tables

List of Figures

References

[AD92] Gelman A. and Rubin DB. Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science*, 7:457–511, 1992.

[Adl81] S. L. Adler. Over-relaxation method for the Monte-Carlo evaluation of the partition function for multiquadratic actions. *Physical Review D-Particles and Fields*, 23(12):2901–2904, 1981.

[AS92] Raftery AL and Lewis S. How many iterations in the Gibbs sampler? *Bayesian Statistics*, 4:763–774, 1992.

[Dra97] David Draper. *Bayesian Hierarchical Modeling*. 1997.

[Gam97] Dani Gamerman. *Markov Chain Monte Carlo: Stochastic simulation for Bayesian inference*. Chapman and Hall, 1997.

[J.89] Geweke J. Bayesian inference in econometric models using Monte Carlo ingeration. *Econometrica*, 57:1317–1339, 1989.

[J.92] Geweke J. Evaluating the accuracy of sampling-based approaches to calculating posterior moments (with discussion). *Bayesian Statistics*, 4:169–193, 1992.

[J.L53] J.L.Doob. *Stochastic Processes*. John Willey and Sons, Inc., 1953.

[L.94] Tierney L. Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 22:1701–62, 1994.

[NAM+53] Metropolis N., Rosenbluth AW, Rosenbluth MN, Teller AH, and Teller E. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[Nea95] Radford M. Neal. Suppressing random walks in Markov Chain Monte Carlo using ordered overrelaxation. *Technical Report No. 9508, Department of Statistics, University of Toronto*, 1995.

[Par62] Emanuel Parzen. *Stochastic Processes*. Holden-Day, Inc., 1962.

[PP83] Heidelberger P. and Welch P. How many iterations in the Gibbs sampler? *Operations Research*, 31:1109–1144, 1983.

[SG84] Geman S. and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[T.63] Bayes T. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763.

[WK70] Hastings WK. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.