Markov Chain Monte Carlo in Practice

Edited by

W.R. Gilks
Medical Research Council Biostatistics Unit
Cambridge
UK

S. Richardson
French National Institute for Health and Medical Research
Vilejuif
France

and

D.J. Spiegelhalter
Medical Research Council Biostatistics Unit
Cambridge
UK

CHAPMAN & HALL
London · Glasgow · Weinheim · New York · Tokyo · Melbourne · Madras

Published by Chapman & Hall, 2-6 Boundary Row, London SE1 8HN, UK

Chapman & Hall, 2-6 Boundary Row, London SE1 8HN, UK

Blackie Academic & Professional, Wester Cleddens Road, Bishopbriggs, Glasgow G64 2NZ, UK

Chapman & Hall GmbH, Pappelallee 3, 69469 Weinheim, Germany

Chapman & Hall USA., 115 Fifth Avenue, New York, NY 10003, USA

Chapman & Hall Japan, ITP-Japan, Kyowa Building, 3F, 2-2-1 Hirakawacho, Chiyoda-ku, Tokyo 102, Japan

Chapman & Hall Australia, 102 Dodds Street, South Melbourne, Victoria 3205, Australia

Chapman & Hall India, R. Seshadri, 32 Second Main Road, CIT East, Madras 600 035, India

First edition 1996

Printed in Great Britain by Clays Ltd., St. Ives Plc., Bungay, Suffolk

ISBN 0 412 05551 1

A Catalogue record for this book is available from the British Library

Contents

Contributors

James E Bennett	Department of Mathematics, Imperial College, London, UK.
Carlo Berzuini	Dipartimento di Informatica e Sistemistica, University of Pavia, Italy.
Nicola G Best	Medical Research Council Biostatistics Unit, Institute of Public Health, Cambridge, UK.
Caitlin Buck	School of History and Archaeology, University of Wales, Cardiff, UK.
Bradley P Carlin	Division of Biostatistics, School of Public Health, University of Minnesota, Minneapolis, USA.
David G Clayton	Medical Research Council Biostatistics Unit, Institute of Public Health, Cambridge, UK.
Jean Diebolt	Département de Statistique et Modèles Aléatoires, CNRS, Université Paris 6, France.
W James Gauderman	Department of Preventive Medicine, University of Southern California, Los Angeles, USA.
Alan E Gelfand	Department of Statistics, University of Connecticut, USA.
Andrew Gelman	Department of Statistics, University of California, Berkeley, California, USA.
Edward I George	MSIS Department, University of Texas at Austin, USA.

Charles J Geyer	School of Statistics, University of Minnesota, Minneapolis, USA.
Walter R Gilks	Medical Research Council Biostatistics Unit, Institute of Public Health, Cambridge, UK.
Peter J Green	Department of Mathematics, University of Bristol, UK.
Hazel Inskip	Medical Research Council Environmental Epidemiology Unit, Southampton, UK.
Eddie H S Ip	Educational Testing Service, Princeton, USA.
Steven M Lewis	School of Social Work, University of Washington, Seattle, USA.
Cliff Litton	Department of Mathematics, University of Nottingham, UK.
Robert E McCulloch	Graduate School of Business, University of Chicago, USA.
Xiao-Li Meng	Department of Statistics, University of Chicago, USA.
Annie Mollié	INSERM Unité 351, Villejuif, France.
David B Phillips	NatWest Markets, London, UK.
Amy Racine-Poon	Ciba Geigy, Basle, Switzerland.
Adrian E Raftery	Department of Statistics, University of Washington, Seattle, USA.
Sylvia Richardson	INSERM Unité 170, Villejuif, France.
Christian P Robert	Laboratoire de Statistique, University of Rouen, France.
Gareth O Roberts	Statistical Laboratory, University of Cambridge, UK.
Adrian F M Smith	Department of Mathematics, Imperial College, London, UK.
David J Spiegelhalter	Medical Research Council Biostatistics Unit, Institute of Public Health, Cambridge, UK.

Duncan C Thomas	Department of Preventive Medicine, University of Southern California, Los Angeles, USA.
Luke Tierney	School of Statistics, University of Minnesota, Minneapolis, USA.
Jon C Wakefield	Department of Mathematics, Imperial College, London, UK.

1

Introducing Markov chain Monte Carlo

Walter R Gilks
Sylvia Richardson
David J Spiegelhalter

1.1 Introduction

Markov chain Monte Carlo (MCMC) methodology provides enormous scope for realistic statistical modelling. Until recently, acknowledging the full complexity and structure in many applications was difficult and required the development of specific methodology and purpose-built software. The alternative was to coerce the problem into the over-simple framework of an available method. Now, MCMC methods provide a unifying framework within which many complex problems can be analysed using generic software.

MCMC is essentially Monte Carlo integration using Markov chains. Bayesians, and sometimes also frequentists, need to integrate over possibly high-dimensional probability distributions to make inference about model parameters or to make predictions. Bayesians need to integrate over the posterior distribution of model parameters given the data, and frequentists may need to integrate over the distribution of observables given parameter values. As described below, *Monte Carlo* integration draws samples from the the required distribution, and then forms sample averages to approximate expectations. *Markov chain* Monte Carlo draws these samples by running a cleverly constructed Markov chain for a long time. There are many ways of constructing these chains, but all of them, including the Gibbs sampler (Geman and Geman, 1984), are special cases of the general framework of Metropolis *et al.* (1953) and Hastings (1970).

It took nearly 40 years for MCMC to penetrate mainstream statistical practice. It originated in the statistical physics literature, and has been used for a decade in spatial statistics and image analysis. In the last few years, MCMC has had a profound effect on Bayesian statistics, and has also found applications in classical statistics. Recent research has added considerably to its breadth of application, the richness of its methodology, and its theoretical underpinnings.

The purpose of this book is to introduce MCMC methods and their applications, and to provide pointers to the literature for further details. Having in mind principally an applied readership, our role as editors has been to keep the technical content of the book to a minimum and to concentrate on methods which have been shown to help in real applications. However, some theoretical background is also provided. The applications featured in this volume draw from a wide range of statistical practice, but to some extent reflect our own biostatistical bias. The chapters have been written by researchers who have made key contributions in the recent development of MCMC methodology and its application. Regrettably, we were not able to include all leading researchers in our list of contributors, nor were we able to cover all areas of theory, methods and application in the depth they deserve.

Our aim has been to keep each chapter self-contained, including notation and references, although chapters may assume knowledge of the basics described in this chapter. This chapter contains enough information to allow the reader to start applying MCMC in a basic way. In it we describe the Metropolis–Hastings algorithm, the Gibbs sampler, and the main issues arising in implementing MCMC methods. We also give a brief introduction to Bayesian inference, since many of the following chapters assume a basic knowledge. Chapter 2 illustrates many of the main issues in a worked example. Chapters 3 and 4 give an introduction to important concepts and results in discrete and general state-space Markov chain theory. Chapters 5 through 8 give more information on techniques for implementing MCMC or improving its performance. Chapters 9 through 13 describe methods for assessing model adequacy and choosing between models, using MCMC. Chapters 14 and 15 describe MCMC methods for non-Bayesian inference, and Chapters 16 through 25 describe applications or summarize application domains.

1.2 The problem

1.2.1 Bayesian inference

Most applications of MCMC to date, including the majority of those described in the following chapters, are oriented towards Bayesian inference. From a Bayesian perspective, there is no fundamental distinction between

observables and parameters of a statistical model: all are considered random quantities. Let D denote the observed data, and θ denote model parameters and missing data. Formal inference then requires setting up a joint probability distribution $P(D, \theta)$ over all random quantities. This joint distribution comprises two parts: a *prior* distribution $P(\theta)$ and a *likelihood* $P(D|\theta)$. Specifying $P(\theta)$ and $P(D|\theta)$ gives a *full probability model*, in which

$$P(D, \theta) = P(D|\theta)\, P(\theta).$$

Having observed D, Bayes theorem is used to determine the distribution of θ conditional on D:

$$P(\theta|D) = \frac{P(\theta)P(D|\theta)}{\int P(\theta)P(D|\theta)d\theta}.$$

This is called the *posterior* distribution of θ, and is the object of all Bayesian inference.

Any features of the posterior distribution are legitimate for Bayesian inference: moments, quantiles, highest posterior density regions, etc. All these quantities can be expressed in terms of posterior expectations of functions of θ. The posterior expectation of a function $f(\theta)$ is

$$E[f(\theta)|D] = \frac{\int f(\theta)P(\theta)P(D|\theta)d\theta}{\int P(\theta)P(D|\theta)d\theta}.$$

The integrations in this expression have until recently been the source of most of the practical difficulties in Bayesian inference, especially in high dimensions. In most applications, analytic evaluation of $E[f(\theta)|D]$ is impossible. Alternative approaches include numerical evaluation, which is difficult and inaccurate in greater than about 20 dimensions; analytic approximation such as the Laplace approximation (Kass *et al.*, 1988), which is sometimes appropriate; and Monte Carlo integration, including MCMC.

1.2.2 Calculating expectations

The problem of calculating expectations in high-dimensional distributions also occurs in some areas of frequentist inference; see Geyer (1995) and Diebolt and Ip (1995) in this volume. To avoid an unnecessarily Bayesian flavour in the following discussion, we restate the problem in more general terms. Let X be a vector of k random variables, with distribution $\pi(.)$. In Bayesian applications, X will comprise model parameters and missing data; in frequentist applications, it may comprise data or random effects. For Bayesians, $\pi(.)$ will be a posterior distribution, and for frequentists it will be a likelihood. Either way, the task is to evaluate the expectation

$$E[f(X)] = \frac{\int f(x)\pi(x)dx}{\int \pi(x)dx} \tag{1.1}$$

for some function of interest $f(.)$. Here we allow for the possibility that the distribution of X is known only up to a constant of normalization. That is, $\int \pi(x)dx$ is unknown. This is a common situation in practice, for example in Bayesian inference we know $P(\theta|D) \propto P(\theta)P(D|\theta)$, but we cannot easily evaluate the normalization constant $\int P(\theta)P(D|\theta)d\theta$. For simplicity, we assume that X takes values in k-dimensional Euclidean space, i.e. that X comprises k continuous random variables. However, the methods described here are quite general. For example, X could consist of discrete random variables, so then the integrals in (1.1) would be replaced by summations. Alternatively, X could be a mixture of discrete and continuous random variables, or indeed a collection of random variables on any probability space. Indeed, k can itself be variable: see Section 1.3.3. Measure theoretic notation in (1.1) would of course concisely accommodate all these possibilities, but the essential message can be expressed without it. We use the terms *distribution* and *density* interchangeably.

1.3 Markov chain Monte Carlo

In this section, we introduce MCMC as a method for evaluating expressions of the form of (1.1). We begin by describing its constituent parts: Monte Carlo integration and Markov chains. We then describe the general form of MCMC given by the Metropolis–Hastings algorithm, and a special case: the Gibbs sampler.

1.3.1 Monte Carlo integration

Monte Carlo integration evaluates $E[f(X)]$ by drawing samples $\{X_t, t = 1, \ldots, n\}$ from $\pi(.)$ and then approximating

$$E[f(X)] \approx \frac{1}{n}\sum_{t=1}^{n} f(X_t).$$

So the population mean of $f(X)$ is estimated by a sample mean. When the samples $\{X_t\}$ are independent, laws of large numbers ensure that the approximation can be made as accurate as desired by increasing the sample size n. Note that here n is under the control of the analyst: it is not the size of a fixed data sample.

In general, drawing samples $\{X_t\}$ independently from $\pi(.)$ is not feasible, since $\pi(.)$ can be quite non-standard. However the $\{X_t\}$ need not necessarily be independent. The $\{X_t\}$ can be generated by any process which, loosely speaking, draws samples throughout the support of $\pi(.)$ in the correct proportions. One way of doing this is through a Markov chain having $\pi(.)$ as its stationary distribution. This is then *Markov chain* Monte Carlo.

1.3.2 Markov chains

Suppose we generate a sequence of random variables, $\{X_0, X_1, X_2, \ldots\}$, such that at each time $t \geq 0$, the next state X_{t+1} is sampled from a distribution $P(X_{t+1}|X_t)$ which depends only on the current state of the chain, X_t. That is, *given* X_t, the next state X_{t+1} does not depend further on the history of the chain $\{X_0, X_1, \ldots, X_{t-1}\}$. This sequence is called a *Markov chain*, and $P(.|.)$ is called the *transition kernel* of the chain. We will assume that the chain is time-homogenous: that is, $P(.|.)$ does not depend on t.

How does the starting state X_0 affect X_t? This question concerns the distribution of X_t given X_0, which we denote $P^{(t)}(X_t|X_0)$. Here we are not given the intervening variables $\{X_1, X_2, \ldots, X_{t-1}\}$, so X_t depends directly on X_0. Subject to regularity conditions, the chain will gradually 'forget' its initial state and $P^{(t)}(.|X_0)$ will eventually converge to a unique *stationary* (or *invariant*) distribution, which does not depend on t or X_0. For the moment, we denote the stationary distribution by $\phi(.)$. Thus as t increases, the sampled points $\{X_t\}$ will look increasingly like dependent samples from $\phi(.)$. This is illustrated in Figure 1.1, where $\phi(.)$ is univariate standard normal. Note that convergence is much quicker in Figure 1.1(a) than in Figures 1.1(b) or 1.1(c).

Thus, after a sufficiently long *burn-in* of say m iterations, points $\{X_t; t = m+1, \ldots, n\}$ will be dependent samples approximately from $\phi(.)$. We discuss methods for determining m in Section 1.4.6. We can now use the output from the Markov chain to estimate the expectation $E[f(X)]$, where X has distribution $\phi(.)$. Burn-in samples are usually discarded for this calculation, giving an estimator

$$\bar{f} = \frac{1}{n-m}\sum_{t=m+1}^{n} f(X_t). \tag{1.2}$$

This is called an *ergodic average*. Convergence to the required expectation is ensured by the ergodic theorem.

See Roberts (1995) and Tierney (1995) in this volume for more technical discussion of several of the issues raised here.

1.3.3 The Metropolis–Hastings algorithm

Equation (1.2) shows how a Markov chain can be used to estimate $E[f(X)]$, where the expectation is taken over its stationary distribution $\phi(.)$. This would seem to provide the solution to our problem, but first we need to discover how to construct a Markov chain such that its stationary distribution $\phi(.)$ is precisely our distribution of interest $\pi(.)$.

Constructing such a Markov chain is surprisingly easy. We describe the form due to Hastings (1970), which is a generalization of the method

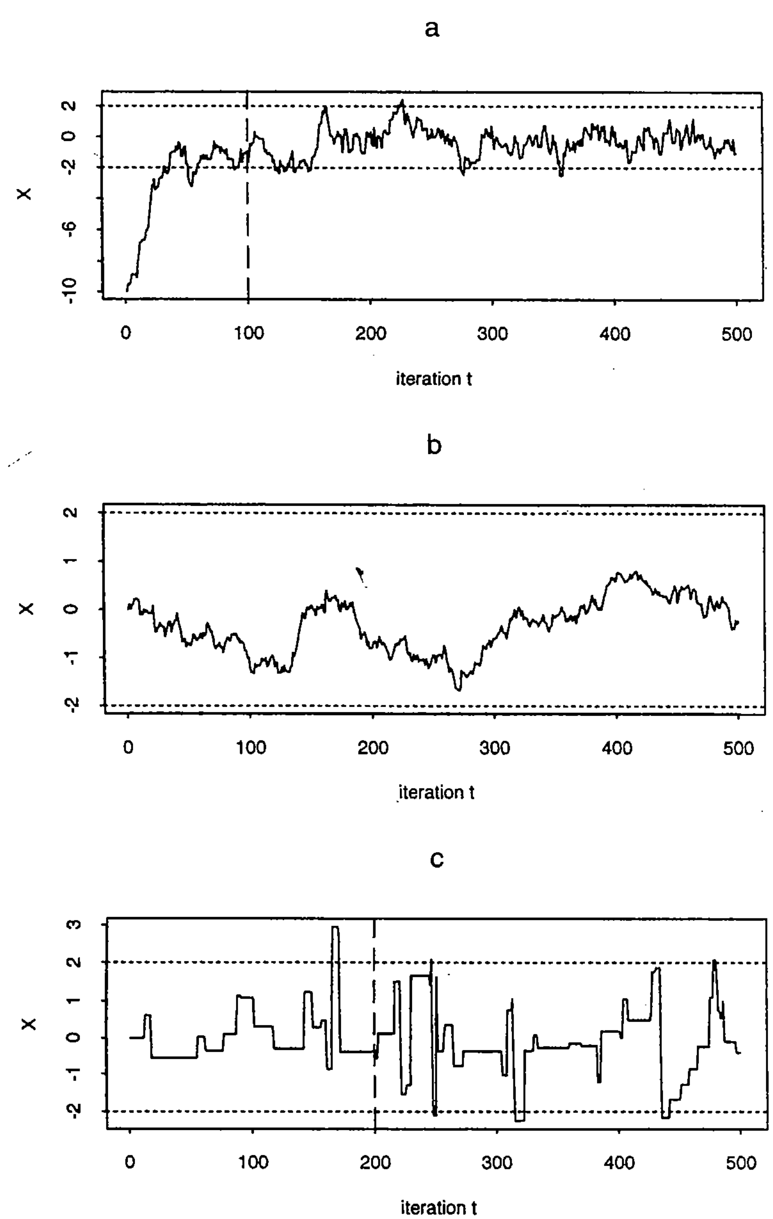

Figure 1.1 *500 iterations from Metropolis algorithms with stationary distribution* N(0, 1) *and proposal distributions (a)* $q(.|X) = \mathrm{N}(X, 0.5)$; *(b)* $q(.|X) = \mathrm{N}(X, 0.1)$; *and (c)* $q(.|X) = \mathrm{N}(X, 10.0)$. *The burn-in is taken to be to the left of the vertical broken line.*

first proposed by Metropolis *et al.* (1953). For the *Metropolis–Hastings* (or *Hastings–Metropolis*) algorithm, at each time t, the next state X_{t+1} is chosen by first sampling a *candidate* point Y from a *proposal* distribution $q(.|X_t)$. Note that the proposal distribution may depend on the current point X_t. For example, $q(.|X)$ might be a multivariate normal distribution with mean X and a fixed covariance matrix. The candidate point Y is then *accepted* with probability $\alpha(X_t, Y)$ where

$$\alpha(X,Y) = \min\left(1, \frac{\pi(Y)q(X|Y)}{\pi(X)q(Y|X)}\right) \tag{1.3}$$

If the candidate point is accepted, the next state becomes $X_{t+1} = Y$. If the candidate is rejected, the chain does not move, i.e. $X_{t+1} = X_t$. Figure 1.1 illustrates this for univariate normal proposal and target distributions; Figure 1.1(c) showing many instances where the chain did not move for several iterations.

Thus the Metropolis–Hastings algorithm is extremely simple:

```
Initialize X_0; set t = 0.
Repeat {
  Sample a point Y from q(.|X_t)
  Sample a Uniform(0,1) random variable U
  If U ≤ α(X_t, Y) set X_{t+1} = Y
    otherwise set X_{t+1} = X_t
  Increment t
}.
```

Remarkably, the proposal distribution $q(.|.)$ can have any form and the stationary distribution of the chain will be $\pi(.)$. (For regularity conditions see Roberts, 1995: this volume.) This can be seen from the following argument. The transition kernel for the Metropolis–Hastings algorithm is

$$\begin{aligned} P(X_{t+1}|X_t) &= q(X_{t+1}|X_t)\alpha(X_t, X_{t+1}) \\ &\quad + I(X_{t+1} = X_t)[1 - \int q(Y|X_t)\alpha(X_t, Y)dY], \end{aligned} \tag{1.4}$$

where $I(.)$ denotes the indicator function (taking the value 1 when its argument is true, and 0 otherwise). The first term in (1.4) arises from acceptance of a candidate $Y = X_{t+1}$, and the second term arises from rejection, for all possible candidates Y. Using the fact that

$$\pi(X_t)q(X_{t+1}|X_t)\alpha(X_t, X_{t+1}) = \pi(X_{t+1})q(X_t|X_{t+1})\alpha(X_{t+1}, X_t)$$

which follows from (1.3), we obtain the *detailed balance* equation:

$$\pi(X_t)P(X_{t+1}|X_t) = \pi(X_{t+1})P(X_t|X_{t+1}). \tag{1.5}$$

Integrating both sides of (1.5) with respect to X_t gives:

$$\int \pi(X_t)P(X_{t+1}|X_t)dX_t = \pi(X_{t+1}). \tag{1.6}$$

The left-hand side of equation (1.6) gives the marginal distribution of X_{t+1} under the assumption that X_t is from $\pi(.)$. Therefore (1.6) says that if X_t is from $\pi(.)$, then X_{t+1} will be also. Thus, once a sample from the stationary distribution has been obtained, all subsequent samples will be from that distribution. This only proves that the stationary distribution is $\pi(.)$, and is not a complete justification for the Metropolis–Hastings algorithm. A full justification requires a proof that $P^{(t)}(X_t|X_0)$ will converge to the stationary distribution. See Roberts (1995) and Tierney (1995) in this volume for further details.

So far we have assumed that X is a fixed-length vector of k continuous random variables. As noted in Section 1.2, there are many other possibilities, in particular X can be of *variable dimension*. For example, in a Bayesian mixture model, the number of mixture components may be variable: each component possessing its own scale and location parameters. In this situation, $\pi(.)$ must specify the joint distribution of k and X, and $q(Y|X)$ must be able to propose moves between spaces of differing dimension. Then Metropolis–Hastings is as described above, with formally the same expression (1.3) for the acceptance probability, but where dimension-matching conditions for moves between spaces of differing dimension must be carefully considered (Green, 1994a,b). See also Geyer and Møller (1993), Grenander and Miller (1994), and Phillips and Smith (1995: this volume) for MCMC methodology in variably dimensioned problems.

1.4 Implementation

There are several issues which arise when implementing MCMC. We discuss these briefly here. Further details can be found throughout this volume, and in particular in Chapters 5–8. The most immediate issue is the choice of proposal distribution $q(.|.)$.

1.4.1 Canonical forms of proposal distribution

As already noted, any proposal distribution will ultimately deliver samples from the target distribution $\pi(.)$. However, the rate of convergence to the stationary distribution will depend crucially on the relationship between $q(.|.)$ and $\pi(.)$. Moreover, having 'converged', the chain may still *mix* slowly (i.e. move slowly around the support of $\pi(.)$). These phenomena are illustrated in Figure 1.1. Figure 1.1(a) shows rapid convergence from a somewhat extreme starting value: thereafter the chain mixes rapidly. Figure 1.1(b),(c) shows slow mixing chains: these would have to be run much longer to obtain reliable estimates from (1.2), despite having been started at the mode of $\pi(.)$.

In high-dimensional problems with little symmetry, it is often necessary to perform exploratory analyses to determine roughly the shape and orientation of $\pi(.)$. This will help in constructing a proposal $q(.|.)$ which leads to rapid mixing. Progress in practice often depends on experimentation and craftmanship, although untuned canonical forms for $q(.|.)$ often work surprisingly well. For computational efficiency, $q(.|.)$ should be chosen so that it can be easily sampled and evaluated.

Here we describe some canonical forms for $q(.|.)$. Roberts (1995), Tierney (1995) and Gilks and Roberts (1995) in this volume discuss rates of convergence and strategies for choosing $q(.|.)$ in more detail.

The Metropolis Algorithm

The *Metropolis algorithm* (Metropolis *et al.*, 1953) considers only symmetric proposals, having the form $q(Y|X) = q(X|Y)$ for all X and Y. For example, when X is continuous, $q(.|X)$ might be a multivariate normal distribution with mean X and constant covariance matrix Σ. Often it is convenient to choose a proposal which generates each component of Y conditionally independently, given X_t. For the Metropolis algorithm, the acceptance probability (1.3) reduces to

$$\alpha(X,Y) = \min\left(1, \frac{\pi(Y)}{\pi(X)}\right) \tag{1.7}$$

A special case of the Metropolis algorithm is *random-walk Metropolis*, for which $q(Y|X) = q(|X - Y|)$. The data in Figure 1.1 were generated by random-walk Metropolis algorithms.

When choosing a proposal distribution, its scale (for example Σ) may need to be chosen carefully. A cautious proposal distribution generating small steps $Y - X_t$ will generally have a high acceptance rate (1.7), but will nevertheless mix slowly. This is illustrated in Figure 1.1(b). A bold proposal distribution generating large steps will often propose moves from the body to the tails of the distribution, giving small values of $\pi(Y)/\pi(X_t)$ and a low probability of acceptance. Such a chain will frequently not move, again resulting in slow mixing as illustrated in Figure 1.1(c). Ideally, the proposal distribution should be scaled to avoid both these extremes.

The independence sampler

The *independence sampler* (Tierney, 1994) is a Metropolis–Hastings algorithm whose proposal $q(Y|X) = q(Y)$ does not depend on X. For this, the acceptance probability (1.3) can be written in the form

$$\alpha(X,Y) = \min\left(1, \frac{w(Y)}{w(X)}\right), \tag{1.8}$$

where $w(X) = \pi(X)/q(X)$.

In general, the independence sampler can work very well or very badly (see Roberts, 1995: this volume). For the independence sampler to work well, $q(.)$ should be a good approximation to $\pi(.)$, but it is safest if $q(.)$ is heavier-tailed than $\pi(.)$. To see this, suppose $q(.)$ is lighter-tailed than $\pi(.)$, and that X_t is currently in the tails of $\pi(.)$. Most candidates will not be in the tails, so $w(X_t)$ will be much larger than $w(Y)$ giving a low acceptance probability (1.8). Thus heavy-tailed independence proposals help to avoid long periods stuck in the tails, at the expense of an increased overall rate of candidate rejection.

In some situations, in particular where it is thought that large-sample theory might be operating, a multivariate normal proposal might be tried, with mean at the mode of $\pi(.)$ and covariance matrix somewhat greater than the inverse Hessian matrix

$$\left[-\frac{d^2 \log \pi(x)}{dx^{\mathrm{T}} dx}\right]^{-1}$$

evaluated at the mode.

Single-component Metropolis–Hastings

Instead of updating the whole of X *en bloc*, it is often more convenient and
computationally efficient to divide X into components $\{X_{.1}, X_{.2}, \ldots, X_{.h}\}$
of possibly differing dimension, and then update these components one by
one. This was the framework for MCMC originally proposed by Metropolis
et al. (1953), and we refer to it as *single-component Metropolis–Hastings.*
Let $X_{.-i} = \{X_{.1}, \ldots, X_{.i-1}, X_{.i+1}, \ldots, X_{.h}\}$, so $X_{.-i}$ comprises all of X
except $X_{.i}$.

An iteration of the single-component Metropolis–Hastings algorithm com-
prises h updating steps, as follows. Let $X_{t.i}$ denote the state of $X_{.i}$ at
the end of iteration t. For step i of iteration $t+1$, $X_{.i}$ is updated using
Metropolis–Hastings. The candidate $Y_{.i}$ is generated from a proposal dis-
tribution $q_i(Y_{.i}|X_{t.i}, X_{t.-i})$, where $X_{t.-i}$ denotes the value of $X_{.-i}$ after
completing step $i-1$ of iteration $t+1$:

$$X_{t.-i} = \{X_{t+1.1}, \ldots, X_{t+1.i-1}, X_{t.i+1}, \ldots, X_{t.h}\},$$

where components $1, 2, \ldots, i-1$ have already been updated. Thus the i^{th}
proposal distribution $q_i(.|.,.)$ generates a candidate only for the i^{th} compon-
ent of X, and may depend on the *current* values of any of the components
of X. The candidate is accepted with probability $\alpha(X_{t.-i}, X_{t.i}, Y_{.i})$ where

$$\alpha(X_{.-i}, X_{.i}, Y_{.i}) = \min\left(1, \frac{\pi(Y_{.i}|X_{.-i})q_i(X_{.i}|Y_{.i}, X_{.-i})}{\pi(X_{.i}|X_{.-i})q_i(Y_{.i}|X_{.i}, X_{.-i})}\right). \quad (1.9)$$

Here $\pi(X_{.i}|X_{.-i})$ is the *full conditional* distribution for $X_{.i}$ under $\pi(.)$ (see below). If $Y_{.i}$ is accepted, we set $X_{t+1.i} = Y_{.i}$; otherwise, we set $X_{t+1.i} = X_{t.i}$. The remaining components are not changed at step i.

Thus each updating step produces a move in the direction of a coordinate axis (if the candidate is accepted), as illustrated in Figure 1.2. The proposal

distribution $q_i(.|.,.)$ can be chosen in any of the ways discussed earlier in this section.

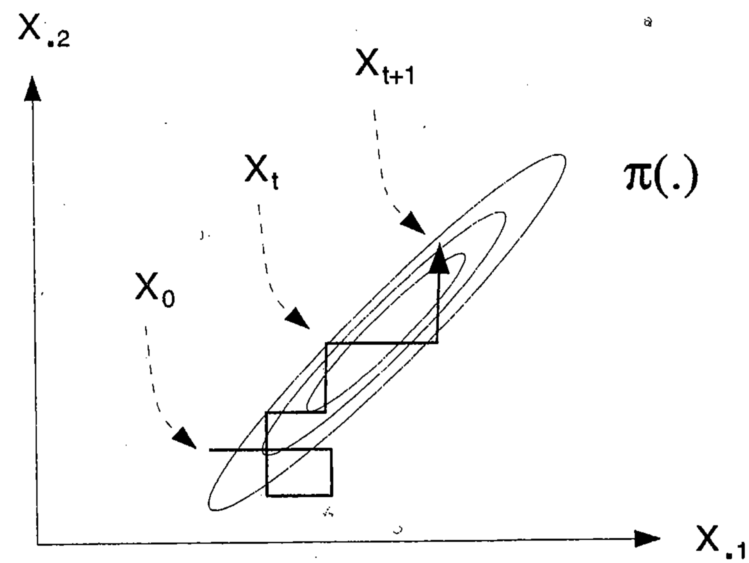

Figure 1.2 *Illustrating a single-component Metropolis–Hastings algorithm for a bivariate target distribution $\pi(.)$. Components 1 and 2 are updated alternately, producing alternate moves in horizontal and vertical directions.*

The full conditional distribution $\pi(X_{.i}|X_{.-i})$ is the distribution of the i^{th} component of X conditioning on all the remaining components, where X has distribution $\pi(.)$:

$$\pi(X_{.i}|X_{.-i}) = \frac{\pi(X)}{\int \pi(X)dX_{.i}}. \quad (1.10)$$

Full conditional distributions play a prominent role in many of the applications in this volume, and are considered in detail by Gilks (1995: this volume). That the single-component Metropolis–Hastings algorithm with acceptance probability given by (1.9) does indeed generate samples from the target distribution $\pi(.)$ results from the fact that $\pi(.)$ is uniquely determined by the set of its full conditional distributions (Besag, 1974).

In applications, (1.9) often simplifies considerably, particularly when $\pi(.)$ derives from a conditional independence model: see Spiegelhalter *et al.* (1995) and Gilks (1995) in this volume. This provides an important computational advantage. Another important advantage of single-component updating occurs when the target distribution $\pi(.)$ is naturally specified in terms of its full conditional distributions, as commonly occurs in spatial models; see Besag (1974), Besag *et al.* (1995) and Green (1995: this volume).

Gibbs sampling

A special case of single-component Metropolis–Hastings is the *Gibbs sampler*. The Gibbs sampler was given its name by Geman and Geman (1984), who used it for analysing Gibbs distributions on lattices. However, its applicability is not limited to Gibbs distributions, so 'Gibbs sampling' is really a misnoma. Moreover, the same method was already in use in statistical physics, and was known there as the *heat bath algorithm*. Nevertheless, the work of Geman and Geman (1984) led to the introduction of MCMC into mainstream statistics via the articles by Gelfand and Smith (1990) and Gelfand *et al.* (1990). To date, most statistical applications of MCMC have used Gibbs sampling.

For the Gibbs sampler, the proposal distribution for updating the i^{th} component of X is

$$q_i(Y_{.i}|X_{.i}, X_{.-i}) = \pi(Y_{.i}|X_{.-i}) \tag{1.11}$$

where $\pi(Y_{.i}|X_{.-i})$ is the full conditional distribution (1.10). Substituting (1.11) into (1.9) gives an acceptance probability of 1; that is, Gibbs sampler candidates are always accepted. Thus Gibbs sampling consists purely in sampling from full conditional distributions. Methods for sampling from full conditional distributions are described in Gilks (1995: this volume).

1.4.2 Blocking

Our description of single-component samplers in Section 1.4.1 said nothing about how the components should be chosen. Typically, low-dimensional or scalar components are used. In some situations, multivariate components are natural. For example, in a Bayesian random-effects model, an entire precision matrix would usually comprise a single component. When components are highly correlated in the stationary distribution $\pi(.)$, mixing can be slow; see Gilks and Roberts (1995: this volume). Blocking highly correlated components into a higher-dimensional component may improve mixing, but this depends on the choice of proposal.

1.4.3 Updating order

In the above description of the single-component Metropolis–Hastings algorithm and Gibbs sampling, we assumed a fixed updating order for the components of X_t. Although this is usual, a fixed order is not necessary: random permutations of the updating order are quite acceptable. Moreover, not all components need be updated in each iteration. For example, we could instead update only one component per iteration, selecting component i with some fixed probability $s(i)$. A natural choice would be to set $s(i) = \frac{1}{h}$. Zeger and Karim (1991) suggest updating highly correlated components more frequently than other components, to improve mixing. Note that if $s(i)$ is allowed to depend on X_t then the acceptance probability (1.9) should be modified, otherwise the stationary distribution of the chain may no longer be the target distribution $\pi(.)$. Specifically, the acceptance probability becomes

$$\min\left(1, \frac{\pi(Y_{.i}|X_{.-i})s(i|Y_{.i}, X_{.-i})q_i(X_{.i}|Y_{.i}, X_{.-i})}{\pi(X_{.i}|X_{.-i})s(i|X_{.i}, X_{.-i})q_i(Y_{.i}|X_{.i}, X_{.-i})}\right).$$

1.4.4 Number of chains

So far we have considered running only one chain, but multiple chains are permissible. Recommendations in the literature have been conflicting, ranging from many short chains (Gelfand and Smith, 1990), to several long ones (Gelman and Rubin, 1992a,b), to one very long one (Geyer, 1992). It is now generally agreed that running many short chains, motivated by a desire to obtain independent samples from $\pi(.)$, is misguided unless there is some special reason for needing independent samples. Certainly, independent samples are not required for ergodic averaging in (1.2). The debate between the several-long-runs school and the one-very-long-run school seems set to continue. The latter maintains that one very long run has the best chance of finding new modes, and comparison between chains can never prove convergence, whilst the former maintains that comparing several seemingly converged chains might reveal genuine differences if the chains have not yet approached stationarity; see Gelman (1995: this volume). If several processors are available, running one chain on each will generally be worthwhile.

1.4.5 Starting values

Not much has been written on this topic. If the chain is irreducible, the choice of starting values X_0 will not affect the stationary distribution. A rapidly mixing chain, such as in Figure 1.1(a), will quickly find its way from extreme starting values. Starting values may need to be chosen more carefully for slow-mixing chains, to avoid a lengthy burn-in. However, it is seldom necessary to expend much effort in choosing starting values. Gelman and Rubin (1992a,b) suggest using 'over-dispersed' starting values in multiple chains, to assist in assessing convergence; see below and Gelman (1995: this volume).

1.4.6 Determining burn-in

The length of burn-in m depends on X_0, on the rate of convergence of $P^{(t)}(X_t|X_0)$ to $\pi(X_t)$ and on how similar $P^{(t)}(.|.)$ and $\pi(.)$ are required to be. Theoretically, having specified a criterion of 'similar enough', m can be determined analytically. However, this calculation is far from computationally feasible in most situations (see Roberts, 1995: this volume). Visual inspection of plots of (functions of) the Monte-Carlo output $\{X_t, t = 1, \ldots, n\}$ is the most obvious and commonly used method for determining burn-in, as in Figure 1.1. Starting the chain close to the mode of $\pi(.)$ does not remove the need for a burn-in, as the chain should still be run long enough for it to 'forget' its starting position. For example, in Figure 1.1(b) the chain has not wandered far from its starting position in 500 iterations. In this case, m should be set greater than 500.

More formal tools for determining m, called *convergence diagnostics*, have been proposed. Convergence diagnostics use a variety of theoretical methods and approximations, but all make use of the Monte Carlo output in some way. By now, at least 10 convergence diagnostics have been proposed; for a recent review, see Cowles and Carlin (1994). Some of these diagnostics are also suited to determining run length n (see below).

Convergence diagnostics can be classified by whether or not they are based on an arbitrary function $f(X)$ of the Monte Carlo output; whether they use output from a single chain or from multiple chains; and whether they can be based purely on the Monte Carlo output.

Methods which rely on monitoring $\{f(X_t), t = 1, \ldots, n\}$ (e.g. Gelman and Rubin, 1992b; Raftery and Lewis, 1992; Geweke, 1992) are easy to apply, but may be misleading since $f(X_t)$ may appear to have converged in distribution by iteration m, whilst another unmonitored function $g(X_t)$ may not have. Whatever functions $f(.)$ are monitored, there may be others which behave differently.

From a theoretical perspective, it is better to compare globally the full joint distribution $P^{(t)}(.)$ with $\pi(.)$. To avoid having to deal with $P^{(t)}(.)$ directly, several methods obtain samples from it by running multiple parallel chains (Ritter and Tanner, 1992; Roberts, 1992; Liu and Liu, 1993), and make use of the transition kernel $P(.|.)$. However, for stability in the procedures, it may be necessary to run many parallel chains. When convergence is slow, this is a serious practical limitation.

Running parallel chains obviously increases the computational burden, but can be useful, even informally, to diagnose slow convergence. For example, several parallel chains might individually appear to have converged, but comparisons between them may reveal marked differences in the apparent stationary distributions (Gelman and Rubin, 1992a).

From a practical perspective, methods which are based purely on the Monte Carlo output are particularly convenient, allowing assessment of convergence without recourse to the transition kernel $P(.|.)$, and hence without model-specific coding.

This volume does not contain a review of convergence diagnostics. This is still an active area of research, and much remains to be learnt about the behaviour of existing methods in real applications, particularly in high dimensions and when convergence is slow. Instead, the chapters by Raftery and Lewis (1995) and Gelman (1995) in this volume contain descriptions of two of the most popular methods. Both methods monitor an arbitrary function $f(.)$, and are based purely on the Monte Carlo output. The former uses a single chain and the latter multiple chains.

Geyer (1992) suggests that calculation of the length of burn-in is unnecessary, as it is likely to be less than 1% of the total length of a run sufficiently long to obtain adequate precision in the estimator $\overline{f}$ in (1.2), (see below). If extreme starting values are avoided, Geyer suggests setting m to between 1% and 2% of the run length n.

1.4.7 Determining stopping time

Deciding when to stop the chain is an important practical matter. The aim is to run the chain long enough to obtain adequate precision in the estimator $\overline{f}$ in (1.2). Estimation of the variance of $\overline{f}$ (called the *Monte Carlo* variance) is complicated by lack of independence in the iterates $\{X_t\}$.

The most obvious informal method for determining run length n is to run several chains in parallel, with different starting values, and compare the estimates $\overline{f}$ from (1.2). If they do not agree adequately, n must be increased. More formal methods which aim to estimate the variance of $\overline{f}$ have been proposed: see Roberts (1995) and Raftery and Lewis (1995) in this volume for further details.

1.4.8 Output analysis

In Bayesian inference, it is usual to summarize the posterior distribution $\pi(.)$ in terms of means, standard deviations, correlations, credible intervals and marginal distributions for components $X_{.i}$ of interest. Means, standard deviations and correlations can all be estimated by their sample equivalents in the Monte Carlo output $\{X_{t.i}, t = m+1, \ldots, n\}$, according to (1.2). For example, the marginal mean and variance of $X_{.i}$ are estimated by

$$\overline{X}_{.i} = \frac{1}{n-m} \sum_{t=m+1}^{n} X_{t.i}$$

and

$$S_{.i}^2 = \frac{1}{n-m-1} \sum_{t=m+1}^{n} (X_{t.i} - \overline{X}_{.i})^2.$$

Note that these estimates simply ignore other components in the Monte Carlo output.

A $100(1-2p)\%$ credible interval $[c_p, c_{1-p}]$ for a scalar component $X_{.i}$ can be estimated by setting c_p equal to the p^{th} quantile of $\{X_{t.i},\ t = m+1, \ldots, n\}$, and c_{1-p} equal to the $(1-p)^{th}$ quantile. Besag *et al.* (1995) give a procedure for calculating rectangular credible regions in two or more dimensions.

Marginal distributions can be estimated by kernel density estimation. For the marginal distribution of $X_{.i}$, this is

$$\pi(X_{.i}) \approx \frac{1}{n-m} \sum_{t=m+1}^{n} K(X_{.i}|X_t),$$

where $K(.|X_t)$ is a density concentrated around $X_{t.i}$. A natural choice for $K(X_{.i}|X_t)$ is the full conditional distribution $\pi(X_{.i}|X_{t.-i})$. Gelfand and Smith (1990) use this construction to estimate expectations under $\pi(.)$. Thus their *Rao–Blackwellized* estimator of $E[f(X_{.i})]$ is

$$\bar{f}_{RB} = \frac{1}{n-m} \sum_{t=m+1}^{n} E[f(X_{.i})|X_{t.-i}], \tag{1.12}$$

where the expectation is with respect to the full conditional $\pi(X_{.i}|X_{t.-i})$. With reasonably long runs, the improvement from using (1.12) instead of (1.2) is usually slight, and in any case (1.12) requires a closed form for the full conditional expectation.

1.5 Discussion

This chapter provides a brief introduction to MCMC. We hope we have convinced readers that MCMC is a simple idea with enormous potential. The following chapters fill out many of the ideas sketched here, and in particular give some indication of where the methods work well and where they need some tuning or further development.

MCMC methodology and Bayesian estimation go together naturally, as many of the chapters in this volume testify. However, Bayesian model validation is still a difficult area. Some techniques for Bayesian model validation using MCMC are described in Chapters 9–13.

The philosophical debate between Bayesians and non-Bayesians has continued for decades and has largely been sterile from a practical perspective. For many applied statisticians, the most persuasive argument is the availability of robust methods and software. For many years, Bayesians had difficulty solving problems which were straightforward for non-Bayesians, so it is not surprising that most applied statisticians today are non-Bayesian. With the arrival of MCMC and related software, notably the Gibbs sampling program BUGS (see Spiegelhalter *et al.*, 1995: this volume), we hope

more applied statisticians will become familiar and comfortable with Bayesian ideas, and apply them.

References

Besag, J. (1974) Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. R. Statist. Soc.* B, **36**, 192–236.

Besag, J., Green, P., Higdon, D. and Mengersen, K. (1995) Bayesian computation and stochastic systems. *Statist. Sci.* (in press).

Cowles, M. K. and Carlin, B. P. (1994) Markov chain Monte Carlo convergence diagnostics: a comparative review. *Technical Report 94-008*, Division of Biostatistics, School of Public Health, University of Minnesota.

Diebolt, J. and Ip, E. H. S. (1995) Stochastic EM: methods and application. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 259–273. London: Chapman & Hall.

Gelfand, A. E. and Smith, A. F. M. (1990) Sampling-based approaches to calculating marginal densities. *J. Am. Statist. Ass.*, **85**, 398–409.

Gelfand, A. E., Hills, S. E., Racine-Poon, A. and Smith, A. F. M. (1990) Illustration of Bayesian inference in normal data models using Gibbs sampling. *J. Am. Statist. Ass.*, **85**, 972–985.

Gelman, A. (1995) Inference and monitoring convergence. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 131–143. London: Chapman & Hall.

Gelman, A. and Rubin, D. B. (1992a) A single series from the Gibbs sampler provides a false sense of security. In *Bayesian Statistics 4* (eds J. M. Bernardo, J. Berger, A. P. Dawid and A. F. M. Smith), pp. 625–631. Oxford: Oxford University Press.

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

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

Geweke, J. (1992) Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In *Bayesian Statistics 4* (eds J. M. Bernardo, J. Berger, A. P. Dawid and A. F. M. Smith), pp. 169–193. Oxford: Oxford University Press.

Geyer, C. J. (1992) Practical Markov chain Monte Carlo. *Statist. Sci.*, **7**, 473–511.

Geyer, C. J. (1995) Estimation and optimization of functions. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 241–258. London: Chapman & Hall.

Geyer, C. J. and Møller, J. (1993) Simulation procedures and likelihood inference for spatial point processes. *Technical Report*, University of Aarhus.

Gilks, W. R. (1995) Full conditional distributions. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 75–88. London: Chapman & Hall.

Gilks, W. R. and Roberts, G. O. (1995) Strategies for improving MCMC. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 89–114. London: Chapman & Hall.

Green, P. J. (1994a) Discussion on Representations of knowledge in complex systems (by U. Grenander and M. I. Miller). *J. R. Statist. Soc.* B, **56**, 589–590.

Green, P. J. (1994b) Reversible jump MCMC computation and Bayesian model determination. *Technical Report*, Department of Mathematics, University of Bristol.

Green, P. J. (1995) MCMC in image analysis. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 381–399. London: Chapman & Hall.

Grenander, U. and Miller, M. I. (1994) Representations of knowledge in complex systems. *J. R. Statist. Soc.* B, **56**, 549–603.

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

Kass, R. E., Tierney, L. and Kadane, J. B. (1988) Asymptotics in Bayesian computation (with discussion). In *Bayesian Statistics 3* (eds J. M. Bernardo, M. H. DeGroot, D. V. Lindley and A. F. M. Smith), pp. 261–278. Oxford: Oxford University Press.

Liu, C. and Liu, J. (1993) Discussion on the meeting on the Gibbs sampler and other Markov chain Monte Carlo methods. *J. R. Statist. Soc.* B, **55**, 82–83.

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

Phillips, D. B. and Smith, A. F. M. (1995) Bayesian model comparison via jump diffusions. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 215–239. London: Chapman & Hall.

Raftery, A. E. and Lewis, S. M. (1992) How many iterations of the Gibbs sampler? In *Bayesian Statistics 4* (eds J. M. Bernardo, J. Berger, A. P. Dawid and A. F. M. Smith), pp. 641–649. Oxford: Oxford University Press.

Raftery, A. E. and Lewis, S. M. (1995) Implementing MCMC. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 115–130. London: Chapman & Hall.

Ritter, C. and Tanner, M. A. (1992) Facilitating the Gibbs sampler: the Gibbs stopper and the Griddy-Gibbs sampler. *J. Am. Statist. Ass.*, **87**, 861–868.

Roberts, G. O. (1992) Convergence diagnostics of the Gibbs sampler. In *Bayesian Statistics 4* (eds J. M. Bernardo, J. Berger, A. P. Dawid and A. F. M. Smith), pp. 775–782. Oxford: Oxford University Press.

Roberts, G. O. (1995) Markov chain concepts related to samping algorithms. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 45–57. London: Chapman & Hall.

Spiegelhalter, D. J., Best, N. G., Gilks, W. R. and Inskip, H. (1995) Hepatitis B: a case study in MCMC methods. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 21–43. London: Chapman & Hall.

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

Tierney, L. (1995) Introduction to general state-space Markov chain theory. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 59–74. London: Chapman & Hall.

Zeger, S. L. and Karim, M. R. (1991) Generalized linear models with random effects: a Gibbs sampling approach. *J. Am. Statist. Ass.*, **86**, 79–86.

Rosenthal, J. S. (1993) Minorization conditions and convergence rates for Markov chain Monte Carlo. *Technical Report*, School of Mathematics, University of Minnesota.

Smith, A. F. M. and Roberts, G. O. (1993) Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *J. R. Statist. Soc.* B, **55**, 3–24.

Taylor, H. M. and Karlin, S. (1984) *An Introduction to Stochastic Modeling.* Orlando: Academic Press.

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

5

Full conditional distributions

Walter R Gilks

5.1 Introduction

As described in Gilks *et al.* (1995b: this volume), Gibbs sampling involves little more than sampling from full conditional distributions. This chapter shows how full conditional distributions are derived, and describes methods for sampling from them.

To establish notation, vector X denotes a point in the state-space of the Gibbs sampler and $\pi(X)$ denotes its stationary distribution. The elements of X are partitioned into k components $(X_{.1}, X_{.2}, \ldots, X_{.k})$. Each of the k components of X may be scalar or vector. We define an iteration of the Gibbs sampler to be an updating of one component of X; X_t denotes the state of X at iteration t. Vector X without component s is denoted $X_{.-s} = (X_{.1}, \ldots, X_{.s-1}, X_{.s+1}, \ldots, X_{.k})$. The full conditional distribution for $X_{.s}$ at iteration t is denoted $\pi(X_{.s}|X_{t.-s})$. To avoid measure-theoretic notation, all random variables are assumed real and continuous, although much of this chapter applies also to other kinds of variable. $P(.)$ generically denotes a probability density function.

5.2 Deriving full conditional distributions

Full conditional distributions are derived from the joint distribtion of the variables:

$$\pi(X_{.s}|X_{t.-s}) = \frac{\pi(X_{.s}, X_{t.-s})}{\int \pi(X_{.s}, X_{t.-s}) dX_{.s}}. \tag{5.1}$$

5.2.1 *A simple example*

Consider the following simple two-parameter Bayesian model:

$$\begin{aligned} y_i &\sim \mathrm{N}(\mu, \tau^{-1}), \qquad i=1,\ldots,n; \\ \mu &\sim \mathrm{N}(0, 1); \\ \tau &\sim \mathrm{Ga}(2, 1), \end{aligned} \tag{5.2}$$

where $\mathrm{N}(a, b)$ generically denotes a normal distribution with mean a and variance b, and $\mathrm{Ga}(a, b)$ generically denotes a gamma distribution with mean a/b and variance a/b^2. Here we assume the $\{y_i\}$ are conditionally independent given μ and τ, and μ and τ are themselves independent. Let $y = \{y_i;\ i = 1, \ldots, n\}$.

The joint distribution of y, μ and τ is

$$\begin{aligned} P(y,\mu,\tau) &= \prod_{i=1}^{n} P(y_i|\mu,\tau)P(\mu)P(\tau) \\ &= (2\pi)^{-\frac{n+1}{2}}\tau^{\frac{n}{2}} \exp\left\{-\frac{\tau}{2}\Sigma(y_i-\mu)^2\right\}\exp\left\{-\frac{1}{2}\mu^2\right\}\tau e^{-\tau}. \end{aligned} \tag{5.3}$$

When y is observed, the joint posterior distribution of μ and τ is

$$\pi(\mu,\tau) = P(\mu,\tau|y) = \frac{P(y,\mu,\tau)}{\int P(y,\mu,\tau)d\mu d\tau}. \tag{5.4}$$

From (5.1) and (5.4), the full conditional for μ is

$$\begin{aligned} \pi(\mu|\tau) &= \frac{P(\mu,\tau|y)}{P(\tau|y)} \\ &= \frac{P(y,\mu,\tau)}{P(y,\tau)} \\ &\propto P(y,\mu,\tau). \end{aligned} \tag{5.5}$$

Here, proportionality follows because $\pi(\mu|\tau)$ is a distribution for μ, and the denominator of (5.5) does not depend on μ. Thus, to construct the full conditional for μ, we need only pick out the terms in (5.3) which involve μ, giving:

$$\begin{aligned} \pi(\mu|\tau) &\propto \exp\left\{-\frac{\tau}{2}\Sigma(y_i-\mu)^2\right\}\exp\left\{-\frac{1}{2}\mu^2\right\} \\ &\propto \exp\left\{-\frac{1}{2}(1+n\tau)\left(\mu-\frac{\tau\Sigma y_i}{1+n\tau}\right)^2\right\}. \end{aligned}$$

Thus, the full conditional for μ is a normal distribution with mean $\frac{\tau\Sigma y_i}{1+n\tau}$ and variance $(1+n\tau)^{-1}$. Similarly, the full conditional for τ depends only on the terms in (5.3) involving τ, giving:

$$\begin{aligned} \pi(\tau|\mu) &\propto \tau^{\frac{n}{2}}\exp\left\{-\frac{\tau}{2}\Sigma(y_i-\mu)^2\right\}\tau e^{-\tau} \\ &= \tau^{1+\frac{n}{2}}\exp\left\{-\tau\left[1+\frac{1}{2}\Sigma(y_i-\mu)^2\right]\right\}, \end{aligned}$$

which is the kernel of a gamma distribution with index $2+\frac{n}{2}$ and scale $1+\frac{1}{2}\Sigma(y_i-\mu)^2$.

In this simple example, prior distributions are conjugate to the likelihood (5.2), so full conditionals reduce analytically to closed-form distributions. Highly efficient sampling routines are available for these distributions; see for example Ripley (1987).

5.2.2 *Graphical models*

Full conditional distributions for complex models can also be constructed easily. In particular, for Bayesian directed acyclic graphical (DAG) models, the joint distribution of the data and parameters is a product of many terms, each involving only a subset of the parameters. For such models, the full conditional distribution for any given parameter can be constructed from those few terms of the joint distribution which depend on it; see Spiegelhalter *et al.* (1995b: this volume).

Normal random-effects model

For example, consider the random-effects model:

$$\begin{aligned} y_{ij} &\sim N(\alpha_i, \tau^{-1}), \qquad j=1,\ldots,m_i, \qquad i=1,\ldots,n; \\ \alpha_i &\sim N(\mu, \omega^{-1}), \qquad i=1,\ldots,n; \\ \mu &\sim N(0, 1); \\ \tau &\sim Ga(2, 1); \\ \omega &\sim Ga(1, 1), \end{aligned}$$

where we assume independence between the $\{y_{ij}\}$ given all model parameters; between the $\{\alpha_i\}$ given the hyperparameters μ, τ and ω; and between the hyperparameters themselves. The joint distribution of the data and parameters for this model is:

$$P(y,\alpha,\mu,\tau,\omega) = \prod_{i=1}^{n}\left\{\prod_{j=1}^{m_i} P(y_{ij}|\alpha_i,\tau)P(\alpha_i|\mu,\omega)\right\}P(\mu)P(\tau)P(\omega).$$

Then the full conditional for α_i is

$$\pi(\alpha_i|y,\alpha_{-i},\mu,\tau,\omega) \propto \prod_{j=1}^{m_i} P(y_{ij}|\alpha_i,\tau)P(\alpha_i|\mu,\omega) \tag{5.6}$$

$$\propto \exp\left\{-\frac{1}{2}(\omega + m_i\tau)\left(\alpha_i - \frac{\omega\mu + \tau\sum_{j=1}^{m_i} y_{ij}}{\omega + m_i\tau}\right)^2\right\},$$

which is a normal distribution with mean

$$\frac{\omega\mu + \tau\Sigma_{j=1}^{m_i} y_{ij}}{\omega + m_i\tau}$$

and variance $(\omega + m_i\tau)^{-1}$.

Logistic regression model

Although for DAG models it is trivial to write down expressions for full conditionals, as in (5.6), it is often not possible to make further progress analytically. For example, consider the following Bayesian logistic regression model of y on covariate z:

$$\begin{aligned} y_i &\sim \text{Bernoulli}\left(\frac{1}{1+e^{-(\mu+\alpha z_i)}}\right), \qquad i = 1, \ldots, n; \qquad (5.7)\\ \alpha &\sim \text{N}(0, 1);\\ \mu &\sim \text{N}(0, 1), \end{aligned}$$

where we assume conditional independence between the $\{y_i\}$ given the model parameters and covariates, and independence between the parameters themselves. Here, the full conditional for α is

$$\pi(\alpha|\mu) \quad\propto\quad e^{-\frac{1}{2}\alpha^2}\prod_{i=1}^{n}\{1+e^{-(\mu+\alpha z_i)}\}^{-y_i}\{1+e^{\mu+\alpha z_i}\}^{y_i - 1}, \qquad (5.8)$$

which unfortunately does not simplify. Thus methods are required for sampling from arbitrarily complex full conditional distributions. This is the subject of the remainder of this chapter.

Undirected graphical models

For non-DAG models, full conditionals may be difficult to derive, although for some partially-DAG models the derivation is straightforward; see for example Mollié (1995: this volume).

5.3 Sampling from full conditional distributions

Full conditionals change from iteration to iteration as the conditioning $X_{t.-s}$ changes, so each full conditional is used only once and then disposed of. Thus it is essential that sampling from full conditional distributions is highly efficient computationally. When analytical reduction of a full conditional is not possible, it will be necessary to evaluate the full conditional function at a number of points, and in typical applications each function evaluation will be computationally expensive. Thus any method for sampling from full conditional distributions should aim to minimize the number of function evaluations. Sampling methods such as inversion (see Ripley, 1987), which require a large number of function evaluations, should be avoided if possible.

Two techniques for sampling from a general density $g(y)$ are rejection sampling and the ratio-of-uniforms method. A third method, which does not produce independent samples, is the Metropolis–Hastings algorithm. All three methods can be used for sampling multivariate distributions, and none require evaluation of the normalizing constant for g. This is an important practical point, since the normalizing constant for full conditional distributions is typically unavailable in closed form (as in (5.8), for example). We now describe these methods, and hybrids of them, for sampling from full conditional distributions. Below, Y represents $X_{t+1.s}$ and $g(Y)$ is proportional to the density of interest $\pi(X_{t+1.s}|X_{t.-s})$.

5.3.1 Rejection sampling

Rejection sampling requires an envelope function G of g (so $G(Y) \geq g(Y)$ for all Y: see Figure 5.1). Samples are drawn from the density proportional to G, and each sampled point Y is subjected to an accept/reject test. This test takes the form: accept point Y with probability $g(Y)/G(Y)$. If the point is not accepted, it is discarded. Sampling continues until the required number of points have been accepted: for Gibbs sampling just one point is required from each full conditional g. Accepted points are then exactly independent samples from the density proportional to g (see for example Ripley, 1987).

The algorithm then is:

```
Repeat {
  Sample a point Y from G(.);
  Sample a Uniform(0,1) random variable U;
  If U ≤ g(Y)/G(Y) accept Y; }
until one Y is accepted.
```

Several rejections may occur before an acceptance. Each accept/reject test involves evaluating $g(Y)$ and $G(Y)$, and typically the former will be computationally expensive. Marginally, the probability of accepting a point is $\int g(Y)dY / \int G(Y)dY$, so to reduce the number of rejections, it is essential that the envelope G be close to g. For computational efficiency, it is also essential that G be cheap to evaluate and sample from.

Some computational savings may result from using *squeezing functions* $a(Y)$ and $b(Y)$, where $a(Y) \geq g(Y) \geq b(Y)$ for all Y, and a and b are cheaper to evaluate than g (see Figure 5.1). The accept/reject test on line 4 of the above algorithm can then be replaced by

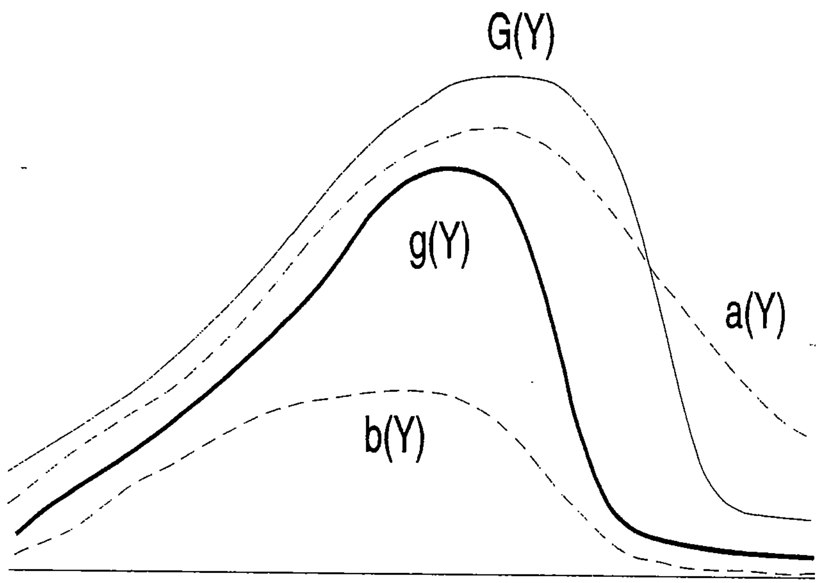

Figure 5.1 *Functions for rejection sampling. Thin line: envelope $G(Y)$; heavy line: density $g(Y)$; broken lines: squeezing functions $a(Y)$ and $b(Y)$.*

```
If U > a(Y)/G(Y) reject Y;
  else if U ≤ b(Y)/G(Y) accept Y;
  else if U ≤ g(Y)/G(Y) accept Y.
```

The first two tests enable a decision to be made about Y without calculating $g(Y)$.

Zeger and Karim (1991) and Carlin and Gelfand (1991) propose rejection sampling for multivariate full conditional distributions, using multivariate normal and multivariate split-t distributions as envelopes. A difficulty with these methods is in establishing that the proposed envelopes are true envelopes. Bennett *et al.* (1995: this volume) use rejection sampling for multivariate full conditional distributions in nonlinear models. For the envelope function G, they use the prior distribution multiplied by the likelihood at the maximum likelihood estimate.

5.3.2 Ratio-of-uniforms method

Suppose Y is univariate. Let U and V be two real variables, and let $\mathcal{D}$ denote a region in U, V space defined by $0 \le U \le \sqrt{g(V/U)}$ (see Figure 5.2). Sample a point U, V uniformly from $\mathcal{D}$. This can be done by first determining an envelope region $\mathcal{E}$ which contains $\mathcal{D}$ and from which it is easy to sample uniformly. U and V can then be generated by rejection sampling

from $\mathcal{E}$. Rather surprisingly, $Y = V/U$ is a sample from the density proportional to g (see for example Ripley, 1987).

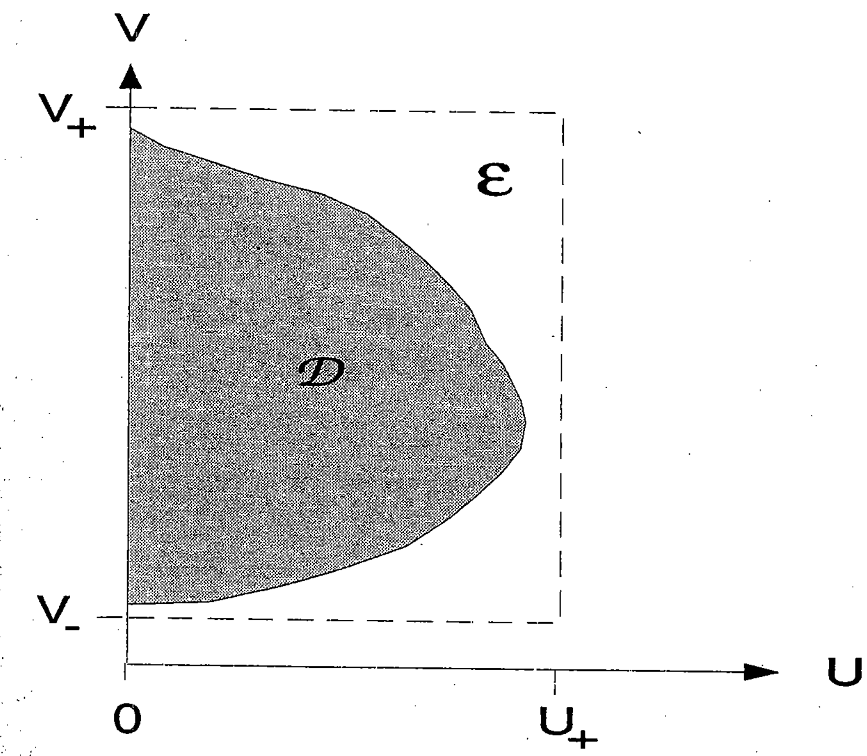

Figure 5.2 *An envelope $\mathcal{E}$ (broken line) for a region $\mathcal{D}$ defined by $0 \le U \le \sqrt{g(V/U)}$, for the ratio-of-uniforms method.*

Typically $\mathcal{E}$ is chosen to be a rectangle with vertices at $(0, v_-)$; (u_+, v_-); $(0, v_+)$; and (u_+, v_+), where constants u_+, v_- and v_+ are such that $\mathcal{E}$ contains $\mathcal{D}$. This leads to the following algorithm.

```
Determine constants u+, v-, v+;
Repeat {
  Sample a Uniform(0, u+) random variable U;
  Sample a Uniform(v-, v+) random variable V;
  If (U,V) is in D, accept Y = V/U; }
until one Y is accepted.
```

As in pure rejection sampling, it is important for computational efficiency to keep the number of rejections low. If squeezing regions can be found, efficiency may be improved. Wakefield *et al.* (1991) give a multivariate generalization of the ratio-of-uniforms method, and suggest variable transformations to improve its efficiency. Bennett *et al.* (1995: this volume) compare the ratio-of-uniforms method with other methods for sampling from full conditional distributions in nonlinear models.

5.3.3 *Adaptive rejection sampling*

The practical problem with both rejection sampling and the ratio-of-uniforms method is in finding a tight envelope function G or region $\mathcal{E}$. Often this will involve time-consuming maximizations, exploiting features peculiar to g. However, for the important class of log-concave univariate densities, efficient methods of envelope construction have been developed. A function $g(Y)$ is log-concave if the determinant of $\frac{d^2 \log g}{dY\, dY^T}$ is non-positive.

In many applications of Gibbs sampling, all full conditional densities $g(Y)$ are log-concave (Gilks and Wild, 1992). In particular, this is true for all generalized linear models with canonical link function (Dellaportas and Smith, 1993). For example, full conditional distributions in the logistic regression model (5.7) are log-concave. Gilks and Wild (1992) show that, for univariate Y, an envelope function $\log G_S(Y)$ for $\log g(Y)$ can be constructed by drawing tangents to $\log g$ at each abscissa in a given set of abscissae $\mathcal{S}$. An envelope between any two adjacent abscissae is then constructed from the tangents at either end of that interval (Figure 5.3(a)). An alternative envelope construction which does not require evaluation of derivatives of $\log g$ is given by Gilks (1992). For this, secants are drawn through $\log g$ at adjacent abscissae, and the envelope between any two adjacent abscissae is constructed from the secants immediately to the left and right of that interval (Figure 5.3(b)). For both constructions, the envelope is piece-wise exponential, from which sampling is straightforward. Also, both constructions automatically provide a lower squeezing function $\log b_S(Y)$.

Three or four starting abscissae usually suffice, unless the density is exceptionally concentrated. Both methods require starting abscissae to be placed on both sides of the mode if the support of g is unbounded. This does not involve locating the mode, since gradients of tangents or secants determine whether the starting abscissae are acceptable. If desired, starting abscissae can be set with reference to the envelope constructed at the previous Gibbs iteration.

The important feature of both of these envelope constructions is that they can be used *adaptively*. When a Y is sampled, $g(Y)$ must be evaluated to perform the rejection step. Then, with negligible computational cost, the point $(Y, g(Y))$ can be incorporated in the envelope, just as if Y had been among the initial abscissae. This is called *adaptive rejection sampling* (ARS):

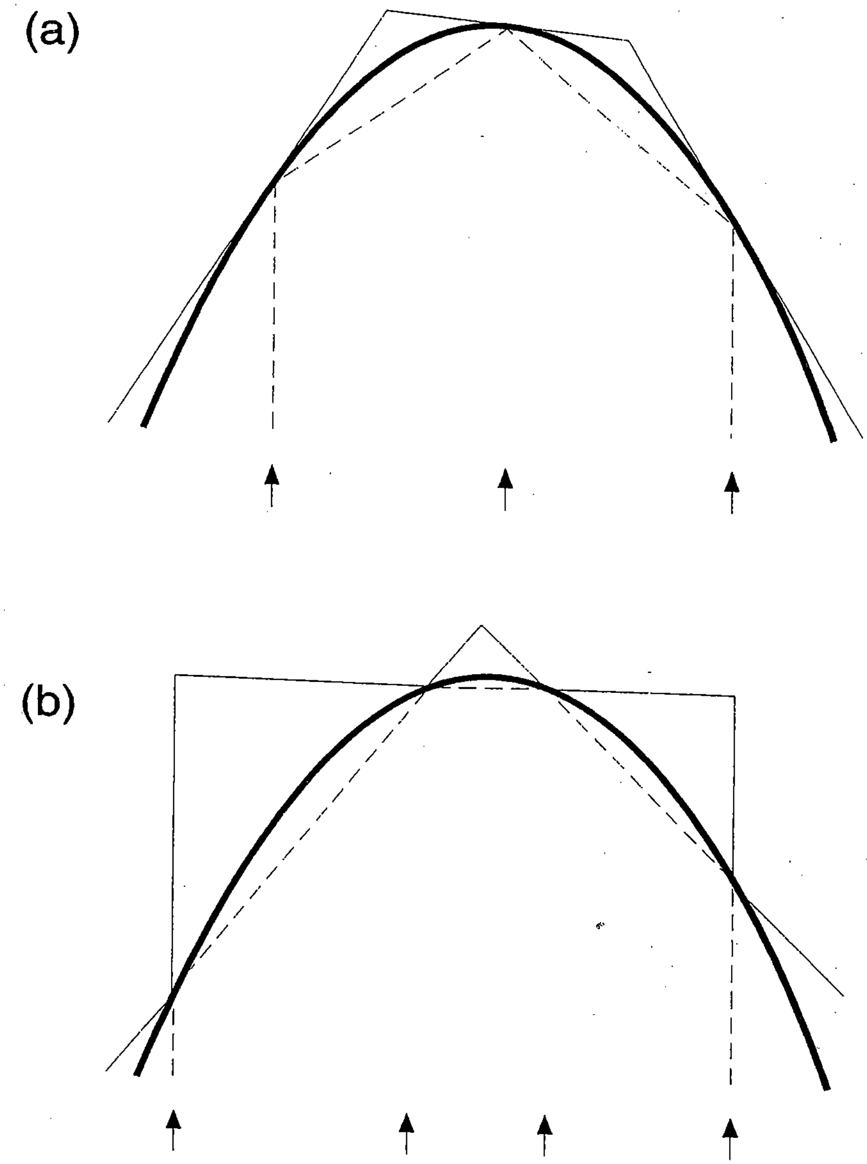

Figure 5.3 *Adaptive rejection sampling: (a) tangent method; (b) secant method. Heavy line:* $\log g(Y)$*; thin line: envelope* $\log G_S(Y)$*; broken line: squeezing function* $\log b_S(Y)$*; arrows: abscissae used in the construction.*

```
Initialize S
Repeat {
  Sample Y from G_S(.);
  Sample U from Uniform(0,1);
  If U ≤ g(Y)/G_S(Y) accept Y;
  Include Y in S; }
until one Y is accepted.
```

At each iteration of ARS, the envelope $G_S(Y)$ is brought closer to g and the risk of further rejections and function evaluations is reduced. To accept one Y, the tangent version of adaptive rejection sampling typically involves about four function evaluations including those at the initial abscissae; for the secant version, five or six function evaluations are usually required. These performance figures are surprisingly robust to location of starting abscissae and to the form of g.

Multivariate generalizations of adaptive rejection sampling are possible, but have not yet been implemented. The amount of computation for such methods could be of order m^5, where m is the number of dimensions. Thus multivariate adaptive rejection sampling would probably be useful only in low dimensions.

5.3.4 Metropolis–Hastings algorithm

When an approximation $h(Y)$ to full conditional $g(Y)$ is available, from which sampling is easy, it is tempting to sample from h instead of from g. Then ergodic averages calculated from the output of the Gibbs sampler will not correspond exactly to π, no matter how long the chain is run. Ritter and Tanner (1992) propose grid-based methods for approximate sampling from full conditional distributions, successively refining the grid as the iterations proceed to reduce the element of approximation. Thus, approximation is improved at the cost of increasing computational burden.

Tierney (1991) and Gelman (1992) suggest a way to sample from approximate full conditional distributions whilst maintaining exactly the required stationary distribution of the Markov chain. This involves using the approximate full conditional h as a proposal distribution in an independence-type Metropolis–Hastings algorithm (see Gilks *et al.*, 1995b: this volume):

Sample a point Y **from** $h(.)$;
Sample a Uniform(0,1) random variable U;
If $U \leq \min[1, \frac{g(Y)h(Y')}{g(Y')h(Y)}]$ **accept** Y;
  else set Y **equal to** Y';

where $Y' = X_{t.s}$ is the 'old' value of $X_{.s}$. Note that only one iteration of Metropolis–Hastings is required, because if X_t is from π, then so is $X_{t+1} = (X_{t+1.s}, X_{t.-s})$. Note also that multivariate full conditionals can be handled using this technique.

If $g(Y)$ is unimodal and not heavy-tailed, a convenient independence-type proposal $h(Y)$ might be a normal distribution whose scale and location are chosen to match g, perhaps via a least-squares fit of $\log h$ to $\log g$ at several well-spaced points. For more complex g, proposals could be mixtures of normals or scale- and location-shifted t-distributions. In general, if h approximates g well, there will be few Metropolis–Hastings rejections, and this will generally assist mixing in the Markov chain. However, there is clearly a trade-off between reducing the rejection rate and the computational burden of calculating good approximations to g.

The above algorithm is no longer purely Gibbs sampling: it produces a different Markov chain but with the same stationary distribution π. The proposal density h need not be an approximation to g, nor need it be of the independence type. Tierney (1991) and Besag and Green (1993) suggest that it can be advantageous to use an h which is distinctly different from g, to produce an antithetic variables effect in the output which will reduce Monte-Carlo standard errors in ergodic averages. Such chains have been called 'Metropolis–Hastings-within-Gibbs', but as the original algorithm described by Metropolis *et al.* (1953) uses single-component updating, the term 'single-component Metropolis–Hastings' is more appropriate (Besag and Green, 1993).

5.3.5 Hybrid adaptive rejection and Metropolis–Hastings

Tierney (1991) discusses the use of Metropolis–Hastings in conjunction with rejection sampling. Extending this idea, ARS can be used to sample adaptively from non-log-concave univariate full conditional distributions. For non-log-concave densities, the 'envelope' functions $G_S(Y)$ calculated as described in Section 5.3.3 may not be true envelopes; in places the full conditional $g(Y)$ may protrude above $G_S(Y)$. Then the sample delivered by ARS will be from the density proportional to $h(Y) = \min[g(Y), G_S(Y)]$, where S is the set of abscissae used in the final accept/reject step of ARS. A sample Y from g can then be obtained by appending the following Metropolis–Hastings step to ARS:

Sample U **from Uniform(0,1)**;
If $U \leq \min\{1, \frac{g(Y)h(Y')}{g(Y')h(Y)}\}$ **accept** Y;
  else set Y **equal to** Y'.

Here, as before, $Y' = X_{t.s}$. This is *adaptive rejection Metropolis sampling* (ARMS) (Gilks *et al.*, 1995a).

ARMS works well when g is nearly log-concave, and reduces to ARS when g is exactly log-concave. When g is grossly non-log-concave, ARMS still delivers samples from g, but rejections at the Metropolis–Hastings step will be more frequent. As for ARS, the initial set of abscissae in S may be chosen to depend on G_S constructed at the previous Gibbs iteration.

Besag *et al.* (1995) note that the number of iterations in the repeat loop of ARS (or ARMS) is unbounded, and suggest using Metropolis–Hastings to curtail the number of these iterations. Roberts *et al.* (1995) suggest the following implementation of that idea. Let c denote the maximum permitted number of iterations of the repeat loop of ARS. If each of the c iterations result in rejection, perform a Metropolis–Hastings step as for ARMS above, but with $h(Y) = G_S(Y) - \min[g(Y), G_S(Y)]$, and using the value of Y generated at the c^{th} step of ARS. It is unlikely that curtailment for log-concave g would offer computational advantages, since log-concavity ensures acceptance of a Y within very few iterations. However, curtailment for very non-log-concave g may sometimes be worthwhile.

5.4 Discussion

In general, the Gibbs sampler will be more efficient (better mixing) if the number of components k of X is small (and the dimensionality of the individual components is correspondingly large). However, sampling from complex multivariate distributions is generally not possible unless MCMC itself is used, as in Section 5.3.4. Why not therefore abandon Gibbs sampling in favour of Metropolis–Hastings applied to the whole of X simultaneously? Often this would be a sensible strategy, but Metropolis–Hastings requires finding a reasonably efficient proposal distribution, which can be difficult in problems where dimensions are scaled very differently to each other. In many problems, Gibbs sampling applied to univariate full conditional distributions works well, as demonstrated by the wealth of problems efficiently handled by the BUGS software (Spiegelhalter *et al.*, 1994, 1995a), but for difficult problems and for robust general-purpose software, hybrid methods are likely to be most powerful. See Gilks and Roberts (1995: this volume) for a discussion of techniques for improving the efficiency of MCMC, and Bennett *et al.* (1995: this volume) for a comparison of various methods for sampling from full conditional distributions in the context of nonlinear models.

FORTRAN code for ARS and C code for ARMS are available from the author (e-mail wally.gilks@mrc-bsu.cam.ac.uk).

References

Bennett, J. E., Racine-Poon, A. and Wakefield, J. C. (1995) MCMC for nonlinear hierarchical models. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 339–357. London: Chapman & Hall.

Besag, J. and Green, P. J. (1993) Spatial statistics and Bayesian computation. *J. R. Statist. Soc.* B, **55**, 25–37.

Besag, J., Green, P. J., Higdon, D. and Mengerson, K. (1995) Bayesian computation and stochastic systems. *Statist. Sci.*, **10**, 3–41.

Carlin, B. P. and Gelfand, A. E. (1991) An iterative Monte Carlo method for nonconjugate Bayesian analysis. *Statist. Comput.*, **1**, 119–128.

Dellaportas, P. and Smith, A. F. M. (1993) Bayesian inference for generalised linear and proportional hazards models via Gibbs sampling. *Appl. Statist.*, **42**, 443–460.

Gelman, A. (1992) Iterative and non-iterative simulation algorithms. In *Computing Science and Statistics* (ed. H. J. Newton), pp. 433–438. Fairfax Station: Interface Foundation of North America.

Gilks, W. R. (1992) Derivative-free adaptive rejection sampling for Gibbs sampling. In *Bayesian Statistics 4* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith), pp. 641–649. Oxford: Oxford University Press.

Gilks, W. R. and Roberts, G. O. (1995) Strategies for improving MCMC. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 89–114. London: Chapman & Hall.

Gilks, W. R. and Wild, P. (1992) Adaptive rejection sampling for Gibbs sampling. *Appl. Statist.*, **41**, 337–348.

Gilks, W. R., Best, N. G. and Tan, K. K. C. (1995a) Adaptive rejection Metropolis sampling within Gibbs sampling. *Appl. Statist.*, (in press).

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (1995b) Introducing Markov chain Monte Carlo. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 1–19. London: Chapman & Hall.

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

Mollié, A. (1995) Bayesian mapping of disease. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 359–379. London: Chapman & Hall.

Ripley, B. D. (1987) *Stochastic Simulation*. New York: Wiley.

Ritter, C. and Tanner, M. A. (1992) Facilitating the Gibbs sampler: the Gibbs stopper and the griddy–Gibbs sampler. *J. Am. Statist. Ass.*, **87**, 861–868.

Roberts, G. O., Sahu, S. K. and Gilks, W. R. (1995) Discussion on Bayesian computation and stochastic systems (by J. Besag, P. J. Green, D. Higdon and K. Mengerson). *Statist. Sci.*, **10**, 49–51.

Spiegelhalter, D. J., Thomas, A. and Best, N. G. (1995a). Computation on Bayesian graphical models. In *Bayesian Statistics 5* (eds J. M. Bernardo, J. Berger, A. P. Dawid and A. F. M. Smith). Oxford: Oxford University Press (in press).

Spiegelhalter, D. J., Best, N. G., Gilks, W. R. and Inskip, H. (1995b) Hepatitis B: a case study in MCMC methods. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 21–43. London: Chapman & Hall.

Spiegelhalter, D. J., Thomas, A., Best, N. G. and Gilks, W. R. (1994) *BUGS: Bayesian inference Using Gibbs Sampling.* Cambridge: MRC Biostatistics Unit.

Tierney, L. (1991) Exploring posterior distributions using Markov chains. In *Computer Science and Statistics: Proc. 23rd Symp. Interface* (ed. E. Keramidas), pp. 563–570. Fairfax Station: Interface Foundation.

Wakefield, J. C., Gelfand, A. E. and Smith, A. F. M. (1991) Efficient generation of random variates via the ratio-of-uniforms method. *Statist. Comput.*, **1**, 129–133.

Zeger, S. and Karim, M. R. (1991) Generalised linear models with random effects: a Gibbs sampling approach. *J. Am. Statist. Ass.*, **86**, 79–86.

6

Strategies for improving MCMC

Walter R Gilks
Gareth O Roberts

6.1 Introduction

In many applications raw MCMC methods, in particular the Gibbs sampler, work surprisingly well. However, as models become more complex, it becomes increasingly likely that untuned methods will not *mix* rapidly. That is, the Markov chain will not move rapidly throughout the support of the target distribution. Consequently, unless the chain is run for very many iterations, Monte-Carlo standard errors in output sample averages will be large. See Roberts (1995) and Tierney (1995) in this volume for further discussion of Monte-Carlo standard errors and Markov chain mixing.

In almost any application of MCMC, many models must be explored and refined. Thus poor mixing can be severely inhibiting. Run times of the order of seconds or minutes are desirable, runs taking hours are tolerable, but longer run times are practically impossible to work with. As models become more ambitious, the practitioner must be prepared to experiment with strategies for improving mixing. Techniques for reducing the amount of computation per iteration are also important in reducing run times.

In this chapter, we review strategies for improving run times of MCMC. Our aim is to give sufficient detail for these strategies to be implemented: further information can be found in the original references. For readers who are new to MCMC methodology, we emphasize that familiarity with the material in this chapter is not a prerequisite for successful application of MCMC; Gilks *et al.* (1995b: this volume) provide enough information to permit application of MCMC in straightforward situations.

For simplicity, we will mostly assume that the Markov chain takes values in k-dimensional Euclidean space $\mathbb{R}^k$, although most of the techniques we discuss apply more generally. The target density (for example a posterior