Bayesian Computation Via Markov Chain Monte Carlo

Radu V. Craiu and Jeffrey S. Rosenthal

Department of Statistics, University of Toronto, Toronto, Ontario M5S SG3, Canada; email: jeff@math.toronto.edu

Annu. Rev. Stat. Appl. 2014. 1:179–201

First published online as a Review in Advance on October 30, 2013

The *Annual Review of Statistics and Its Application* is online at statistics.annualreviews.org

This article's doi: 10.1146/annurev-statistics-022513-115540

Keywords

Markov chain Monte Carlo, adaptive MCMC, parallel tempering, Gibbs sampler, Metropolis sampler

Abstract

Markov chain Monte Carlo (MCMC) algorithms are an indispensable tool for performing Bayesian inference. This review discusses widely used sampling algorithms and illustrates their implementation on a probit regression model for lupus data. The examples considered highlight the importance of tuning the simulation parameters and underscore the important contributions of modern developments such as adaptive MCMC. We then use the theory underlying MCMC to explain the validity of the algorithms considered and to assess the variance of the resulting Monte Carlo estimators.

1. INTRODUCTION

A search for Markov chain Monte Carlo (MCMC) articles on Google Scholar yields over 100,000 hits, and a general web search on Google yields 1.7 million hits. These results stem largely from the ubiquitous use of these algorithms in modern computational statistics, as we now describe.

MCMC algorithms are used to solve problems in many scientific fields, including physics (where many MCMC algorithms originated), chemistry, and computer science. The widespread popularity of MCMC samplers is largely due to their impact on solving statistical computation problems related to Bayesian inference. Specifically, suppose we are given an independent and identically distributed (i.i.d.) sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ from a parametric sampling density $f(\mathbf{x}|\theta)$, where $\mathbf{x} \in \mathbf{X} \subset \mathbf{R}^k$ and $\theta \in \Theta \subset \mathbf{R}^d$. Suppose we also have some prior density $p(\theta)$. Then, the Bayesian paradigm prescribes that all aspects of inference should be based on the posterior density

$$\pi(\theta|\vec{\mathbf{x}}) = \frac{p(\theta) f(\vec{\mathbf{x}}|\theta)}{\int_\Theta p(\theta) f(\vec{\mathbf{x}}|\theta) \mathrm{d}\theta}, \qquad 1.$$

where $\vec{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Of greatest interest are the posterior means of functionals $g : \mathbf{X} \to \mathbf{R}$, defined by

$$I = \int_\Theta g(\theta)\pi(\theta|\vec{\mathbf{x}}) d\theta = \frac{\int_\Theta g(\theta) p(\theta) f(\vec{\mathbf{x}}|\theta) \mathrm{d}\theta}{\int_\Theta p(\theta) f(\vec{\mathbf{x}}|\theta) \mathrm{d}\theta}. \qquad 2.$$

Such expectations are usually impossible to compute directly because of the integrals that appear in the denominators of Equations 1 and 2. However, we can still study Equation 2 as long as we can sample from π. In the traditional Monte Carlo paradigm, we generate an i.i.d. sample $\theta_1, \ldots, \theta_M$ from π and estimate I from Equation 2 using

$$\hat{I}_M = \frac{1}{M} \sum_{i=1}^{M} g(\theta_i). \qquad 3.$$

This estimate generally works well in cases where the i.i.d. sample $\theta_1, \ldots, \theta_M$ can be generated, and in particular $\hat{I}_M \to I$ with probability 1 as $M \to \infty$.

However, when π is complicated and high-dimensional, classical methods devised to draw independent samples from the distribution of interest cannot be implemented. In this case, an MCMC algorithm proceeds instead by constructing an updating algorithm for generating θ_{t+1} once we know θ_t. MCMC updating algorithms are constructed by specifying a set of transition probabilities for an associated Markov chain (e.g., Meyn & Tweedie 1993, Tierney 1994). The MCMC method then uses the realizations $\theta_1, \ldots, \theta_M$ obtained from the Markov chain as the Monte Carlo sample in Equation 3, or more commonly with the slight modification

$$\hat{I}_M = \frac{1}{M - B} \sum_{i=B+1}^{M} g(\theta_i), \qquad 4.$$

where B is a fixed nonnegative integer (e.g., 1,000) indicating the amount of burn-in, i.e., the number of initial samples that will be discarded because they are excessively biased toward the (arbitrary) initial value θ_0. If the Markov chain has π as an invariant distribution, and if it satisfies the mild technical conditions of being aperiodic and irreducible, then the ergodic theorem implies that with probability one, $\hat{I}_M \to I$ as $M \to \infty$ (see Section 8.1).

Unlike the traditional Monte Carlo methods, in which the samples are independent, MCMC samplers yield dependent draws. Thus, the theoretical study of these algorithms is much more difficult, as is the assessment of their convergence speed and Monte Carlo variance. A comprehensive evolution of the field can be traced through the articles included in volumes edited by Spiegelhalter et al. (2002) and Brooks et al. (2011) and can be found in books devoted to Monte

Table 1 The number of latent membranous lupus nephritis cases (numerator), and the total number of cases (denominator), for each combination of the values of the two covariates

	$IgA = 0$	$IgA = 0.5$	$IgA = 1$	$IgA = 1.5$	$IgA = 2$
$\Delta IgG = -3.0$	0/1	–	–	–	–
$\Delta IgG = -2.5$	0/3	–	–	–	–
$\Delta IgG = -2.0$	0/7	–	–	–	0/1
$\Delta IgG = -1.5$	0/6	0/1	–	–	–
$\Delta IgG = -1.0$	0/6	0/1	0/1	–	0/1
$\Delta IgG = -0.5$	0/4	–	–	1/1	–
$\Delta IgG = 0$	0/3	–	0/1	1/1	–
$\Delta IgG = 0.5$	3/4	–	1/1	1/1	1/1
$\Delta IgG = 1.0$	1/1	–	1/1	1/1	4/4
$\Delta IgG = 1.5$	1/1	–	–	2/2	–

Carlo methods in statistics, such as those by Chen et al. (2000), Liu (2001), and Robert & Casella (2004, 2010). We recognize that for those scientists who are not familiar with MCMC techniques but need to use them for statistical analysis, the wealth of information contained in the literature can be overwhelming. Therefore, this review provides a concise overview of the ingredients needed for using MCMC in most applications. As we discuss these ingredients, we point the reader in need of more sophisticated methods to the relevant literature.

1.1. Example: Lupus Data

As a specific example, we present lupus data that were analyzed first by van Dyk & Meng (2001) and subsequently by Craiu & Meng (2005) and Craiu & Lemieux (2007). These data, presented in **Table 1**, contain disease statuses for 55 patients, 18 of whom have been diagnosed with latent membranous lupus, together with two clinical covariates, IgA and ΔIgG ($\Delta IgG = IgG3 - IgG4$), which are computed from the patients' levels of immunoglobulin of types A and G, respectively.

To model the data generation process we need to formulate the sampling distribution of the binary response variable. We can follow van Dyk & Meng (2001) and consider a probit regression (PR) model: For each patient $1 \le i \le 55$, we model the disease indicator variables as independent,

$$Y_i \sim \text{Bernoulli}\left(\Phi(\mathbf{x}_i^T \beta)\right), \tag{5.}$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of $N(0, 1)$, $x_i = (1, \Delta IgG_i, IgA_i)$ is the covariate vector, and β is a 3×1 parameter vector. We assume a flat prior $p(\beta) \propto 1$ throughout the paper.

For the PR model, the posterior is thus

$$\pi_{\text{PR}}(\vec{\boldsymbol{\beta}}|\vec{\mathbf{Y}}, \vec{\mathbf{IgA}}, \vec{\Delta}\mathbf{IgG}) \propto \prod_{i=1}^{55} \left[\Phi(\beta_0 + \Delta IgG_i\beta_1 + IgA_i\beta_2)^{Y_i} \times (1 - \Phi(\beta_0 + \Delta IgG_i\beta_1 + IgA_i\beta_2))^{(1-Y_i)}\right]. \tag{6.}$$

We return to this example several times below.

1.2. Choice of Markov Chain Monte Carlo Algorithm

Not all MCMC samplers are used equally. Ease of implementation (e.g., preexisting software), simplicity of formulation, computational efficiency, and good theoretical properties all contribute

(not necessarily in that order) to an algorithm's successful and rapid dissemination. In this article, we focus on the most widely used MCMC samplers: the Metropolis–Hastings (MH) algorithm (Section 2), the Gibbs sampler (Section 3), and variable-at-a-time Metropolis (Section 4). We also discuss the optimization and adaptation of MCMC algorithms (Section 5), the use of simulated tempering (Section 6), the assessment of MCMC errors (Section 7), and the theoretical foundations of MCMC (Section 8).

2. THE METROPOLIS–HASTINGS ALGORITHM

2.1. Overview of the Metropolis–Hastings Algorithm

The MH algorithm was developed by Metropolis et al. (1953) and Hastings (1970). It updates the state of the Markov chain as follows. [For simplicity, we write the target (posterior) distribution as simply $\pi(\theta)$.] Assume that the state of the chain at time t is θ_t. Then, the updating rule to construct θ_{t+1} (i.e., the transition kernel for the MH chain) is defined by the following two steps:

Step 1: A proposal ω_t is drawn from a proposal density $q(\omega|\theta_t)$;

Step 2: Set

$$\theta_{t+1} = \begin{cases} \omega_t & \text{with probability } r \\ \theta_t & \text{with probability } 1-r \end{cases},$$

where

$$r = \min\left\{1, \frac{\pi(\omega_t)q(\theta_t|\omega_t)}{\pi(\theta_t)q(\omega_t|\theta_t)}\right\}. \qquad 7.$$

The acceptance probability generated by Equation 7 is independent of the normalizing constant for π (i.e., this probability does not require the value of the denominator in Equation 1) and is chosen precisely to ensure that π is an invariant distribution, the key condition to ensure that $\hat{I}_M \to I$ as $M \to \infty$ as discussed above; see Section 8.2.

The most popular variant of the MH algorithm is the random walk Metropolis (RWM) algorithm, in which $\omega_t = \theta_t + \epsilon_t$, and ϵ_t is generated from a spherically symmetric distribution, e.g., the Gaussian case for which $\epsilon_t \sim N(0, \Sigma)$. Another common choice is the independence sampler (IS), in which $q(\omega|\theta_t) = q(\omega)$; i.e., ω_t does not depend on the current state of the chain, θ_t. In general, RWM is used in situations for which we have little idea about the shape of the target distribution and therefore need to meander through the parameter space. In the opposite situation, in which we have a pretty good idea about the target π, we are able to produce a credible approximation q that can be used as the proposal in the IS algorithm. Modifications of these MH samplers include the delayed-rejection (Green & Mira 2001), multiple-try Metropolis (Casarin et al. 2013, Liu et al. 2000), and reversible-jump algorithms (Green 1995, Richardson & Green 1997), among others.

In practice, one must decide which sampler to use and, maybe more importantly, what values to choose for the simulation parameters. For instance, in the case of the RWM, the proposal covariance matrix Σ plays a crucial role in the performance of the sampling algorithm (Roberts et al. 1997, Roberts & Rosenthal 2001).

2.2. Application to the Lupus Data

To see the effect of these choices in action, let us consider the lupus data under the PR model formulation. The target distribution has density given by Equation 6. Because we have little idea of the shape of π_{PR}, selecting a suitable independence proposal distribution will be difficult. Instead,

we use the RWM algorithm with a Gaussian proposal. We illustrate this using two possible choices for the variance-covariance matrix $\boldsymbol{\Sigma}$ of the Gaussian proposal distribution: $\Sigma_1 = 0.6\,\mathbf{I}_3$ and $\Sigma_2 = 1.2\,\mathbf{I}_3$, where $\mathbf{I}_d$ is the identity matrix in $\mathbf{R}^{d\times d}$.

In **Figure 1*a***, we plot 5,000 samples for (β_0, β_1) obtained from the RWM with proposal variance Σ_1. This plot is superimposed on the two-dimensional projection of the contour plot for the density π_{PR}, which has been obtained from a large Monte Carlo sample produced by a state-of-the-art sampler and which offers an accurate description of the target. The two red lines mark the coordinates of the initial value of the chain chosen to be the maximum likelihood estimate for β. Note that the samples do not cover the entire support of the distribution. Moreover, from the autocorrelation plots shown in **Figure 1*b***, we can see that this chain is very "sticky," i.e., the realizations of the chain are strongly dependent despite an acceptance rate of 39%. As discussed in Section 5, this rate is usually considered reasonably high. Thus, one may be tempted to believe that the strong dependence between the Monte Carlo draws is due to the proposal variance being too small because sampling from a normal distribution with a small variance results in draws that are close to the mean.

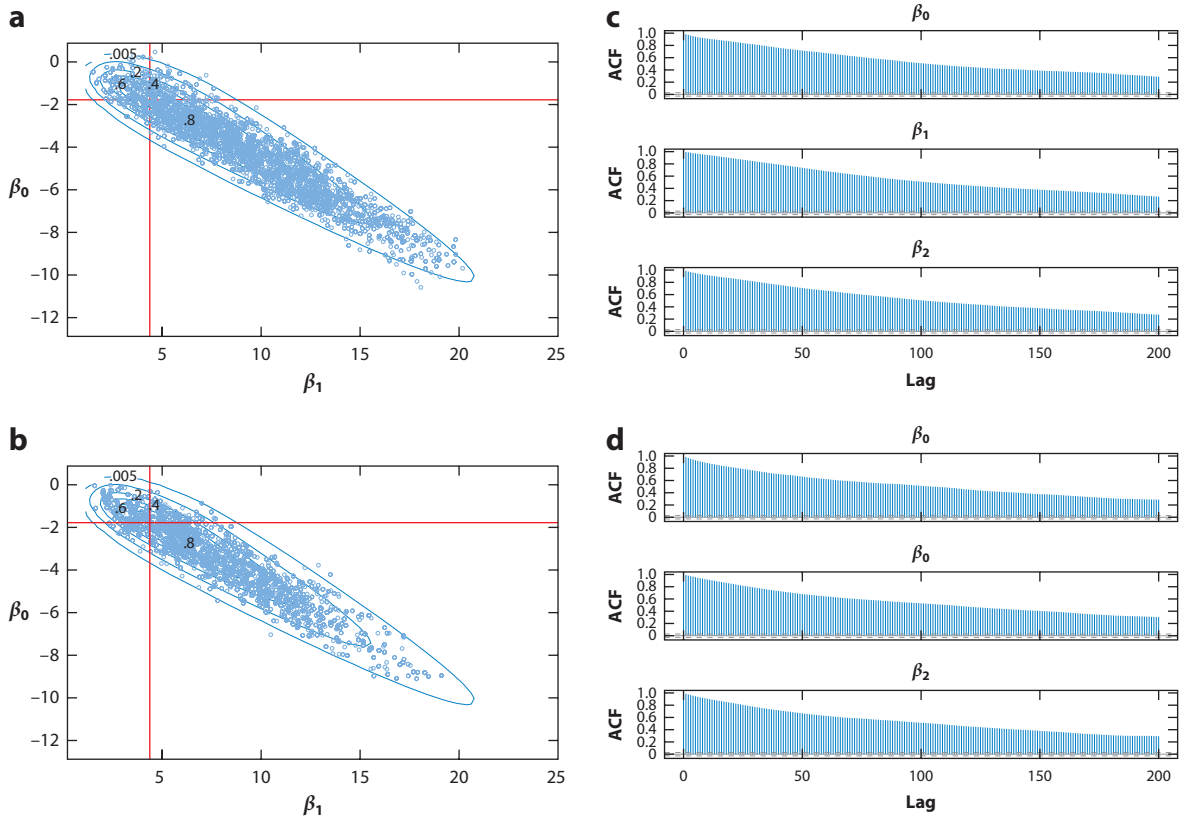

Figure 1

(*a*,*c*) Scatterplots of 5,000 samples for $(\boldsymbol{\beta}_0, \boldsymbol{\beta}_1)$ obtained using random walk Metropolis (RWM) with proposal variances (*a*) $\boldsymbol{\Sigma}_1$ and (*c*) $\boldsymbol{\Sigma}_2$. The points are superimposed on the two-dimensional projection of the contour plot for the target π_{PR}. (*b*,*d*) Autocorrelation plots for the three components of the chain for RWM with proposal variances (*b*) $\boldsymbol{\Sigma}_1$ and (*d*) $\boldsymbol{\Sigma}_2$. Abbreviation: ACF, autocorrelation function.

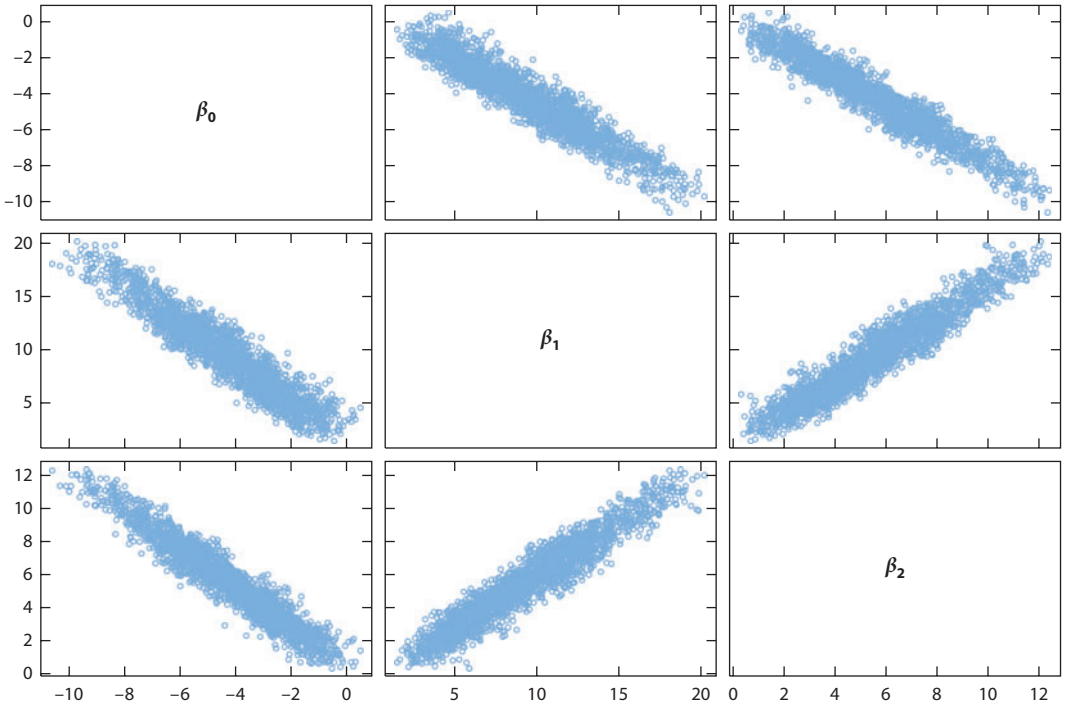

Figure 2

Pair plots for the samples obtained using random walk Metropolis with proposal variance $\boldsymbol{\Sigma}_2$.

We consider doubling the variance and use $\boldsymbol{\Sigma}_2 = 1.2\,\mathbf{I}_3$ as the proposal's covariance matrix. The larger variance brings the acceptance rate down to 24%. **Figure 1*c*** shows the same plots as **Figure 1*a*** for the sampler that uses $\boldsymbol{\Sigma}_2$. The chain seems to travel further into the tails of the distribution, but the serial correlation remains extremely high. Such a high autocorrelation implies that the 5,000-element Monte Carlo sample contains the same amount of information that a much smaller sample of independent realizations would contain. This reduction in effective sample size is computationally wasteful because we spend a lot of time collecting samples that are essentially uninformative. In fact, under certain conditions, Geyer (1992) has shown that the asymptotic variance of $\hat{I}_M$ is σ^2/M, where

$$\sigma^2 = \mathrm{Var}_\pi\{g(\theta)\} + 2\sum_{k=1}^{\infty} \mathrm{cov}\{g(\theta_1), g(\theta_{k+1})\}, \qquad 8.$$

which illustrates the importance of having small correlations between the successive draws θ_t.

The high autocorrelation between the successive draws can be explained if we consider the the strong posterior dependence between the parameters, as illustrated by **Figure 2**, in which we have plotted the samples obtained in pairs. These plots provide an intuitive explanation for the poor mixing exhibited by the two RWM samplers because their proposals have independent components and therefore deviate significantly from the target configuration. We use these RWM algorithms for the aforementioned lupus data to illustrate various theoretical considerations about MCMC in Section 8.4.

3. THE GIBBS SAMPLER

3.1. Overview of the Gibbs Sampler

The Gibbs sampler algorithm was first used by Geman & Geman (1984) in the context of image restoration. Subsequently, Gelfand & Smith (1992) and Tanner & Wong (1987) recognized the algorithm's power for fitting statistical models. Assume that the vector of parameters $\theta \in \mathbf{R}^d$ is partitioned into s subvectors so that $\theta = (\eta_1, \ldots, \eta_s)$. Assume that the current state of the chain is $\theta^{(t)} = (\eta_1^{(t)}, \ldots, \eta_s^{(t)})$. The transition kernel for the Gibbs chain requires updating each subvector in turn by sampling it from its conditional distribution, given all of the other subvectors. More precisely, step $t+1$ of the sampler involves the following updates:

$$\begin{aligned}
\eta_1^{(t+1)} &\sim \pi\left(\eta_1 | \eta_2^{(t)}, \ldots, \eta_s^{(t)}\right) \\
\eta_2^{(t+1)} &\sim \pi\left(\eta_2 | \eta_1^{(t+1)}, \eta_3^{(t)}, \ldots, \eta_s^{(t)}\right) \\
&\ldots \quad \ldots \quad \ldots \\
\eta_s^{(t+1)} &\sim \pi\left(\eta_s | \eta_1^{(t+1)}, \eta_2^{(t+1)}, \ldots, \eta_{s-1}^{(t+1)}\right).
\end{aligned} \tag{9.}$$

Cycling through the blocks in a fixed order defines the Gibbs sampler with deterministic scan; an alternative implementation involves a random scan in which the next block to be updated is sampled at random, and each η_j has a strictly positive probability of being updated. In general, it is not known whether the Gibbs sampler with random scan is more efficient than the Gibbs sampler with deterministic scan (Amit 1991, 1996; Liu et al. 1995). An obvious choice for the blocks η is obtained when $s = d$ and $\eta_j = \theta_j$ for $1 \le j \le d$. Whenever possible, however, the blocks η should contain as many individual components of θ as possible while being able to sample from the conditional distributions in Equation 9 (see the analysis of Liu et al. 1994).

3.2. Application to the Lupus Data

The Gibbs sampler cannot be implemented directly because, as can be seen from Equation 6, the conditional distribution of β_j given the data and all of the other parameters cannot be sampled directly. However, this difficulty dissolves once we expand the model specification to include auxiliary variables (see also Albert & Chib 1993). Specifically, for each $i \in \{1, \ldots, 55\}$, consider the latent variables $\psi_i \sim N(\mathbf{x}_i^T \beta, 1)$, of which only the sign Y_i is observed [i.e., $Y_i = \mathbf{1}(\psi_i > 0)$]. Let $\mathbf{X}$ be the $n \times p$ matrix with ith row $\mathbf{x}_i$ and $\psi = (\psi_1, \ldots, \psi_n)$. After introducing ψ, we notice that the conditional distributions of $\beta|\psi, \mathbf{X}$ and $\psi|\beta, Y$ can be sampled directly. Alternatively, sampling from these two conditional distributions will yield draws from the conditional distribution $p(\beta, \psi|\mathbf{X}, Y)$, the marginal of which, in β, is the target $\pi(\beta)$. The Monte Carlo approach makes marginalization easy because we need only to drop the ψ values from the samples $\{(\beta_t, \psi_t); 1 \le t \le M\}$ drawn from $p(\beta, \psi|\mathbf{X}, Y)$ and thereby retain only the samples $\{\beta_t; 1 \le t \le m\}$ as draws from the target $\pi(\beta)$. This computational strategy of expanding the model so that conditional distributions are available in closed form is known as the data augmentation (DA) algorithm (Tanner & Wong 1987).

The Gibbs sampler (or DA algorithm) for the lupus data alternates between sampling from

$$\beta|\psi, \mathbf{X} \sim N(\tilde{\beta}, (\mathbf{X}^T\mathbf{X})^{-1}),$$

with $\tilde{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\psi$, and

$$\psi_i|\beta, Y_i \sim TN(x_i^T\beta, 1, Y_i),$$

where $TN(\mu, \sigma^2, Y)$ is $N(\mu, \sigma^2)$ truncated to be positive if $Y = 1$ and negative if $Y = 0$. In this formulation, $\eta_1 = (\beta_0, \beta_1, \beta_2)^T$ and $\eta_{j+1} = \psi_j$ for every $j = 1, \ldots, n$.

The Gibbs sampling algorithm does not require tuning and does not reject any of the samples produced. Despite these apparent advantages, the Gibbs sampler is not always preferred over the MH algorithm. For instance, in the PR model considered here, the chain moves slowly across the parameter space. In **Figure 3*a*** we plot its trajectory for the first 300 samples when started at the maximum likelihood estimate (MLE). The sluggishness suggested by **Figure 3*a*** is confirmed by the autocorrelation plots, which show strong and persistent serial dependence for each parameter (**Figure 3*b***). This dependence is not necessarily a characteristic of Gibbs samplers; the high posterior dependence between parameters in the lupus data makes convergence to the target

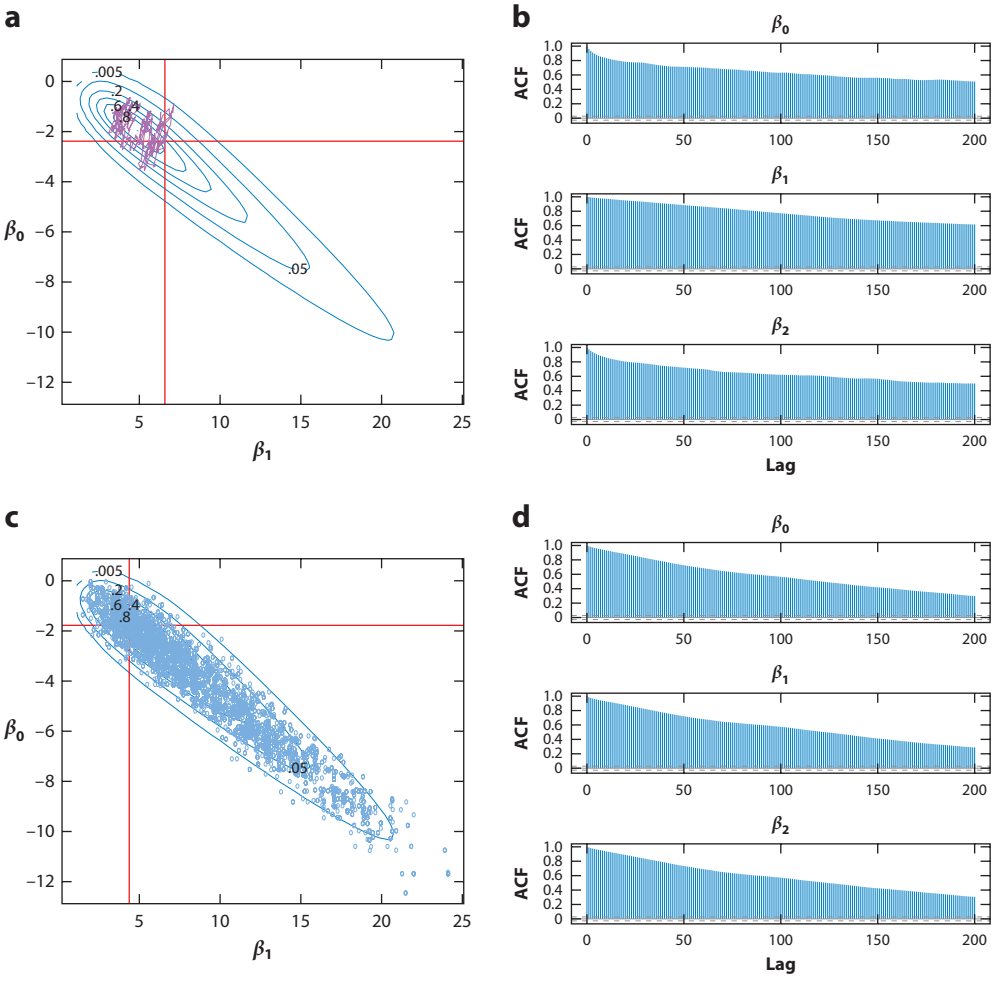

Figure 3

(*a*) Trajectory of the Gibbs chain for 300 updates for (β_0, β_1) (*c*) Scatterplots of 5,000 samples for (β_0, β_1) obtained using variable-at-a-time MH. The points are superimposed on the two-dimensional projection of the contour plot for the target π_{PR}. (*b*,*d*) Autocorrelation plots for the three components of the chain for (*b*) the Gibbs sampler and (*d*) variable-at-a-time MH. Abbreviation: ACF, autocorrelation function; MH, Metropolis–Hastings.

difficult because the Gibbs sampler will always attempt to move the chain in directions that are parallel to the coordinate axes.

DA algorithms have been studied extensively owing to their intensive use in statistical modelling, e.g., linear and nonlinear mixed models and mixture models for which the auxiliary latent variables are natural extensions of the model specification. Liu & Wu (1999) and Meng & van Dyk (1999) propose modified versions of the basic DA algorithm that are designed to break the serial dependence between the Monte Carlo samples and that have the potential to drastically improve the mixing of the Markov chain. We refer the reader to van Dyk & Meng (2001) for an implementation of a marginal DA algorithm for the lupus data.

4. VARIABLE-AT-A-TIME METROPOLIS

4.1. Overview of Variable-at-a-Time Metropolis

Metropolis-style moves can be combined with Gibbs-style variable-at-a-time moves to create a variable-at-a-time Metropolis algorithm. [This algorithm is also sometimes called Metropolis-within-Gibbs, but it was actually the original form of the algorithm used by Metropolis et al. (1953).]

Assume again that the vector of parameters $\theta \in \mathbf{R}^d$ is partitioned into s subvectors such that $\theta = (\eta_1, \ldots, \eta_s)$. Variable-at-a-time Metropolis then proceeds by proposing to move just one coordinate (or subset of coordinates) at a time, leaving all other coordinates fixed. In its most common form, we might try to move the ith coordinate by proposing a new state ω_{t+1}, where $\omega_{t+1,j} = \eta_{t,j}$ for all $j \neq i$, and where $\eta_{t,i} \sim N(\eta_{t,i}, \sigma^2)$. (Here $\omega_{t+1,j}$ is the jth coordinate of ω_{t+1}, etc.) We then accept the proposal ω_{t+1} according to the MH rule (see Equation 7).

As with the Gibbs sampler, we need to choose which coordinate to update each time. Again, we can proceed either by choosing coordinates in the sequence $1, 2, \ldots, d, 1, 2, \ldots$ (systematic-scan) or by choosing the coordinate to update uniformly from $\{1, 2, \ldots, d\}$ on each iteration (random-scan). (In this formulation, one systematic-scan iteration is roughly equivalent to d random-scan ones.)

The variable-at-a-time Metropolis algorithm is often a good generic choice. Unlike the full Metropolis algorithm, it does not require moving all coordinates at once (which can be challenging to do efficiently). In addition, unlike Gibbs sampling, variable-at-a-time Metropolis does not require the ability to sample from the full conditional distributions (which could be infeasible).

4.2. Application to the Lupus Data

We now try using a componentwise RWM to update each coordinate of β. Specifically, the proposal $\omega_{t+1,b}$ is generated from $N(\beta_{t,b}, \sigma_b^2)$ at time $t+1$ for each coordinate b and is accepted with probability

$$\min\{1, \pi(\omega_{t+1,b}|\beta_{t,[-b]})/\pi(\beta_{t,b}|\beta_{t,[-b]})\}, \qquad 10.$$

where $\beta_{t,[-b]}$ is the vector of the most recent updates for all the components of β except β_b. Note that the ratio involved in Equation 10 is identical to $\pi(\omega_{t+1,b}, \beta_{t,[-b]})/\pi(\beta_{t,b}, \beta_{t,[-b]})$ and can be computed in closed form because it is independent of any unknown normalizing constants.

We have implemented the algorithm using $\sigma = (\sqrt{5}, 5, 2\sqrt{2})$. These values were chosen to yield acceptance rates for each component of between 20% and 25%. **Figure 3*c*** shows the samples obtained, and **Figure 3*d*** presents the autocorrelation functions. Notice that although the serial dependence is smaller than in the full MH implementation, it remains high. Also, the samples cover most of the support of the posterior density π. In the one-at-a-time implementation, we are

no longer forcing all components of the chain to move together simultaneously, which seems to improve the spread of the resulting sample.

5. OPTIMIZING AND ADAPTING THE RANDOM WALK METROPOLIS ALGORITHM

Consider the RWM algorithm with proposals $\omega_t = \theta_t + \epsilon_t$, where $\epsilon_t \sim N(0, \Sigma)$ (i.i.d.). Although very specific, this algorithm still allows for great flexibility in the choice of proposal covariance matrix Σ. This raises the question of what Σ leads to the best performance of the algorithm, which we now discuss.

5.1. Optimal Scaling

We first note that if the elements on the diagonal of Σ are very small, then the proposals ω_t will usually be very close to the previous states θ_t. Thus, the proposals will usually be accepted, but the chain will hardly move, which is clearly suboptimal. However, if Σ is very large, then the proposals ω_t will usually be very far from the previous states θ_t. Thus, these proposals (especially in high dimensions) will likely be out in the tails of the target density π in at least one coordinate and thus will likely have much lower π values. This implies that they will almost always be rejected, which is again clearly suboptimal. The optimal scaling, then, is somewhere in between these two extremes. That is, we want our proposal scaling to be neither too small nor too large. Rather, it should be "just right" (this is sometimes called the Goldilocks principle).

In a pioneering paper, Roberts and colleagues (1997) took this a step further, proving that (for a certain idealized high-dimensional limit, at least) the optimal acceptance rate (i.e., the limiting fraction of accepted proposed moves) is equal to the specific fraction 0.234, ~23%. However, any acceptance rate between ~15% and 50% is still fairly efficient (see, e.g., Roberts & Rosenthal 2001, figure 3). Later optimal scaling results obtained by Roberts & Rosenthal (2001) and Bedard (2006) indicate that (again, for a certain idealized high-dimensional limit) the optimal proposal covariance Σ should be chosen to be proportional to the true covariance matrix of the target distribution π (with the constant of proportionality chosen to achieve the 0.234 acceptance rate). In **Figure 4**, we compare two RWM chains with similar acceptance rates but different choices of

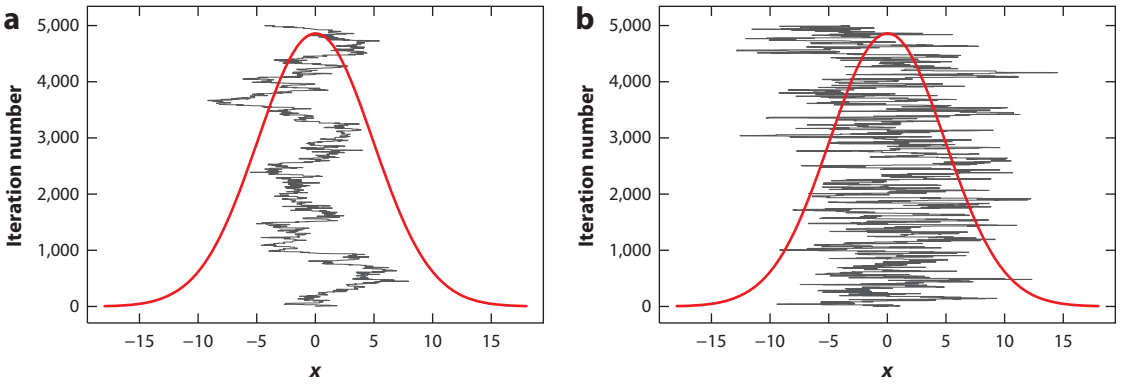

Figure 4

Trace plots of the first coordinate of random walk Metropolis on the same 20-dimensional target. Acceptance rates in both plots are approximately 0.234, and the proposal covariance matrix Σ is proportional to either (*a*) the identity $\mathbf{I}_{20}$ or (*b*) the target covariance matrix. The run in (*b*) clearly mixes much faster.

the proposal distribution variance. The chain for which the variance of the proposal distribution is proportional to that of the target exhibits considerably better mixing.

5.2. Adaptive Markov Chain Monte Carlo

Unfortunately, one generally has little idea about the true covariance of π at the beginning of a simulation. Thus, direct application of the optimal scaling results of the previous section is difficult or impossible. One possible approach is to first perform a number of exploratory MCMC runs to get an idea of the geography of the target's important regions and to then use this knowledge to tune the proposal to be approximately optimal. However, this approach requires restarting the chain multiple times, using each run to tune different subsets of the simulation parameters. Because this process can be lengthy and onerous, especially in high-dimensional spaces, it generally has limited success in complex models.

Alternatively, one can build upon the recent advances in adaptive MCMC (AMCMC) in which the proposal distribution is updated continuously at any time t using the information contained in the samples obtained up until that time (see, e.g., Bai et al. 2011, Craiu et al. 2009, Haario et al. 2001, Roberts & Rosenthal 2009). Such an approach does not require restarting the chain and can be fully automated. However, this approach requires careful theoretical analysis because the process loses its Markovian property by using the past realizations of the chain (instead of using only its current state), and asymptotic ergodicity must be proven on a case-by-case basis. Fortunately, the general frameworks developed by, e.g., Andrieu et al. (2005) and Roberts & Rosenthal (2007), have made proving the validity of adaptive samplers easier.

5.3. Application to the Lupus Data

We have implemented the adaptive RWM proposed by Haario et al. (2001) (see also Roberts & Rosenthal 2009) in which at each time $t > 1{,}000$, we use the following approximation for Σ in the Gaussian proposal:

$$\Sigma_t = \frac{(2.4)^2}{3} SamVar_t + \epsilon \mathbf{I}_3, \qquad 11.$$

where $\epsilon = 0.01$ and $SamVar_t$ is the sample variance of all samples drawn up to time $t-1$. This adaptation attempts to mimic the theoretical optimal scaling results discussed in Section 5.1; if $SamVar_t$ happened to equal the true covariance matrix of π and if $\epsilon = 0$, then Equation 11 would indeed be the optimal proposal covariance. **Figure 5** shows the same plots as those generated for **Figure 1*a–d***. The reduction in serial autocorrelation is apparent. For instance, the mean, median, lower quartile, and upper quartile for the autocorrelations of the RWM sampler with Σ_2 computed up to lag 200 equal 0.537, 0.513, 0.377, and 0.664, respectively; they equal the much smaller values 0.065, 0.029, 0.007, and 0.059, respectively, for the adaptive RWM.

6. SIMULATED TEMPERING

Particular challenges arise in MCMC when the target density π is multimodal, i.e., the target density has distinct high-probability regions separated by low-probability barriers that are difficult for the Markov chain to traverse. In such cases, a simple MCMC algorithm such as RWM may easily explore well within any one modal region, but the chain may take an unfeasibly long time to move between modes. This leads to extremely slow convergence and poor resulting estimates.

Simulated tempering attempts to flatten out the distribution into related distributions with less pronounced modes that can be sampled more easily. If done carefully, simulated tempering can

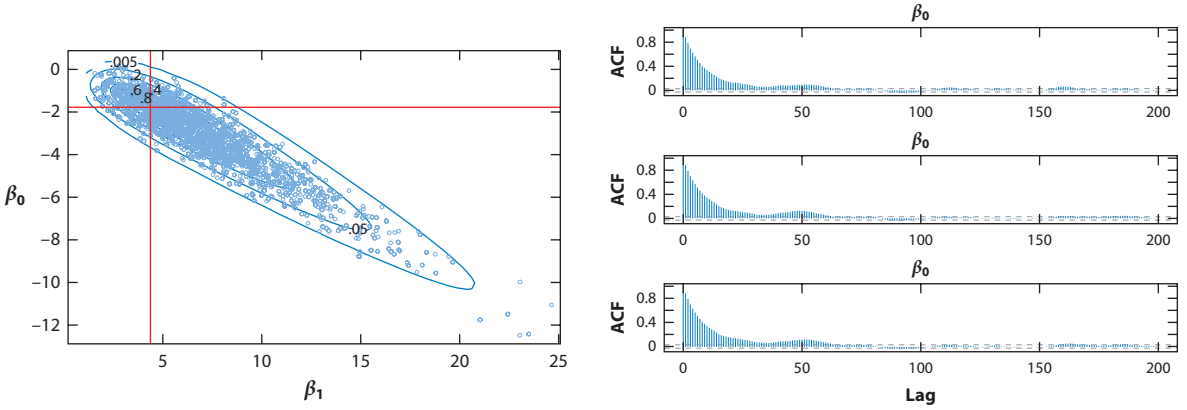

Figure 5

Left panel: Scatterplot of 30,000 samples for (β_0, β_1) obtained using random walk Metropolis (RWM) with adaptive variance. The points are superimposed on the two-dimensional projection of the contour plot for the target π_{PR}. *Right panels*: Autocorrelation plots for the three components of the chain show much lower serial dependence when compared with nonadaptive RWM samplers. Abbreviation: ACF, autocorrelation function.

compensate for this flattening out, ultimately yielding good estimates for expected values from the original target density π, as explained below.

Specifically, simulated tempering requires a sequence $\pi_1, \pi_2, \ldots, \pi_m$ of target densities, where $\pi_1 = \pi$ is the original density and π_τ is flatter for the distributions for which τ is large. (The parameter τ is usually referred to as the temperature, making π_1 the cold density and π_τ for larger values of τ the so-called hot densities.) These different densities can then be combined to define a single joint density $\bar{\pi}$ on $\Theta \times \{1, 2, \ldots, m\}$, defined by $\bar{\pi}(\theta, \tau) = \frac{1}{m}\pi_\tau(\theta)$ for $1 \leq \tau \leq m$ and $\theta \in \Theta$. (Weights other than the uniform choice, $\frac{1}{m}$, may also be used.)

Simulated tempering then uses $\bar{\pi}$ to define a joint Markov chain (θ, τ) on $\Theta \times \{1, 2, \ldots, m\}$, with target density $\bar{\pi}$. In the simplest case, this chain is a version of variable-at-a-time Metropolis that alternates (say) between spatial moves, which propose (say) $\theta' \sim N(\theta, \sigma_\theta^2)$ and accept with the usual Metropolis probability $\min(1, \frac{\bar{\pi}(\theta', \rightleftarrows \tau)}{\bar{\pi}(\theta, \rightleftarrows \tau)}) = \min(1, \frac{\pi_\tau(\theta')}{\pi_\tau(\theta)})$, and temperature moves, which propose (say) $\tau' = \tau \pm 1$ (with a probability of $\frac{1}{2}$ each) and accept with the usual Metropolis probability $\min(1, \frac{\bar{\pi}(\theta, \tau')}{\bar{\pi}(\theta, \tau)}) = \min(1, \frac{\pi_{\tau'}(\theta)}{\pi_\tau(\theta)})$.

As is usual for Metropolis algorithms, this chain should converge in distribution to the density $\bar{\pi}$. But, of course, our interest is in the original density $\pi = \pi_1$, not in $\bar{\pi}$. The genius of simulated tempering is that ultimately, we count only those samples corresponding to $\tau = 1$. That is, once we have a good sample from $\bar{\pi}$, we simply discard all the sample values corresponding to $\tau \neq 1$, and what remains is a good sample from π.

6.1. A Simple Example

For a specific example, suppose the target density is given by $\pi(\theta) = \frac{1}{2}N(0, 1; \theta) + \frac{1}{2}N(20, 1; \theta)$. This target density is a mixture of the standard normal density $N(0, 1; \theta)$ and the normal density $N(20, 1; \theta)$ with mean 20 and variance 1. This chain is highly multimodal (**Figure 6*a***), leading to very poor mixing of ordinary RWM (**Figure 7*a***).

However, if $\pi_\tau(\theta) = \frac{1}{2}N(0, \tau^2; \theta) + \frac{1}{2}N(20, \tau^2; \theta)$, i.e., a mixture of two normal densities with means 0 and 20 but with variances τ^2 instead of 1, then $\pi_1 = \pi$ is the original target

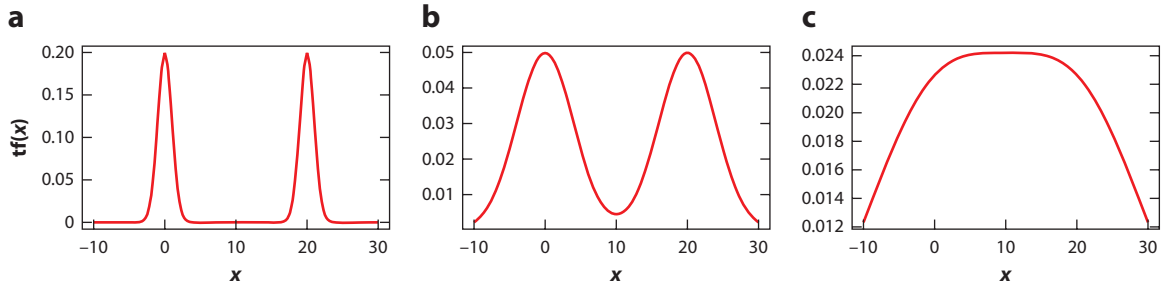

Figure 6

(*a*) The highly multimodal target density $\pi(\theta) = \frac{1}{2}N(0, 1; \theta) + \frac{1}{2}N(20, 1; \theta)$. (*b*) A somewhat flatter density $\pi_4 = \frac{1}{2}N(0, 4^2; \theta) + \frac{1}{2}N(20, 4^2; \theta)$. (*c*) An even flatter density $\pi_{10} = \frac{1}{2}N(0, 10^2; \theta) + \frac{1}{2}N(20, 10^2; \theta)$.

density, but π_τ becomes flatter for larger τ (**Figure 6*a–c***). This behavior allows us to define a joint simulated tempering chain on $\bar{\pi}$ (with proposal scaling $\sigma_\theta = 1$, say), which mixes much faster owing to the flattened high-temperature distributions (**Figure 7*b***). We can then identify the θ values corresponding to $\tau = 1$ in this faster-mixing joint chain to get a very good sample from $\pi_1 = \pi$ (**Figure 7*c***). As with all Monte Carlo sampling, a good sample from π allows us to compute good estimates for expected values $I = \mathrm{E}(g)$ for functionals g of interest.

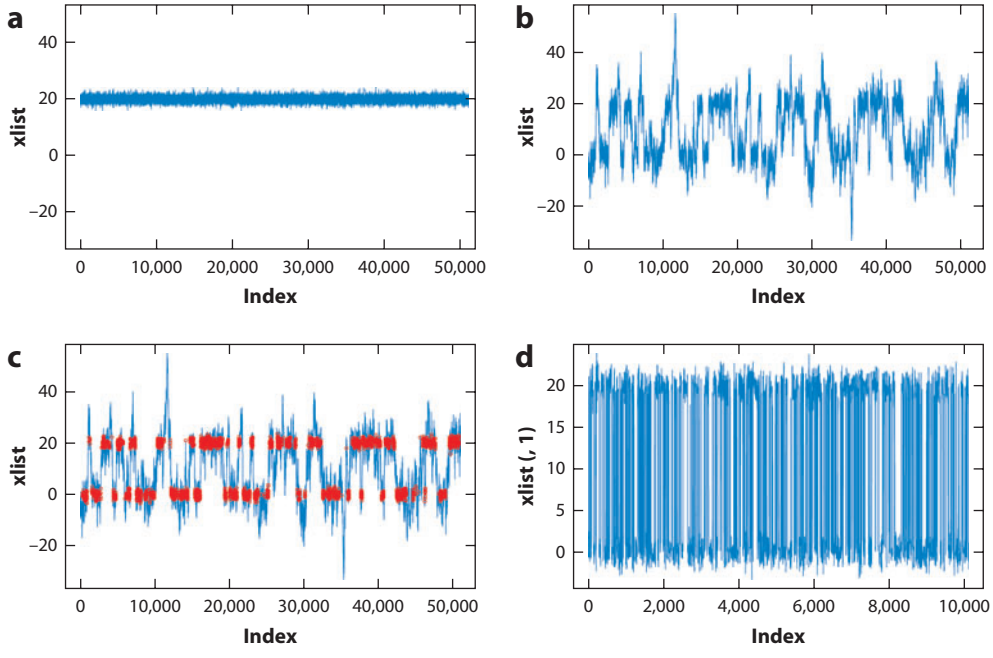

Figure 7

Trace plots for the highly multimodal target density $\pi(\theta) = \frac{1}{2}N(0, 1; \theta) + \frac{1}{2}N(20, 1; \theta)$. (*a*) Ordinary random walk Metropolis gets stuck in the modal region of π near 20 and cannot find the second modal region near 0. (*b*) The θ coordinates of simulated tempering for $\bar{\pi}$. (*c*) Red circles indicate the θ values of the simulated tempering corresponding to $\tau = 1$ (and hence to π). (*d*) The θ_1 coordinates for the corresponding parallel tempering algorithm, showing excellent mixing.

6.2. Choosing the Tempered Distributions

Simulated tempering often works quite well, but it raises the question of how to find appropriate tempered distributions π_τ. Usually, we will not know convenient choices such as the one given above, $\pi_\tau = \frac{1}{2}N(0, \tau^2) + \frac{1}{2}N(20, \tau^2)$. Thus, we require more generic choices.

One promising approach is to let the hotter densities $\pi_\tau(\theta)$ correspond to taking smaller and smaller powers of the original target density $\pi(\theta)$, i.e., to let $\pi_\tau(\theta) = c_\tau(\pi(\theta))^{1/\tau}$ for an appropriate normalizing constant c_τ. (It is common to write $\beta = 1/\tau$ and refer to β as the inverse temperature.) This formula guarantees that $\pi_1 = \pi$ and that π_τ will be flatter for larger τ (because small positive powers move all positive numbers closer to 1), which is precisely what we need. As a specific example, if $\pi(\theta)$ happened to be the density of $N(\mu, \sigma^2)$, then $c_\tau(\pi(\theta))^{1/\tau}$ would be the density of $N(\mu, \tau\sigma^2)$. This is indeed a flatter density, similar to the simple example above, which confirms that this approach holds promise.

Unfortunately, this approach has the following problem. If we propose to move τ to τ', then with this formula, we should accept this proposal with probability

$$\min\left(1, \frac{\pi_{\tau'}(\theta)}{\pi_\tau(\theta)}\right) = \min\left(1, \frac{c_{\tau'}}{c_\tau}(\pi(\theta))^{(1/\tau')-(1/\tau)}\right).$$

This formula explicitly depends on the normalizing constants c_τ and $c_{\tau'}$; these constants do not cancel as they do in ordinary RWM. This dependence is problematic because the values of c_τ are usually unknown and infeasible to calculate. So, what can be done?

6.3. Parallel Tempering

One idea is to use parallel tempering, sometimes called Metropolis-coupled MCMC (MCMCMC). In this algorithm, the state space is Θ^m, corresponding to m different chains, each with its own value of θ. So, the state at time t is given by $\theta_t = (\theta_{t1}, \theta_{t2}, \ldots, \theta_{tm})$. Intuitively, each $\theta_{t\tau}$ is at its own temperature τ, i.e., converging towards its own target density π_τ. The overall target density is now $\bar{\pi}(\theta) = \pi_1(\theta_1)\pi_2(\theta_2)\ldots\pi_m(\theta_m)$, i.e., the density that makes each coordinate of θ independent and following the density of its own temperature. For any $1 \le \tau \le m$, then, the algorithm can update the chain $\theta_{t-1,\tau}$ at temperature τ by proposing (say) $\theta'_{t,\tau} \sim N(\theta_{t-1,\tau}, \sigma^2)$ and accepting this proposal with the usual Metropolis probability $\min(1, \frac{\pi_\tau(\theta'_{t,\tau})}{\pi_\tau(\theta_{t-1,\tau})})$.

Crucially, the chain can also choose temperatures τ and τ' (perhaps choosing each temperature uniformly from $\{1, 2, \ldots, m\}$), and it can then propose to swap the values $\theta_{n,\tau}$ and $\theta_{t,\tau'}$. This proposal will then be accepted with its usual Metropolis probability, $\min(1, \frac{\pi_\tau(\theta_{t,\tau'})\pi_{\tau'}(\theta_{t,\tau})}{\pi_\tau(\theta_{t,\tau})\pi_{\tau'}(\theta_{t,\tau'})})$. The beauty of parallel tempering is that it allows the normalizing constants to cancel. That is, if $\pi_\tau(\theta) = c_\tau(\pi(\theta))^{1/\tau}$, then the acceptance probability becomes

$$\min\left(1, \frac{c_\tau\pi(\theta_{t,\tau'})^{1/\tau}c_{\tau'}\pi(\theta_{t,\tau})^{1/\tau'}}{c_\tau\pi(\theta_{t,\tau})^{1/\tau}c_{\tau'}\pi(\theta_{t,\tau'})^{1/\tau'}}\right) = \min\left(1, \frac{\pi(\theta_{t,\tau'})^{1/\tau}\pi(\theta_{t,\tau})^{1/\tau'}}{\pi(\theta_{t,\tau})^{1/\tau}\pi(\theta_{t,\tau'})^{1/\tau'}}\right).$$

Thus, the values of c_τ and $c_{\tau'}$ are not required to run the algorithm.

As a first test, we can apply parallel tempering to the simple example given above, again using $\pi_\tau(\theta) = \frac{1}{2}N(0, \tau^2; \theta) + \frac{1}{2}N(20, \tau^2; \theta)$ for $\tau = 1, 2, \ldots, 10$. Parallel tempering works pretty well in this case (**Figure 7*d***). Of course, the normalizing constants in this example were known, so parallel tempering was not really required. However, these constants are unknown in many applications; parallel tempering is often a useful sampling method in such cases.

7. ASSESSING MARKOV CHAIN MONTE CARLO ERRORS

When considering any statistical estimation procedure, the amount of uncertainty in the estimate, e.g., some measure of its standard error, is an important issue. In conventional Monte Carlo algorithms, where the $\{\theta_i\}$ are i.i.d., as in Equation 3, the standard error is given by $\frac{1}{\sqrt{M}}\mathrm{SD}(g(\theta))$, where $\mathrm{SD}(g(\theta))$ is the usual estimate of the standard deviation of the distribution of the $g(\theta_i)$. However, with MCMC there is usually extensive serial correlation in the samples θ_i, so the usual i.i.d.-based estimate of standard error does not apply. So-called perfect sampling algorithms are an exception [see, for example, Propp & Wilson (1996) or Craiu & Meng (2011)], but they are hard to adapt for Bayesian computation. Indeed, the standard error for MCMC is usually both larger than in the i.i.d. case (owing to the correlations) and harder to quantify.

The simplest way to estimate standard error from an MCMC estimate is to rerun the entire Markov chain several times using the same values of run length M and burn-in B as in Equation 4 but starting from different initial values θ_0 drawn from the same overdispersed (i.e., well spread-out) initial distribution. This process leads to a sequence of i.i.d. estimates of the target expectation I, and standard errors from the resulting sequence of estimates can then be computed in the usual i.i.d. manner. (We illustrate this in Section 8.4 using the RWM algorithms for the lupus data presented in Section 2.1.) However, such a procedure is often highly inefficient, raising the question of how to estimate standard error from a single run of a single Markov chain. Specifically, we would like to estimate $v \equiv \mathrm{Var}(\frac{1}{M-B}\sum_{i=B+1}^{M} g(\theta_i))$.

7.1. Variance Estimate

To estimate the variance of v above, let $\bar{g}(\theta) = g(\theta) - \mathrm{E}(g)$, so $\mathrm{E}(\bar{g}) = 0$. And, assume that B is large enough that $\theta_i \approx \pi$ for $i > B$. Then, writing $\approx$ to mean "equal in the limit as $M \to \infty$," we compute that

$$
\begin{aligned}
v &\approx \mathrm{E}\left[\left(\left(\frac{1}{M-B}\sum_{i=B+1}^{M} g(\theta_i)\right) - \mathrm{E}(g)\right)^2\right] = \mathrm{E}\left[\left(\frac{1}{M-B}\sum_{i=B+1}^{M} \bar{g}(\theta_i)\right)^2\right] \\
&= \frac{1}{(M-B)^2}\left[(M-B)\mathrm{E}(\bar{g}(\theta_i)^2) + 2(M-B-1)\mathrm{E}(\bar{g}(\theta_i)\bar{g}(\theta_{i+1}))\right. \\
&\quad \left. + 2(M-B-2)\mathrm{E}(\bar{g}(\theta_i)\bar{g}(\theta_{i+2})) + \cdots\right] \\
&\approx \frac{1}{M-B}\left(\mathrm{E}(\bar{g}(\theta_i)^2) + 2\mathrm{E}(\bar{g}(\theta_i)\bar{g}(\theta_{i+1})) + 2\mathrm{E}(\bar{g}(\theta_i)\bar{g}(\theta_{i+2})) + \cdots\right) \\
&= \frac{1}{M-B}\left(\mathrm{Var}_\pi(g) + 2\mathrm{Cov}_\pi(g(\theta_i)g(\theta_{i+1})) + 2\mathrm{Cov}_\pi(g(\theta_i)g(\theta_{i+2})) + \cdots\right) \\
&= \frac{1}{M-B}\mathrm{Var}_\pi(g)\left(1 + 2\mathrm{Corr}_\pi(g(\theta_i), g(\theta_{i+1})) + 2\mathrm{Corr}_\pi(g(\theta_i), g(\theta_{i+2})) + \cdots\right) \\
&\equiv \frac{1}{M-B}\mathrm{Var}_\pi(g)(ACT) = (\textit{i.i.d. variance})\,(ACT),
\end{aligned}
$$

where *i.i.d. variance* is the value for the variance that we would obtain if the samples $\{\theta_i\}$ were in fact i.i.d., and

$$
ACT = 1 + 2\sum_{k=1}^{\infty}\mathrm{Corr}_\pi(g(\theta_0), g(\theta_k)) \equiv 1 + 2\sum_{k=1}^{\infty}\rho_k = \sum_{k=-\infty}^{\infty}\rho_k = 2\left(\sum_{k=0}^{\infty}\rho_k\right) - 1
$$

is the factor by which the variance is multiplied owing to the serial correlations from the Markov chain (sometimes called the integrated autocorrelation time). Here, Corr_π refers to the theoretical correlation that would arise from a sequence $\{\theta_i\}_{i=-\infty}^{\infty}$ that was in stationarity (so each θ_i had density π) and that followed the Markov chain transitions. This assumption implies that the correlations

are a function of the time lag between the two variables and, in particular, that $\rho_{-k} = \rho_k$ as above. The standard error is then given by $se = \sqrt{v} = (i.i.d. - se)\sqrt{ACT}$.

Now, both the *i.i.d. variance* and the quantity ACT can be estimated from the sample run. (For example, the built-in ACF function in R automatically computes the lag correlations ρ_k. Note also that when computing ACT in practice, we do not sum over all k. Rather, we sum only until, say, $|\rho_k| < 0.05$ or $\rho_k < 0$, because for large k we should have $\rho_k \approx 0$ but the estimates of ρ_k will always contain some sampling error.) This procedure provides a method of estimating the standard error of the sample. It also provides a method of comparing different MCMC algorithms because in most cases, $ACT \gg 1$ and better chains would have smaller ACT values. In the most extreme case, one sometimes even tries to design antithetic chains for which $ACT < 1$ (see Adler 1981, Barone & Frigessi 1990, Craiu & Lemieux 2007, Craiu & Meng 2005, Neal 1995).

7.2. Confidence Intervals

Suppose we estimate $u \equiv \mathrm{E}(g)$ by the quantity $e = \frac{1}{M-B}\sum_{i=B+1}^{M} g(\theta_i)$ and obtain an estimate e and an approximate variance (as above) v. Then what is, say, a 95% confidence interval for u?

Well, if a central limit theorem (CLT) applies (as discussed in Section 8), then for large values of $M - B$, we have the normal approximation that $e \approx N(u, v)$. It then follows as usual that $(e-u)v^{-1/2} \approx N(0,1)$, so $\mathbf{P}(-1.96 < (e-u)v^{-1/2} < 1.96) \approx 0.95$, so $\mathbf{P}(-1.96\sqrt{v} < e - u < 1.96\sqrt{v}) \approx 0.95$. This gives us our desired confidence interval: With a probability of 95%, the interval $(e - 1.96\sqrt{v}, e + 1.96\sqrt{v})$ will contain u. (Strictly speaking, we should use the t distribution, not the normal distribution. But if $M - B$ is at all large, then the t and normal distributions are very similar. Thus, we can ignore this issue for now.) Such confidence intervals allow us to more appropriately assess the uncertainty of our MCMC estimates (e.g., Flegal et al. 2008).

The above analysis raises the question of whether a CLT even holds for Markov chains. We answer this and other questions when we consider the theory of MCMC in the following section.

8. THEORETICAL FOUNDATIONS OF MARKOV CHAIN MONTE CARLO

We close with some theoretical considerations about MCMC. Specifically, why does MCMC work? The key is that the distribution of θ_n converges in various senses to the target distribution $\pi(\cdot)$. This convergence follows from basic Markov chain theory, as we discuss below.

8.1. Markov Chain Convergence

A basic fact about Markov chains is that if a Markov chain is irreducible and aperiodic, with stationarity distribution π, then θ_t converges in distribution to π as $t \to \infty$. More precisely, we have Theorem 1 (see, e.g., Meyn & Tweedie 1993, Roberts & Rosenthal 2004, Rosenthal 2001, Tierney 1994).

> **Theorem 1:** If a Markov chain is irreducible, with stationarity probability density π, then for π-a.e. initial value θ_0: (*a*) if $g:\Theta \to \mathbf{R}$ with $\mathrm{E}(|g|) < \infty$, then $\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} g(\theta_i) = \mathrm{E}(g) \equiv \int g(\theta)\pi(\theta)\mathrm{d}\theta$, and (*b*) if the chain is also aperiodic, then furthermore $\lim_{t\to\infty} \mathbf{P}(\theta_t \in S) = \int_S \pi(\theta)\,\mathrm{d}\theta$ for all measurable $S \subseteq \Theta$.

We now discuss the various conditions of the theorem, one at a time.

Being irreducible means, essentially, that the chain has positive probability of eventually getting from any one location to any other. In the discrete case, we can define irreducibility as saying that for all $i, j \in \Theta$ there exists $t \in \mathbf{N}$ such that $\mathbf{P}(\theta_t = j|\theta_0 = i) > 0$. In the general case, however, this definition is problematic because the probability of hitting any particular state is usually zero. Instead, we can define irreducibility (or ϕ-irreducibility) as saying that there is some reference measure ϕ such that for all $\zeta \in \Theta$, and for all $A \subseteq \Theta$ with $\phi(A) > 0$, there exists $t \in \mathbf{N}$ such that $\mathbf{P}(\theta_t \in A|\theta_0 = \zeta) > 0$. This condition is usually satisfied for MCMC (aside from certain rare cases in which the state space consists of highly disjoint pieces) and is generally not a concern.

Being aperiodic means that there are no forced cycles, i.e., that there do not exist disjoint nonempty subsets $\Theta_1, \Theta_2, \ldots, \Theta_d \subseteq \Theta$ for some $d \geq 2$ such that $P(\theta_{t+1} \in \Theta_{i+1}|\theta_t = \zeta) = 1$ for all $\zeta \in \Theta_i (1 \leq i \leq d-1)$, and $P(\theta_{t+1} \in \Theta_1|\theta_t = \zeta) = 1$ for all $\zeta \in \Theta_d$. This condition virtually always holds for MCMC. For example, it holds (*a*) if $P(\theta_{t+1} = \zeta|\theta_t = \zeta) > 0$, as for the Metropolis algorithm (owing to the positive probability of rejection); (*b*) if two iterations are sometimes equivalent to just one, as for the Gibbs sampler; or (*c*) if the transition probabilities have positive densities throughout Θ, as is often the case. In short, we have never known aperiodicity to be a problem for MCMC.

The condition that the density π be stationary for the chain is the most subtle one, as we discuss next.

8.2. Reversibility and Stationarity of Markov Chains

For ease of notation, this section focuses on discrete Markov chains, although the general case is similar upon replacing probability mass functions with measures and sums with integrals. We thus let π be a probability mass function on Θ and assume for simplicity that $\pi(\theta) > 0$ for all $\theta \in \Theta$. We also let $P(i, j) = \mathbf{P}(\theta_1 = j|\theta_0 = i)$ be the Markov chain's transition probabilities.

We say that π is stationary for the Markov chain if it is preserved under the chain's dynamics, i.e., if the chain has the property that whenever $\theta_0 \sim \pi$ [meaning that $\mathbf{P}(\theta_0 = i) = \pi(i)$ for all $i \in \Theta$], then also $\theta_1 \sim \pi$ [i.e., $\mathbf{P}(\theta_1 = i) = \pi(i)$ for all $i \in \Theta$]. Equivalently, $\sum_{i \in \Theta} \pi(i) P(i, j) = \pi(j)$ for all $j \in \Theta$. Intuitively, this means that the probabilities π are left invariant by the chain, which explains why the chain might converge to those probabilities in the limit.

We now show that reversibility is automatically satisfied by MH algorithms, thereby explaining why the Metropolis acceptance probabilities are defined as they are. Indeed, let $q(i, j) = \mathbf{P}(\omega_t = j|\theta_{t-1} = i)$ be the proposal distribution, which is then accepted with probability $\min(1, \frac{\pi(j)q(j,i)}{\pi(i)q(i,j)})$. Then, for $i, j \in \Theta$ with $i \neq j$,

$$P(i, j) = q(i, j) \min\left(1, \frac{\pi(j)q(j, i)}{\pi(i)q(i, j)}\right).$$

It follows that

$$\pi(i)P(i, j) = \pi(i)q(i, j) \min\left(1, \frac{\pi(j)q(j, i)}{\pi(i)q(i, j)}\right) = \min(\pi(i)q(i, j), \pi(j)q(j, i)).$$

By inspection, this last expression is symmetric in i and j. It then follows that $\pi(i)P(i, j) = \pi(j)P(j, i)$ for all $i, j \in \Theta$ (at least for $i \neq j$, but the case $i = j$ is trivial). This property is described as π being reversible for the chain. [Intuitively, reversibility implies that if $\theta_0 \sim \pi$, then $\mathbf{P}(\theta_0 = i, \theta_1 = j) = \mathbf{P}(\theta_0 = j, \theta_1 = i)$, i.e., the probability of starting at i and moving to j is the same as that of starting at j and moving to i. This property is also called being time reversible.]

The importance of reversibility is that it implies stationarity of π. Indeed, using reversibility, we compute that if $\theta_0 \sim \pi$, then

$$\mathbf{P}(\theta_1 = j) = \sum_{i \in \Theta} \mathbf{P}(\theta_0 = i)\mathbf{P}(i, j) = \sum_{i \in \Theta} \pi(i)P(i, j) = \sum_{i \in \Theta} \pi(j)P(j, i)$$
$$= \pi(j) \sum_{i \in \Theta} P(j, i) = \pi(j).$$

Thus, $\theta_1 \sim \pi$, too, so π is stationary.

We conclude that the stationarity condition holds automatically for any MH algorithm. Hence, assuming irreducibility and aperiodicity (which, as noted above, are virtually always satisfied for MCMC), Theorem 1 applies and establishes the asymptotic validity of MCMC.

8.3. Markov Chain Monte Carlo Convergence Rates

Write $P^t(\zeta, S) = \mathbf{P}[\theta_t \in S | \theta_0 = \zeta]$ for the t-step transition probabilities for the chain, and let $D(\zeta, t) = ||P^t(\zeta, \cdot) - \Pi(\cdot)|| \sup_{S \subseteq \Theta} |P^t(\zeta, S) - \Pi(S)|$ be a measure (specifically, the total variation distance) of the chain's distance from stationarity after t steps, where $\Pi(S) = \int_S \pi(\zeta)\, d\zeta$ is the target probability distribution. Then, the chain is said to be ergodic if $\lim_{t \to \infty} D(\zeta, t) = 0$ for π-a.e. $\zeta \in \Theta$, i.e., if the chain transition probabilities $P^t(\zeta, S)$ converge (uniformly) to Π as $t \to \infty$, which Theorem 1 indicates is usually true for MCMC. However, ergodicity alone says nothing about the convergence rate, i.e., how quickly this convergence occurs.

By contrast, a quantitative bound on convergence is an actual number t^* such that $D(\zeta, t^*) < 0.01$, i.e., such that the chain's probabilities are within 0.01 of stationary after t^* iterations. (The cutoff value 0.01 is arbitrary but has become fairly standard.) We then sometimes say that the chain "converges in t^* iterations." Quantitative bounds, when available, are the most useful because they provide precise instructions about how long an MCMC algorithm must be run. Unfortunately, these bounds are often difficult to establish for complicated statistical models, although some progress has been made (e.g., Douc et al. 2004; Jones & Hobert 2001; Rosenthal 1995, 2002).

Halfway between these two extremes is geometric ergodicity, which is more useful than plain ergodicity but which is often easier to compute than are quantitative bounds. A chain is geometrically ergodic if there are $\rho < 1$ and Π-a.e.-finite $M : \Theta \to [0, \infty]$ such that $D(\zeta, t) \leq M(\zeta)\rho^t$ for all $\zeta \in \Theta$ and $t \in \mathbf{N}$, i.e., such that the convergence to Π happens exponentially quickly.

If a Markov chain is geometrically ergodic and $g : \Theta \to \mathbf{R}$ such that $\mathrm{E}(|g|^{2+a}) < \infty$ for some $a > 0$, then a CLT holds for quantities such as $e = \frac{1}{M-B} \sum_{i=B+1}^{M} g(\theta_i)$ (Geyer 1992, Tierney 1994), and we have the normal approximation that $e \approx N(u, v)$. [In fact, if the Markov chain is reversible as above, then it suffices to take $a = 0$ (Roberts & Rosenthal 1997).] As explained in Section 7.2, this approximation is key to obtaining confidence intervals and thus more reliable estimates.

Now, if the state space Θ is finite, then assuming irreducibility and aperiodicity, any Markov chain on Θ is always geometrically ergodic. However, this result is not true for infinite state spaces. The RWM algorithm is known to be geometrically ergodic essentially (i.e., under a few mild technical conditions) if and only if π has exponential tails, i.e., there are $a, b, c > 0$ such that $\pi(\theta) \leq a e^{-b|\theta|}$ whenever $|\theta| > c$ (Mengersen & Tweedie 1996, Roberts & Tweedie 1996). The Gibbs sampler is known to be geometrically ergodic for certain models (e.g., Papaspiliopoulos & Roberts 2008). But in some cases, geometric ergodicity can be difficult to ascertain.

In the absence of theoretical convergence bounds, it is difficult to determine whether the chain has reached stationarity. One option is to independently run some large number K of chains, each with an initial state drawn from the same overdispersed starting distribution. If M and B are large enough, we expect the estimators provided by each chain to approximately agree. For

mathematical formalization of this general principle see, e.g., Gelman & Rubin (1992) and Brooks & Gelman (1998).

8.4. Convergence of Random Walk Metropolis for the Lupus Data

We now illustrate some of the above ideas using the RWM algorithm for the lupus data presented in Section 2.1. We consider running RWM for $M = 6{,}000$ iterations, using a burn-in of $B = 1{,}000$ iterations. We initialize the chain using draws from an overdispersed starting distribution centred at the MLE by setting $\beta_{\text{init}} = \hat{\beta}_{\text{MLE}} + W$, where W is a vector of three i.i.d. random variables, each of which is generated from a Student distribution with two degrees of freedom.

We repeated this entire experiment a total of $K = 350$ times with proposal variance-covariance matrix $\Sigma_1 = 0.6\,\mathbf{I}_3$ (**Figure 8**), and we performed it another $K = 350$ times with proposal variance-covariance matrix $\Sigma_2 = 1.2\,\mathbf{I}_3$ (**Figure 9**). The corresponding lists of estimates of the three β_i values illustrated in **Figures 8** and **9** show that despite the use of wide, overdispersed starting distributions, the resulting estimates are concentrated around particular values (boxplots, *top rows*; histograms, *bottom rows*), indicating fairly good convergence. They are also approximately normally distributed (normal Q-Q plots, *middle rows*; histograms, *bottom rows*), indicating approximate consistency with a CLT. Choosing larger values of M and B would likely result in even more concentrated values and more normal-looking distributions of the various estimates. This brief

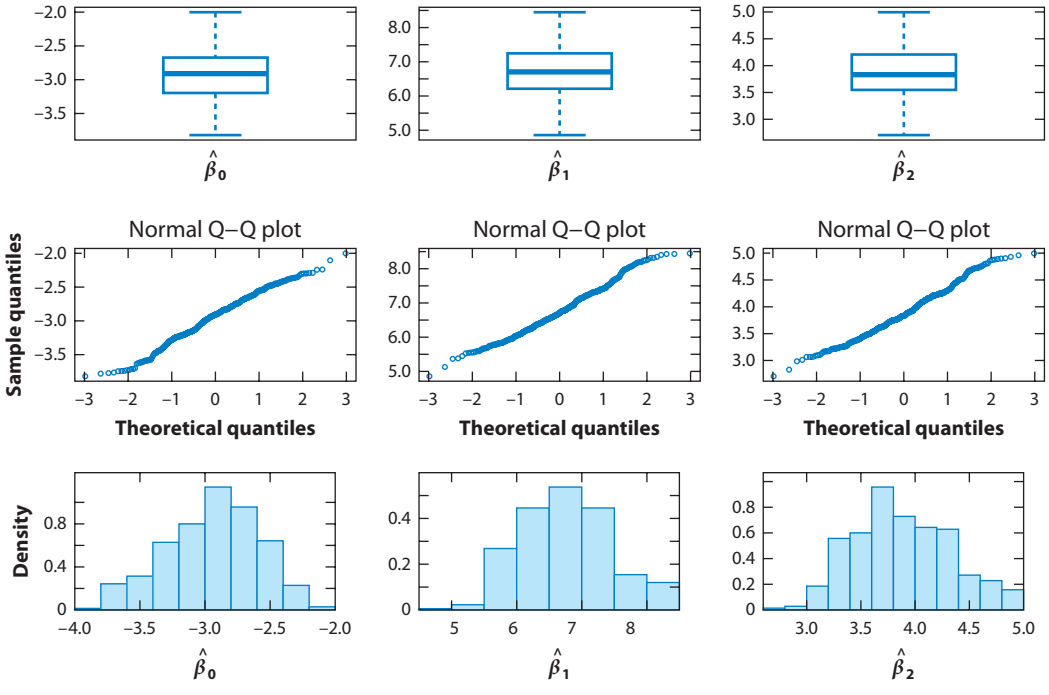

Figure 8

Results of $K = 350$ independent replications of a random walk Metropolis algorithm for the lupus data described in Section 2.1 with proposal variance-covariance matrix $\Sigma_1 = 0.6\,\mathbf{I}_3$. Resulting estimates of the quantities β_0 (*left column*), β_1 (*middle column*), and β_2 (*right column*) are shown in the respective boxplots (*top row*), normal Q-Q plots (*middle row*), and histograms (*bottom row*).

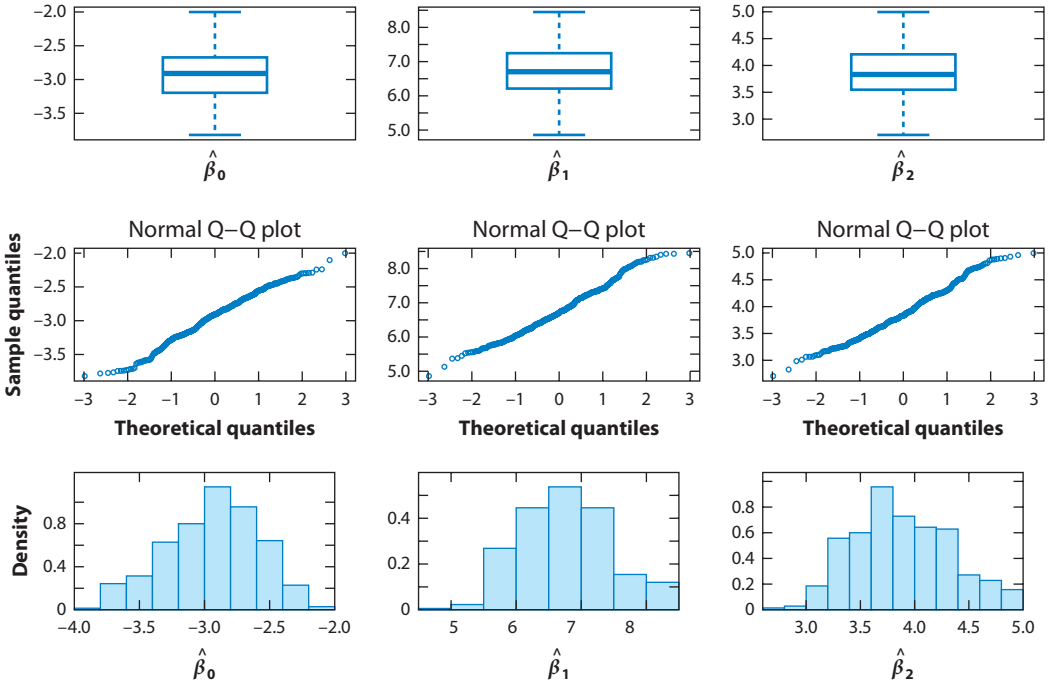

Figure 9

Results of $K = 350$ independent replications of a random walk Metropolis algorithm for the lupus data described in Section 2.1 with proposal variance-covariance matrix $\Sigma_2 = 1.2\,\mathbf{I}_3$. Resulting estimates of the quantities β_0 (*left column*), β_1 (*middle column*), and β_2 (*right column*) are shown in the resulting boxplots (*top row*), normal Q-Q plots (*middle row*), and histograms (*bottom row*).

experiment illustrates that even in the absence of theoretical convergence bounds, one can use multiple independent runs from overdispersed starting distributions to assess the convergence, accuracy, and normality of MCMC estimates.

8.5. The Case of the Independence Sampler

When it comes to MCMC convergence rates, one case is particularly tractable, namely the independence sampler. Unsurprisingly, as long as an independence sampler's proposal satisfies $q(\theta) > 0$ whenever $\pi(\theta) > 0$, irreducibility, aperiodicity, and stationarity all follow easily, and Theorem 1 therefore immediately establishes ergodicity. What is remarkable, however, is that the independence sampler is geometrically ergodic if and only if there is $\delta > 0$ such that $q(\theta) \geq \delta\pi(\theta)$ for π-a.e. $\theta \in \Theta$, and furthermore in this case $D(\zeta, n) \leq (1 - \delta)^n$ for π-a.e. $\zeta \in \Theta$ (Mengersen & Tweedie 1996, Roberts & Tweedie 1996). That is, for the independence sampler, we have not only an easy test for geometric ergodicity but also a free quantitative bound.

For a simple, specific example, consider an independence sampler on $\Theta = [0, \infty)$ with target density $\pi(\theta) = e^{-\theta}$. If the proposal density is, say, $q(\theta) = 0.01\,e^{-0.01\theta}$, then $q(\theta) \geq 0.01\,\pi(\theta)$ for all $\theta \in \Theta$. That is, the above condition for ergodicity holds when $\delta = 0.01$: The chain is geometrically ergodic with $D(\zeta, t) \leq (1-\delta)^t = (0.99)^t$ and hence converges in $t^* = 459$ iterations [because $(0.99)^{459} < 0.01$]. By contrast, if $q(\theta) = 5e^{-5\theta}$, then the above condition does not hold for any value $\delta > 0$. Thus, the chain is not geometrically ergodic. In fact, Rosenthal & Roberts (2011) showed that in this case $4{,}000{,}000 \leq t^* \leq 14{,}000{,}000$, i.e., the chain takes at least four

million iterations to converge. This example illustrates how geometric ergodicity can sometimes make a tremendous difference between MCMC algorithms that converge efficiently and those that converge very poorly. Moreover, it illustrates once again how MCMC methodology can help us explore target probability distributions and understand their statistical properties. The interplay between practical implementation and theoretical analysis involves many novel ideas with great potential for future development.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

We thank Nancy Reid for encouraging us to write this review, and we thank the anonymous referee for a very careful reading that led to many improvements.

LITERATURE CITED

Adler SL. 1981. Over-relaxation methods for the Monte Carlo evaluation of the partition function for multiquadratic actions. *Phys. Rev. D* 23:2901–4

Albert JH, Chib S. 1993. Bayesian analysis of binary and polychotomous response data. *J. Am. Stat. Assoc.* 88:669–79

Amit Y. 1991. On rates of convergence of stochastic relaxation for Gaussian and non-Gaussian distributions. *J. Multivar. Anal.* 38:82–100

Amit Y. 1996. Convergence properties of the Gibbs sampler for perturbations of Gaussians. *Ann. Stat.* 24:122–40

Andrieu C, Moulines E, Priouret P. 2005. Stability of stochastic approximation under verifiable conditions. *SIAM J. Control Optim.* 44:283–312

Bai Y, Craiu RV, DiNarzo AF. 2011. Divide and conquer: a mixture-based approach to regional adaptation for MCMC. *J. Comput. Graph. Stat.* 20:63–79

Barone P, Frigessi A. 1990. Improving stochastic relaxation for Gaussian random fields. *Probab. Eng. Inf. Sci.* 4:369–89

Bedard M. 2006. *On the robustness of optimal scaling for random walk Metropolis algorithms*. PhD Thesis, Department of Statistics, Univ. Toronto

Brooks S, Gelman A, Jones GL, Meng X-L, eds. 2011. *Handbook of Markov Chain Monte Carlo*. Boca Raton, FL: Chapman & Hall/CRC

Brooks SP, Gelman A. 1998. General methods for monitoring convergence of iterative simulations. *J. Comput. Graph. Stat.* 7:434–55

Casarin R, Craiu RV, Leisen F. 2013. Interacting multiple try algorithms with different proposal distributions. *Stat. Comput.* 23:185–200

Chen M-H, Shao Q-M, Ibrahim JG. 2000. *Monte Carlo Methods in Bayesian Computation*. New York: Springer

Craiu RV, Lemieux C. 2007. Acceleration of the multiple-try Metropolis algorithm using antithetic and stratified sampling. *Stat. Comput.* 17:109–20

Craiu RV, Meng X-L. 2005. Multi-process parallel antithetic coupling for forward and backward MCMC. *Ann. Stat.* 33:661–97

Craiu RV, Meng X-L. 2011. Perfection within reach: exact MCMC sampling. In *Handbook of Markov Chain Monte Carlo*, ed. S Brooks, A Gelman, GL Jones, X-L Meng, pp. 199–226. Boca Raton, FL: Chapman & Hall/CRC

Craiu RV, Rosenthal JS, Yang C. 2009. Learn from thy neighbor: parallel-chain adaptive and regional MCMC. *J. Am. Stat. Assoc.* 104:1454–66

Douc R, Moulines E, Rosenthal JS. 2004. Quantitative bounds on convergence of time-inhomogeneous Markov chains. *Ann. Appl. Probab.* 14:1643–65

Flegal JM, Haran M, Jones GL. 2008. Markov chain Monte Carlo: Can we trust the third significant figure? *Stat. Sci.* 23:250–60

Gelfand AE, Smith AFM. 1990. Sampling-based approaches to calculating marginal densities. *J. Am. Stat. Assoc.* 85:398–409

Gelman A, Rubin DB. 1992. Inference from iterative simulation using multiple sequences. *Stat. Sci.* 7:457–72

Geman S, Geman D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* 6:721–41

Geyer CJ. 1992. Practical Markov chain Monte Carlo (with discussion). *Stat. Sci.* 7:473–83

Green PJ. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82:711–32

Green PJ, Mira A. 2001. Delayed rejection in reversible jump Metropolis-Hastings. *Biometrika* 88:1035–53

Haario H, Saksman E, Tamminen J. 2001. An adaptive Metropolis algorithm. *Bernoulli* 7:223–42

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

Jones G, Hobert J. 2001. Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Stat. Sci.* 16:312–34

Liu JS. 2001. *Monte Carlo Strategies in Scientific Computing*. New York: Springer

Liu JS, Liang F, Wong WH. 2000. The multiple-try method and local optimization in Metropolis sampling. *J. Am. Stat. Assoc.* 95:121–34

Liu JS, Wong WH, Kong A. 1994. Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika* 81:27–40

Liu JS, Wong WH, Kong A. 1995. Covariance structure and convergence rate of the Gibbs sampler with various scans. *J. R. Stat. Soc. B* 57:157–69

Liu JS, Wu YN. 1999. Parameter expansion for data augmentation. *J. Am. Stat. Assoc.* 94:1264–74

Meng X-L, van Dyk DA. 1999. Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika* 86:301–20

Mengersen KL, Tweedie RL. 1996. Rates of convergence of the Hastings and Metropolis algorithms. *Ann. Stat.* 24:101–21

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. 1953. Equations of state calculations by fast computing machines. *J. Chem. Phys.* 21:1087–92

Meyn SP, Tweedie RL. 1993. *Markov Chains and Stochastic Stability*. London: Springer-Verlag

Neal RM. 1995. *Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation*. Tech. Rep. 9508, Univ. Toronto. Dep. Stat., Toronto, Can. **http://arxiv.org/pdf/bayes-an/9506004v1.pdf**

Papaspiliopoulos O, Roberts GO. 2008. Stability of the Gibbs sampler for Bayesian hierarchical models. *Ann. Stat.* 36:95–117

Propp JG, Wilson DB. 1996. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Struct. Algorithms* 9:223–52

Richardson S, Green PJ. 1997. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *J. R. Stat. Soc. B* 59:731–92

Robert CP, Casella G. 2004. *Monte Carlo Statistical Methods*. New York: Springer

Robert CP, Casella G. 2010. *Introducing Monte Carlo Methods with R*. New York: Springer

Roberts GO, Gelman A, Gilks WR. 1997. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Probab.* 7:110–20

Roberts GO, Rosenthal JS. 1997. Geometric ergodicity and hybrid Markov chains. *Electron. Commun. Probab.* 2(2):13–25

Roberts GO, Rosenthal JS. 2001. Optimal scaling for various Metropolis-Hastings algorithms. *Stat. Sci.* 16:351–67

Roberts GO, Rosenthal JS. 2004. General state space Markov chains and MCMC algorithms. *Probab. Surv.* 1:20–71

Roberts GO, Rosenthal JS. 2007. Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *J. Appl. Probab.* 44:458–75

Roberts GO, Rosenthal JS. 2009. Examples of adaptive MCMC. *J. Comput. Graph. Stat.* 18:349–67
Roberts GO, Tweedie RL. 1996. Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika* 83:95–110
Rosenthal JS. 1995. Minorization conditions and convergence rates for Markov chain Monte Carlo. *J. Am. Stat. Assoc.* 90:558–66
Rosenthal JS. 2001. A review of asymptotic convergence for general state space Markov chains. *Far East J. Theor. Stat.* 5:37–50
Rosenthal JS. 2002. Quantitative convergence rates of Markov chains: a simple account. *Electron. Commun. Probab.* 7:123–28
Rosenthal JS, Roberts GO. 2011. Quantitative non-geometric convergence bounds for independence samplers. *Methodol. Comput. Appl. Probab.* 13:391–403
Spiegelhalter DJ, Best NG, Carlin BP, Van Der Linde A. 2002. Bayesian measures of model complexity and fit (with discussion). *J. R. Stat. Soc. B* 64:583–639
Tanner MA, Wong WH. 1987. The calculation of posterior distributions by data augmentation. *J. Am. Stat. Assoc.* 82:528–40
Tierney L. 1994. Markov chains for exploring posterior distributions. *Ann. Stat.* 22:1701–28
van Dyk DA, Meng X-L. 2001. The art of data augmentation (with discussion). *J. Comput. Graph. Stat.* 10:1–111

Annual Review of Statistics and Its Application

Volume 1, 2014

Contents