Markov Chain Monte Carlo Method and Its Application
Author(s): Stephen P. Brooks
Source: *Journal of the Royal Statistical Society. Series D (The Statistician)*, Vol. 47, No. 1 (1998), pp. 69-100
Published by: Blackwell Publishing for the Royal Statistical Society
Stable URL: http://www.jstor.org/stable/2988428
Accessed: 18/08/2010 18:26

The Statistician (1998)
47, *Part* 1, *pp.* 69–100

Markov chain Monte Carlo method and its application

Stephen P. Brooks†

University of Bristol, UK

[Received April 1997. Revised October 1997]

Summary. The Markov chain Monte Carlo (MCMC) method, as a computer-intensive statistical tool, has enjoyed an enormous upsurge in interest over the last few years. This paper provides a simple, comprehensive and tutorial review of some of the most common areas of research in this field. We begin by discussing how MCMC algorithms can be constructed from standard building-blocks to produce Markov chains with the desired stationary distribution. We also motivate and discuss more complex ideas that have been proposed in the literature, such as continuous time and dimension jumping methods. We discuss some implementational issues associated with MCMC methods. We take a look at the arguments for and against multiple replications, consider how long chains should be run for and how to determine suitable starting points. We also take a look at graphical models and how graphical approaches can be used to simplify MCMC implementation. Finally, we present a couple of examples, which we use as case-studies to highlight some of the points made earlier in the text. In particular, we use a simple changepoint model to illustrate how to tackle a typical Bayesian modelling problem via the MCMC method, before using mixture model problems to provide illustrations of good sampler output and of the implementation of a reversible jump MCMC algorithm.

Keywords: Bayesian statistics; Gibbs sampler; Metropolis–Hastings updating; Simulation; Software

1. Introduction

The integration operation plays a fundamental role in Bayesian statistics. For example, given a sample $\mathbf{y}$ from a distribution with likelihood $L(\mathbf{y}|x)$ and a prior density for $\mathbf{x} \in \mathbb{R}^p$ given by $p(\mathbf{x})$, Bayes's theorem relates the posterior $\pi(\mathbf{x}|\mathbf{y})$ to the prior via the formula

$$\pi(\mathbf{x}|\mathbf{y}) \propto L(\mathbf{y}|\mathbf{x})\, p(\mathbf{x}),$$

where the constant of proportionality is given by

$$\int L(\mathbf{y}|x) p(x)\, \mathrm{d}x. \tag{1}$$

Given the posterior, and in the case where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ is multivariate, for example, we may be interested in the marginal posterior distributions, such as

$$\pi(\mathbf{x}_1|\mathbf{y}) = \int \pi(\mathbf{x}_1, \mathbf{x}_2|\mathbf{y})\, \mathrm{d}\mathbf{x}_2. \tag{2}$$

Alternatively, we might be interested in summary inferences in the form of posterior expectations, e.g.

†*Address for correspondence*: Department of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TW, UK.

0039–0526/98/47069

$$\mathbb{E}\{\theta(x)|\mathbf{y}\} = \int \theta(x)\,\pi(x|\mathbf{y})\,\mathrm{d}x. \tag{3}$$

Thus, the ability to integrate often complex and high dimensional functions is extremely important in Bayesian statistics, whether it is for calculating the normalizing constant in expression (1), the marginal distribution in equation (2) or the expectation in equation (3).

Often, an explicit evaluation of these integrals is not possible and, traditionally, we would be forced to use numerical integration or analytic approximation techniques; see Smith (1991). However, the Markov chain Monte Carlo (MCMC) method provides an alternative whereby we sample from the posterior directly, and obtain sample estimates of the quantities of interest, thereby performing the integration implicitly.

The idea of MCMC sampling was first introduced by Metropolis *et al.* (1953) as a method for the efficient simulation of the energy levels of atoms in a crystalline structure and was subsequently adapted and generalized by Hastings (1970) to focus on statistical problems, such as those described above. The idea is extremely simple. Suppose that we have some distribution $\pi(x)$, $x \in E \subseteq \mathbb{R}^p$, which is known only up to some multiplicative constant. We commonly refer to this as the *target* distribution. If π is sufficiently complex that we cannot sample from it directly, an indirect method for obtaining samples from π is to construct an aperiodic and irreducible Markov chain with state space E, and whose stationary (or invariant) distribution is $\pi(x)$, as discussed in Smith and Roberts (1993), for example. Then, if we run the chain for sufficiently long, simulated values from the chain can be treated as a dependent sample from the target distribution and used as a basis for summarizing important features of π.

Under certain regularity conditions, given in Roberts and Smith (1994) for example, the Markov chain sample path mimics a random sample from π. Given realizations $\{X^t\colon t = 0, 1, \ldots\}$ from such a chain, typical asymptotic results include the distributional convergence of the realizations, i.e. the distribution of the state of the chain at time t converges to π as $t \to \infty$, and the consistency of the ergodic average, i.e., for any scalar functional θ,

$$\frac{1}{n}\sum_{t=1}^{n} \theta(X^t) \overset{n\to\infty}{\longrightarrow} \mathbb{E}_\pi\{\theta(X)\} \qquad \text{almost surely.}$$

Many important implementational issues are associated with MCMC methods. These include (among others) the choice of transition mechanism for the chain, the number of chains to be run and their length, the choice of starting values and both estimation and efficiency problems. These will be discussed further in Section 3.

In the next section we discuss some of the basic MCMC methods proposed in the literature, before moving on to discuss implementational issues associated with these methods. In Section 4 we discuss some more complex forms of the MCMC method, for use in cases where the standard algorithms perform poorly. Finally, in Section 5 we present some applications of these methods, which illustrate many of the points raised in the earlier sections of the paper.

2. The standard Markov chain Monte Carlo updating schemes

The common undergraduate approach to Markov chain theory, that is familiar to us all, is to start with some transition distribution (a transition matrix in the discrete case) modelling some process of interest, to determine conditions under which there is an invariant or stationary distribution and then to identify the form of that limiting distribution. MCMC methods involve the solution of the inverse of this problem whereby the stationary distribution is known, and it is the transition distribution that needs to be identified, though in practice there may be infinitely many distributions to choose from. For the purposes of this paper, we shall denote the Markov chain transition distribution (or transition kernel) by $\mathcal{K}$, so that, if the chain is at present in state $\mathbf{x}$, then the conditional distribution of the next state of the chain $\mathbf{y}$, given the present state, is denoted by $\mathcal{K}(\mathbf{x}, \mathbf{y})$.

The main theorem underpinning the MCMC method is that any chain which is *irreducible* and *aperiodic* will have a unique stationary distribution, and that the t-step transition kernel will 'converge' to that stationary distribution as $t \to \infty$. (See Meyn and Tweedie (1993) for discussion, proof and explanation of the unfamiliar terms.) Thus, to generate a chain with stationary distribution π, we need only to find transition kernels $\mathcal{K}$ that satisfy these conditions and for which $\pi\mathcal{K} = \pi$, i.e. $\mathcal{K}$ is such that, given an observation $\mathbf{x} \sim \pi(\mathbf{x})$, if $\mathbf{y} \sim \mathcal{K}(\mathbf{x}, \mathbf{y})$, then $\mathbf{y} \sim \pi(\mathbf{y})$, also.

A Markov chain with stationary distribution π is called *time reversible* (or simply reversible) if its transition kernel $\mathcal{K}$ is such that it exhibits detailed balance, i.e.

$$\pi(\mathbf{x})\,\mathcal{K}(\mathbf{x}, \mathbf{y}) = \pi(\mathbf{y})\,\mathcal{K}(\mathbf{y}, \mathbf{x}). \tag{4}$$

This essentially means that the chain would look the same whether you ran it forwards in time or backwards. The behaviour of reversible chains is well understood, and it is a desirable property for any MCMC transition kernel to have, since any transition kernel for which equation (4) holds will have stationary distribution π.

Traditionally, the MCMC literature has talked only in terms of MCMC *samplers* and *algorithms*. However, this unnecessarily restricts attention to Markov chains based on only a single form of transition kernel. In practice, it is often most sensible to combine a number of different transition kernels to construct a Markov chain which performs well. Thus, it is perhaps more appropriate to discuss not MCMC algorithms but MCMC *updates* or *transitions*, and this is the approach that we adopt here. In practice, the manner in which different transitions are combined is by splitting the state vector of the Markov chain into a number of distinct components and by using different transition kernels to update each component. We shall discuss how these updating schemes can be combined in greater detail in Section 2.3, but first we introduce what might be considered the most standard transition types for updating components of the state vector. We begin with the Gibbs transition kernel.

2.1. The Gibbs transition kernel

The Gibbs sampler was introduced to the image analysis literature by Geman and Geman (1984) and subsequently to the statistics literature by Besag and York (1989), followed by Gelfand and Smith (1990). The Gibbs sampler proceeds by splitting the state vector into a number of components and updating each in turn by a series of Gibbs transitions. We shall discuss the Gibbs sampler in Section 2.3 and concentrate here on the Gibbs transition kernel on which it is based.

Suppose that we split the state vector into $k \leqslant p$ components, so that $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \mathbb{R}^p$. Having selected component $\mathbf{x}_i$ to be updated, the Gibbs transition kernel involves sampling a new state $\mathbf{x}' = (\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{y}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_k)$, sampling $\mathbf{y}$ from the conditional distribution of $\mathbf{x}_i$ given the other variables.

More formally, if we let $\pi(\mathbf{x}_i|\mathbf{x}_{(i)})$ denote the conditional distribution of $\mathbf{x}_i$, given the values of the other components $\mathbf{x}_{(i)} = (\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_k)$, $i = 1, \ldots, k$, $1 < k \leqslant p$, then a single Gibbs transition updates $\mathbf{x}'$ by sampling a new value for $\mathbf{x}_i$ from $\pi(\mathbf{x}_i|\mathbf{x}'_{(i)})$.

Conceptually, the Gibbs transition is fairly straightforward. Ideally, the conditional distribution $\pi(\mathbf{x}_i|\mathbf{x}_{(i)})$ will be of the form of a standard distribution and a suitable prior specification often ensures that this is the case. However, in the cases where it is non-standard, there are many ways to sample from the appropriate conditionals. Examples include the ratio method (Wakefield *et al.*, 1991) or, if the conditional is both univariate and log-concave, adaptive rejection sampling (Gilks and Wild, 1992) may be used. The crucial point to note here is that, contrary to ordinary simulation, we only require a single observation from these distributions; thus simulation methods with low start-up costs are essential. Alternatively, the introduction of auxiliary variables, although increasing the dimensionality, may be used to turn the relevant conditional into a standard distribution; see Section 3.6.

2.2. Metropolis–Hastings updates

An alternative, and more general, updating scheme is as a form of generalized rejection sampling, where values are drawn from arbitrary (yet sensibly chosen) distributions and 'corrected' so that, asymptotically, they behave as random observations from the target distribution. This is the motivation for methods such as the Metropolis–Hastings updating scheme.

The Metropolis–Hastings updating scheme was first described by Hastings (1970) as a generalization of the Metropolis algorithm of Metropolis *et al.* (1953). Given a partition of the state vector into components, i.e. $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$, and that we wish to update the ith component, the Metropolis–Hastings update proceeds as follows. We begin with a density for generating candidate observations $\mathbf{y}$, such that $\mathbf{y}_{(i)} = \mathbf{x}_{(i)}$, and which we denote by $q(\mathbf{x}, \mathbf{y})$. The definition of q is essentially arbitrary, subject to the condition that the resulting chain is aperiodic and irreducible and has stationary distribution π and, in practice, it is generally selected so that observations may be generated from it with reasonable ease. Having generated a new state $\mathbf{y} = (\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{y}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_k)$, from density $q(\mathbf{x}, \mathbf{y})$, we then accept this point as the new state of the chain with probability $\alpha(\mathbf{x}, \mathbf{y})$, given by

$$\alpha(\mathbf{x}, \mathbf{y}) = \min\left\{1, \frac{\pi(\mathbf{y})\, q(\mathbf{y}, \mathbf{x})}{\pi(\mathbf{x})\, q(\mathbf{x}, \mathbf{y})}\right\}.$$

However, if we reject the proposed point, then the chain remains in the current state.

Note that this form of acceptance probability is not unique, since there may be many acceptance functions which provide a chain with the desired properties. However, Peskun (1973) shows that this form is optimal in that suitable candidates are rejected least often and so statistical efficiency is maximized.

The resulting transition kernel $\mathcal{P}(\mathbf{x}, A)$, which denotes the probability that the next state of the chain lies within some set A, given that the chain is currently in state $\mathbf{x}$, is given by

$$\mathcal{P}(\mathbf{x}, A) = \int_A \mathcal{K}(\mathbf{x}, \mathbf{y})\, \mathrm{d}\mathbf{y} + r(\mathbf{x})\, I_A(\mathbf{x}),$$

where

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = q(\mathbf{x}, \mathbf{y})\, \alpha(\mathbf{x}, \mathbf{y}),$$

i.e. the density associated with selecting a point which is accepted, and

$$r(\mathbf{x}) = 1 - \int_E q(\mathbf{x}, \mathbf{y})\, \alpha(\mathbf{x}, \mathbf{y})\, \mathrm{d}\mathbf{y}, \tag{5}$$

i.e. the size of the point mass associated with a rejection. Here, I_A denotes the indicator function, and $\mathcal{K}$ satisfies the reversibility condition of equation (4), implying that the kernel $\mathcal{P}$ also preserves detailed balance for π.

$\pi(\cdot)$ only enters $\mathcal{K}$ through α and the ratio $\pi(\mathbf{y})/\pi(\mathbf{x}^t)$, so knowledge of the distribution only up to a constant of proportionality is sufficient for implementation. Also, in the case where the candidate generating function is symmetric, i.e. $q(\mathbf{x}, \mathbf{y}) = q(\mathbf{y}, \mathbf{x})$, the acceptance function reduces to

$$\alpha(\mathbf{x}, \mathbf{y}) = \min\left\{1, \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}\right\}. \tag{6}$$

This special case is the original Metropolis update of Metropolis *et al.* (1953). There is an infinite range of choices for q; see Tierney (1994) and Chib and Greenberg (1995), for example. However, we restrict our attention to only a few special cases here.

2.2.1. *Random walk Metropolis updating*

If $q(\mathbf{x}, \mathbf{y}) = f(\mathbf{y} - \mathbf{x})$ for some arbitrary density f, then the kernel driving the chain is a random walk, since the candidate observation is of the form $\mathbf{y}^{t+1} = \mathbf{x}^t + \mathbf{z}$, where $\mathbf{z} \sim f$. There are many common choices for f, including the uniform distribution on the unit disc, a multivariate normal or a t-distribution. These are symmetric, so the acceptance probability is of the simple form given in equation (6).

2.2.2. *The independence sampler*

If $q(\mathbf{x}, \mathbf{y}) = f(\mathbf{y})$, then the candidate observation is drawn independently of the current state of the chain. In this case, the acceptance probability can be written as

$$\alpha(\mathbf{x}, \mathbf{y}) = \min\left\{1, \frac{w(\mathbf{y})}{w(\mathbf{x})}\right\},$$

where $w(\mathbf{x}) = \pi(\mathbf{x})/f(\mathbf{x})$ is the importance weight function that would be used in importance sampling given observations generated from f. In some senses, the ideas of importance sampling and the independence sampler are closely related. The essential difference between the two is that the importance sampler builds probability mass around points with large weights, by choosing those points relatively frequently. In contrast, the independence sampler builds up probability mass, on points with high weights, by remaining at those points for long periods of time.

2.2.3. *The Gibbs sampler*

The Gibbs transition can be regarded as a special case of a Metropolis–Hastings transition, as follows. Suppose that we wish to update the ith element of $\mathbf{x}$; then we can choose proposal q, so that

$$q(\mathbf{x}, \mathbf{y}) = \begin{cases} \pi(\mathbf{y}_i|\mathbf{x}_{(i)}) & \mathbf{y}_{(i)} = \mathbf{x}_{(i)},\ i = 1, \ldots, k, \\ 0 & \text{otherwise.} \end{cases}$$

With this proposal, the corresponding acceptance probability is given by

$$\begin{aligned}
\alpha(\mathbf{x}, \mathbf{y}) &= \frac{\pi(\mathbf{y})\, q(\mathbf{y}, \mathbf{x})}{\pi(\mathbf{x})\, q(\mathbf{x}, \mathbf{y})} \\
&= \frac{\pi(\mathbf{y})/\pi(\mathbf{y}_i|\mathbf{x}_{(i)})}{\pi(\mathbf{x})/\pi(\mathbf{x}_i|\mathbf{y}_{(i)})} \\
&= \frac{\pi(\mathbf{y})/\pi(\mathbf{y}_i|\mathbf{y}_{(i)})}{\pi(\mathbf{x})/\pi(\mathbf{x}_i|\mathbf{x}_{(i)})}, \qquad \text{since } \mathbf{y}_{(i)} = \mathbf{x}_{(i)}, \\
&= \frac{\pi(\mathbf{y}_{(i)})}{\pi(\mathbf{x}_{(i)})}, \qquad \text{by definition of conditional probability for } \boldsymbol{\theta} = (\boldsymbol{\theta}_i, \boldsymbol{\theta}_{(i)}), \\
&= 1, \qquad \text{since } \mathbf{y}_{(i)} = \mathbf{x}_{(i)}.
\end{aligned}$$

Thus, at each step, the only possible jumps are to states $\mathbf{y}$ that match $\mathbf{x}$ on all components other than the ith, and these are automatically accepted. Hence, it is clear that the resulting transition function is of the same form as that of the Gibbs transition.

2.3. Combining kernels

As we discussed earlier it is common, in practice, to combine a number of different transition kernels within a single chain to improve desirable aspects of convergence. Given an appropriate set of Markov kernels $\mathcal{K}_1, \ldots, \mathcal{K}_m$, Tierney (1994) showed that, under weak conditions on the $\mathcal{K}_i$, a hybrid algorithm consisting of *mixtures* or *cycles* of these kernels may be combined to produce a chain with the desired stationary distribution. In a mixture (or random sweep), probabilities $\alpha_1, \ldots, \alpha_m$ are specified and, at each step, one of the kernels is selected according to these probabilities whereas, in a cycle (or systematic sweep), each of the kernels is used in a predetermined sequence, which is repeated until the sampler is stopped. Thus, we can combine different transition kernels within a single algorithm to form a chain with stationary distribution π.

Having split the state vector into components, a cycle of a number of different types of component updates, which themselves are not necessarily sufficient to produce a chain with stationary distribution π, may be combined to form a single iteration of an MCMC sampler which does. In many cases, it is natural to work with a complete breakdown of the state space into scalar components, so that $k = p$. However, the convergence rate of the resulting chain may often be improved by *blocking* highly correlated variables and updating them together, as discussed in Roberts and Sahu (1997).

The Gibbs sampler is an example of a component-based MCMC algorithm, since it uses a fixed sequence of Gibbs transition kernels each of which updates a different component of the state vector, as follows. Given an arbitrary starting value $\mathbf{x}^0 = (\mathbf{x}_1^0, \ldots, \mathbf{x}_k^0)$, the Gibbs sampler proceeds by systematically updating each variable in turn, via a single Gibbs update, as follows:

$$\begin{array}{lll}
\mathbf{x}_1^1 & \text{is sampled from} & \pi(\mathbf{x}_1|\mathbf{x}_{(1)}^0); \\
\mathbf{x}_2^1 & \text{is sampled from} & \pi(\mathbf{x}_2|\mathbf{x}_1^1, \mathbf{x}_3^0, \ldots, \mathbf{x}_k^0); \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\mathbf{x}_i^1 & \text{is sampled from} & \pi(\mathbf{x}_i|\mathbf{x}_j^1, j<i, \mathbf{x}_j^0, j>i); \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\mathbf{x}_k^1 & \text{is sampled from} & \pi(\mathbf{x}_k|\mathbf{x}_{(k)}^1).
\end{array}$$

This completes a transition from $\mathbf{x}^0$ to $\mathbf{x}^1$.

Iteration of the full cycle of random variate generations from each of the full conditionals in turn produces a sequence $\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^t, \ldots$ which is a realization of a Markov chain with transition density for going from $\mathbf{x}^t$ to $\mathbf{x}^{t+1}$ given by

$$\mathcal{K}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \prod_{l=1}^{k} \pi(\mathbf{x}_l^{t+1}|\mathbf{x}_j^{t+1}, j<l, \text{ and } \mathbf{x}_j^t, j>l), \tag{7}$$

and stationary distribution π; see Smith and Roberts (1993), for example. This sampling algorithm, where each component is updated in turn, is sometimes referred to as the systematic sweep Gibbs sampler. However, the Gibbs transition kernels need not be used in this systematic manner, and many other implementations are possible, such as the *random sweep* Gibbs sampler, which randomly selects a component to update at each iteration, and thus uses a mixture (rather than a cycle) of Gibbs updates. The fact that Gibbs transitions are a special case of Metropolis–Hastings updates makes it clear that we may also use more general Metropolis–Hastings transitions within this updating framework.

One particularly useful consequence of this observation is that Metropolis–Hastings steps may be introduced into the Gibbs sampler, so that components whose conditional distributions are of a standard form may be sampled directly from the full conditional, whereas those with non-standard distributions are updated via a Metropolis–Hastings step, as discussed in Tierney (1994). Generally, this is simpler to implement than the alternative methods discussed at the end of Section 2.1 but may sometimes result in a 'slow' Markov chain, because of the rejection of Metropolis proposals, restricting 'movement' around the state space in these directions.

We do have some restrictions on the way in which different transition kernels can be combined. For example, we must retain the properties of aperiodicity and irreducibility for the resulting chain. This may often be achieved by ensuring that each component would be updated infinitely often if the chain were continued indefinitely. In addition, we may also lose the property of reversibility by using certain combinations of transition kernels. For example the random scan Gibbs sampler is reversible, whereas the systematic scan Gibbs sampler is not. However, if we systematically update each component in turn and then repeat the same procedure backwards, we obtain the *reversible Gibbs sampler* which, of course, is reversible. See Roberts and Sahu (1997) for further discussion on this topic.

3. Practical implementation issues

Having examined the basic building-blocks for standard MCMC samplers, we now discuss issues associated with their implementation. An excellent discussion of some of these issues is provided by Kass *et al.* (1997).

3.1. How many iterations?

One practical problem associated with using MCMC methods is that, to reduce the possibility of inferential bias caused by the effect of starting values, iterates within an initial transient phase or *burn-in* period are discarded. One of the most difficult implementational problems is that of determining the length of the required burn-in, since rates of convergence of different algorithms on different target distributions may vary considerably.

Ideally, we would like to compute analytically or to estimate a convergence rate and then to take sufficient iterations for any particular desired accuracy. For example, given geometric ergodicity, in which case the t-step transition distribution $\mathcal{P}^t(\mathbf{x}, \cdot)$ is such that

$$|\mathcal{P}^t(\mathbf{x}, \cdot) - \pi(\cdot)| \leqslant M(\mathbf{x})\rho^t,$$

for some $M, \rho \in \mathbb{R}$, we might stop the chain once $|\mathcal{P}^t(\mathbf{x}, \cdot) - \pi(\cdot)| \leqslant \epsilon$, for some $\epsilon > 0$, in which case the length of the burn-in period is given by

$$t^* = \frac{\log\{\epsilon/M(\mathbf{x})\}}{\log(\rho)}.$$

However, in general, it is extremely difficult to prove even the existence of a geometric rate of convergence to stationarity (Roberts and Tweedie, 1996) and there are many commonly used algorithms which frequently fail to converge geometrically quickly at all.

Roberts and Polson (1994), Chan (1993), Schervish and Carlin (1992) and Roberts and Tweedie (1996) provide qualitative geometric convergence results for quite general classes of target densities. Although these results have theoretical import in ensuring the existence of central limit theorems, they do not offer explicit bounds on the rate of convergence, which could be used to determine MCMC sample lengths. However, there are some methods which *do* provide bounds on convergence rates.

Diaconis and Stroock (1991) proposed a bound based on the Poincaré inequality, which may be used in cases where the state space of the chain is discrete. In essence, this method gains a measure of the 'flow' of the chain from state to state to bound the convergence rate. Lawler and Sokal (1988) provided an estimate of the convergence rate of a continuous state space Markov chain, based on Cheeger's inequality. They bounded the convergence rate by a measure of probability flow from a given set to its complement. Sinclair and Jerrum (1988) provided the same bound, motivated in the context of *capacitance* from the physics literature. Similar computable bounds have also been derived by Meyn and Tweedie (1994). However, for complex target distributions, the approximations that are necessary for any of these bounds to be analytically tractable lead to bounds that are too weak to be of much practical value.

Finally, Rosenthal (1995a) provided a bound on the convergence rate of a continuous state space Markov chain, based on a coupling of two independent chains. This method is somewhat more practical than other methods but can require a substantial amount of analytical work to gain the appropriate bound on the convergence rate. This problem is addressed by Cowles and Rosenthal (1996) who discussed various numerical approximation techniques, but these are generally computationally expensive to compute. However, this method has been applied to many problems; see Rosenthal (1993a, b, c).

It is possible to construct less general bounds, applicable to only a small class of samplers. For example, Amit and Grenander (1991), Amit (1991) and Roberts and Sahu (1997) all proposed bounds which can be used in the case where the target distribution is Gaussian (or approximately Gaussian). Similarly, Polson (1996) discussed approaches to the estimation of the convergence rate in the case where the target distribution is log-concave. Finally, Liu (1994) provided a method for exact calculation of the rate of convergence for the independence sampler. If the importance weight $w(\mathbf{x})$ has a finite maximum w^* for some $\mathbf{x} = \mathbf{x}^* \in E$, then the convergence rate of the chain is given by $\rho = 1 - 1/w^*$. Thus, the independence sampler converges geometrically to π, whenever $w^* < \infty$. This result is further discussed in Mengersen and Tweedie (1995) and extended in Smith and Tierney (1996) and is particularly easy to use. However, no similar results exist for more general samplers such as the Gibbs sampler or the Metropolis–Hastings algorithm.

Since general, practical, bounds on the convergence rate are rarely available, a large amount of work has been performed on the statistical analysis of sampler output to tell, either *a posteriori* or during run time, whether or not the chain converges during a particular sample run. Such techniques are known as convergence diagnostics and use the sample path of the chain, together with any other available information, to try to determine how long the chains should be allowed to run. In practice, such diagnostics tend to be of limited use since, for example, Asmussen *et al.* (1992) showed that no one diagnostic method will work for all problems. However, some very useful methods have been proposed.

There are various informal methods for the diagnosis of convergence of MCMC algorithms. Examples include Gelfand's *thick pen technique* (Gelfand *et al.*, 1990) and the use of quantile and autocorrelation plots, as suggested by Gelfand and Smith (1990). There are also other *ad hoc* methods proposed in the physics and computer science literature; see Kimbler and Knight (1987) and Gafarian *et al.* (1978), for example. In general, these informal methods are easy to implement and can provide a feel for the behaviour of the Markov chain.

However, various, more elaborate, methods have been proposed in the literature. See Cowles and Carlin (1996), Robert (1996) and Brooks and Roberts (1996) for reviews. These include eigenvalue estimation techniques, such as those proposed by Raftery and Lewis (1992) and Garren and Smith (1995), which attempt to estimate the convergence rate of the sampler via appropriate eigenvalues. Gelman and Rubin (1992) proposed a diagnostic based on a classical analysis of variance to estimate the utility of continuing to run the chain. This method has subsequently been extended by Brooks and Gelman (1997). Geweke (1992) and Heidelberger and Welch (1983) used spectral density estimation techniques to perform hypothesis tests for stationarity, whereas Yu and Mykland (1997) and Brooks (1996) used cumulative sum path plots to assess convergence.

Convergence diagnostics that include additional information beyond the simulation draws themselves include weighting-based methods proposed by Ritter and Tanner (1992) and Zellner and Min (1995), and the kernel-based techniques of Liu *et al.* (1993) and Roberts (1994), which attempt to estimate the L^2-distance between the t-step transition kernel and the stationary distribution. Similarly, Yu (1995) and Brooks *et al.* (1997) provided methods for calculating the L^1-distances between relevant densities. These methods have the advantage that they assess convergence of the full joint density. However, in general, they have proved computationally expensive to implement and difficult to interpret.

There are also several methods which, although not actually diagnosing convergence, attempt to 'measure' the performance of a given sampler. These can be used either instead of or in addition to the other methods described above. Such methods include those described by Mykland *et al.* (1995) and Robert (1996), which make use of Markov chain splitting ideas to introduce regeneration times into MCMC samplers. Mykland *et al.* (1995) showed how monitoring regeneration rates can be used as a diagnostic of sampler performance. They argued that high regeneration rates, and regeneration patterns which are close to uniform, suggest that the sampler is working well, whereas low regeneration rates may indicate dependence problems with the sampler. Similarly, Robert (1996) suggested monitoring several estimates of the asymptotic variance of a particular scalar functional, each of which is based on a different *atom* of the regenerative chain. An alternative performance assessment technique was suggested by Brooks (1998), who provided a bound on the proportion of the state space covered by an individual sample path, in the case where the target density is log-concave.

3.2. *How many more iterations?*

In any practical application of the MCMC method, it is necessary to determine how long the simulations need to be run. Since the sampler output is generally used as a basis for inference, the number of iterations required will be dependent on the problem in hand. However, computation time and storage limits may also be a consideration. For example, it may often be necessary to *thin* the observations by saving only every kth observation, for some suitable value of k, to reduce these computational 'overheads'.

For an independent and identically distributed (IID) sample of size n from a posterior distribution π, the standard deviation of the sample mean of a scalar functional $\theta(\mathbf{X})$ is $\sigma/\sqrt{n}$, where σ is the posterior standard deviation of θ. If a preliminary estimate of σ is available, perhaps from some pilot run of the chain, then this can be used to estimate the sample size that would be required for an IID sample.

Alternatively, if the series can be approximated by a first-order autoregressive process, then the asymptotic standard deviation of the sample mean is given by

$$\frac{\sigma}{\sqrt{n}}\sqrt{\left(\frac{1+\rho}{1-\rho}\right)},$$

where ρ is the autocorrelation of the series $\theta(\mathbf{X}^t)$, and this may be used to estimate the sample size that is required.

An alternative approach was suggested by Raftery and Lewis (1992), who proposed a method for determining how long the sampler should be run, based on a short pilot run of the sampler. They used the central limit theorem to justify the assumption that the ergodic average, given by

$$\overline{\theta}_n = \frac{1}{n}\sum_{t=1}^{n} \theta(\mathbf{x}^t) \overset{n\rightarrow\infty}{\longrightarrow} \theta \, .$$

for some functional $\theta(\cdot)$, will be asymptotically normal for large n. They then estimated the value of n that is necessary to ensure that

$$\mathbb{P}[\theta - \epsilon \leqslant \overline{\theta}_n \leqslant \theta + \epsilon] = 1 - \alpha,$$

for some $\epsilon > 0$ and $0 < \alpha < 1$. This suggests taking

$$n \approx \widehat{\mathrm{var}}(\overline{\theta})\left\{\frac{\Phi^{-1}(1-\alpha/2)}{\epsilon}\right\}^2,$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function and $\widehat{\mathrm{var}}(\overline{\theta})$ is an estimate of the variance of the ergodic average, based on the observations gained from the pilot run. Geyer (1992) has provided a detailed discussion of such issues.

3.3. One run or many?

One of the most contentious issues associated with the implementation of MCMC algorithms is in choosing whether to run one long chain or several shorter chains in parallel; see Gelman and Rubin (1992) and Geyer (1992). The argument for taking a single long run is that the chain will be 'closer' to the target distribution at the end of one long run than it would be at the end of any number of shorter runs, and that several shorter runs may be wasteful, in that initial 'warm-up' periods must be discarded from each. In contrast, proponents of the 'many replications' approach argue that, although a single run will eventually cover the entire sample space, by taking a number of parallel replications, we can guard against a single chain leaving a 'significant proportion' of the sample space unexplored. By taking several chains, each started in different states, it is also possible to monitor the sample paths to determine how well the chains have *mixed*, i.e. to what extent the outputs from the different chains are indistinguishable. In essence, multiple replications protect against bias by attempting to ensure that the sampler output covers the entire sample space, whereas one long run provides less variable estimates, since ergodic averages are based on a larger sample. However, the one-sided nature of the reassurance provided by multiple runs should be noted, since only poor performance can be truly detected.

An alternative to running multiple replications is to use regenerative methods to *restart* the chain at appropriate *regeneration times*. Implementational details of an approach to regenerative simulation, obtained by interspersing the Gibbs sampler with steps from the independence sampler, are described in Mykland *et al.* (1995). In this manner, a single long chain may be run, and then split, to produce multiple replications which will be closer to the stationary distribution than would many independent shorter chains, since they consist of observations taken from the end of a single long run. However, one advantage of the use of traditional parallel runs is that, in certain applications, the implementation can take advantage of parallel computing technology, whereas the computation of regenerations can be both difficult and expensive.

3.4. Starting point determination

Another important issue is the determination of suitable starting points. Of course, any inference gained via MCMC sampling will be independent of the starting values, since observations are only used after the chain has achieved equilibrium, and hence lost all dependence on those values. However, the choice of starting values may affect the performance of the chain and, in particular, the speed and ease of detection of convergence. If we choose to run several replications in parallel, Gelman and Rubin (1992) argued that, to detect convergence reliably, the distribution of starting points should be overdispersed with respect to the target distribution. Several methods have been proposed for generating initial values for MCMC samplers.

Many users adopt *ad hoc* methods for selecting starting values. Approaches include simplifying the model, by setting hyperparameters to fixed values, or ignoring missing data, for example. Alternatively, maximum likelihood estimates may provide starting points for the chain or, in the case where informative priors are available, the prior might also be used to select suitable starting values. However, more rigorous methods are also available.

Gelman and Rubin (1992) proposed a simple mode-finding algorithm to locate regions of high density and sampling from a mixture of t-distributions located at these modes to generate suitable starting values. An alternative approach is to use a *simulated annealing* algorithm to sample initial values. Brooks and Morgan (1994) described the use of an annealing algorithm to generate 'well-dispersed' starting values for an optimization algorithm, and a similar approach may be used to generate initial values for MCMC samplers. See also Applegate *et al.* (1990) for the use of a simulated annealing algorithm in this context.

3.5. Alternative parameterizations

Another important implementational issue is that of *parameterization*. As we discussed in the context of the Gibbs sampler, high correlations between variables can slow down the convergence of an MCMC sampler considerably. Thus, it is common to reparameterize the target distribution so that these correlations are reduced; see Hills and Smith (1992), Robert and Mengersen (1995) and Gilks and Roberts (1996), for example.

Approximate orthogonalization has been recommended to provide uncorrelated posterior parameters. However, in high dimensions, such approximations become computationally infeasible. An alternative approach, called *hierarchical centring*, is given by Gelfand *et al.* (1995) and involves introducing extra layers in a hierarchical model to provide a 'better behaved posterior surface'. However, such methods may only be applied to specific models with a suitable linear structure, as discussed in Roberts and Sahu (1997). It should also be noted that by reparameterizing it is sometimes possible to ensure that the full conditionals conform to standard distributions; see Hills and Smith (1992). This simplifies the random variate generation in the Gibbs sampler, for example.

3.6. Auxiliary variables

A related issue is that of the introduction of *auxiliary variables* to improve convergence, particularly in the case of highly multimodal target densities. The idea of using auxiliary variables to improve MCMC performance was first introduced in the context of the Ising model by

Swendsen and Wang (1987). Their idea has been subsequently discussed and generalized by many researchers, including Edwards and Sokal (1988), Besag and Green (1993), Higdon (1996), Damien *et al.* (1997), Mira and Tierney (1997) and Roberts and Rosenthal (1997).

Basically, auxiliary variables techniques involve the introduction of one or more additional variables $\mathbf{u} \in U$ to the state variable $\mathbf{x}$ of the Markov chain. In many cases, these additional (or auxiliary) variables have some physical interpretation, in terms of temperature or some unobserved measurement, but this need not be the case. For MCMC simulation, the joint distribution of $\mathbf{x}$ and $\mathbf{u}$ is defined by taking $\pi(\mathbf{x})$ to be the marginal for $\mathbf{x}$ and specifying the conditional $\pi(\mathbf{u}|\mathbf{x})$ arbitrarily. We then construct a Markov chain on $E \times U$, which (at iteration t) alternates between two types of transition. First, $\mathbf{u}^{t+1}$ is drawn from $\pi(\mathbf{u}|\mathbf{x}^t)$. Then, the new state of the chain, $\mathbf{x}^{t+1} = \mathbf{y}$, is generated by any method which preserves detailed balance for the conditional $\pi(\mathbf{x}|\mathbf{u})$, i.e. from some transition function $p(\mathbf{x}^t, \mathbf{y})$, such that

$$\pi(\mathbf{x}|\mathbf{u})p(\mathbf{x}, \mathbf{y}) = \pi(\mathbf{y}|\mathbf{u})p(\mathbf{y}, \mathbf{x}).$$

In this way, we generate observations $\{\mathbf{x}^t\}$, with stationary distribution π, overcoming problems associated with slow convergence and multimodality, for example. In practice, all that we are doing is providing a chain on $E \times U$ which converges to some distribution, such that the marginal distribution for $\mathbf{x}$ is π. Thus, we simply run the chain on the augmented state space and retain only the $\mathbf{x}$-values observed, discarding the $\mathbf{u}^t$.

In general, rather little is known about the theoretical properties of auxiliary variables techniques, though many researchers provide examples where the mixing of the chain is improved by adding additional variables to the state space. One exception to this is the technique known as *slice sampling*, most recently discussed by Roberts and Rosenthal (1997) and Mira and Tierney (1997), and previously by Besag and Green (1993) and Edwards and Sokal (1988). The general idea is that if the target distribution can be written as a product of a finite number of densities, i.e.

$$\pi(\mathbf{x}) = \prod_{i=0}^{k} f_i(\mathbf{x}),$$

then the following algorithm will produce a Markov chain with stationary distribution π. Given $\mathbf{x}^t$, we sample k independent uniform random variables, $u_i^{t+1} \sim U\{0, f_i(\mathbf{x}^t)\}$, $i = 1, \ldots, k$, and then generate the new state of the chain by sampling from the density proportional to $f_0(\mathbf{x})\, I_{L(\mathbf{u}^{t+1})}(\mathbf{x})$, where

$$L(\mathbf{u}^{t+1}) = \{\mathbf{x}\colon f_i(\mathbf{x}) \geqslant u_i,\ i = 1, \ldots, k\},$$

and $I_{(A)}(\cdot)$ denotes the indicator function for the set A. Roberts and Rosenthal (1997) provided convergence results for several different implementations of this scheme. They showed that slice sampling schemes usually converge geometrically and also provided practical bounds on the convergence rate for quite general classes of problems. Some applications of such schemes are provided by Damien *et al.* (1997).

3.7. Graphical models and modelling

Graphical models and graphical representations are playing an increasingly important role in statistics, and in Bayesian statistical modelling in particular; see Whittaker (1990), Almond (1995), Edwards (1995) and Lauritzen (1996). Relationships between variables in a model can be represented graphically by letting *nodes* in a graph represent those variables and *edges* between nodes represent the presence, or otherwise, of a direct relationship, defined in terms of conditional independence, between them. Such graphs are usually presented as a hierarchical structure, with those variables which exert the most direct influence on the data placed closest to the bottom, and those of lesser influence placed in decreasing order up the graph. Such graphs provide simple representations of the conditional independence structure of the model, simplifying the implementation of MCMC algorithms by indicating which other variables feature in the full conditional distribution of any given variable. Such models have been used for a long time, but their representation via graphs of this sort is relatively new and is becoming increasingly common in the MCMC literature; see Lauritzen and Spiegelhalter (1988), Berzuini *et al.* (1997), Madigan and York (1995) and Spiegelhalter *et al.* (1996a). We begin with the concept of a directed acyclic graph (DAG).

A directed graph consists of a collection of nodes and directed edges, where the direction of an edge between two nodes represents the direction of the relationship between the two corresponding variables. Nodes may be represented in two ways: either as a circle, denoting that the value of the corresponding variable is unknown and thus subject to estimation, or by a square in which case the value of that variable is known. Thus, observed data and prior or hyperprior parameters are often represented by square nodes.

Cycles in the graph occur when there is a set of nodes which can be reached in sequence by traversing directed edges, such that the first and final nodes in the set are the same. A directed graph in which there are no cycles is called a DAG. A typical (if extremely simple) DAG may look something like that of Fig. 1. Here, nodes A and B are referred to as the *parents* of C. Similarly, node D is a *child* of C. In addition, nodes A and B may be referred to as *spouses* in that, together, they produce (or inform) a child C.

Fig. 1 represents the belief that parameter C depends on the values of the parameters A and B. Similarly, parameter D depends on the value of parameter C but is *conditionally independent* of the values of A and B, given the value of C. This conditional independence between D and parameters A and B is represented in the graph by the absence of an edge between nodes A and D and between B and D. In general terms, the graph tells us that, given the value of C, knowing the values of A and B provides no extra information about parameter D. We write

$$D \perp A, B|C,$$

In general, we have that for any particular node v

$$v \perp \text{non-descendants of } v|\text{parents of } v.$$

This implies that the joint distribution of both the parameters in the model, $v \in V$ say, and the data, can be factorized as

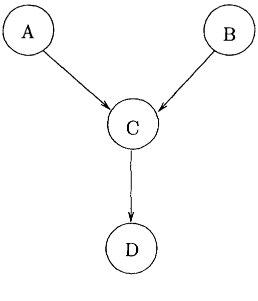

Fig. 1. Simple DAG

$$\begin{aligned} p(V) &= \prod_{v \in V} p(v|\text{parents of } v) \\ &= p(A)\, p(B)\, p(D|C)\, p(C|A, B). \end{aligned}$$

Thus, the DAG is equivalent to the factorization assumption that allows us to decompose the full joint distribution into smaller components.

If we are to use the MCMC method to fit such a model, then the DAG does not immediately provide us with all the information that we require. For example, if we knew C and were interested in estimating A, the value of B would be of use in constructing that estimate. Hence, A is not conditionally independent of B, given C, and thus B features in the conditional distribution of A, given the other parameters. If we intend to use MCMC sampling to fit models of this sort, then we require a complete specification of the dependence structure between all variables. This can be obtained via the conditional independence graph.

The conditional independence graph is obtained by *moralizing* the DAG. This involves *marrying* parents by introducing an edge between the parents of each node v in the graph. We then drop the directions of all edges to obtain the conditional independence graph. This graph provides us with all information regarding the relationships between the parameters in the model. For example, given a particular node v, the full conditional distribution of that node, given the value of all the others, can be expressed as the product

$$p(v|\text{rest}) \propto p(v|\text{parents of } v) \prod_{u \in C_v} p(u|\text{parents of } u),$$

where the set C_v denotes the set of children of node v.

Thus, the conditional independence graph makes it easy to see which other parameters are required in the specification of the full conditional distributions for each parameter, and thus simplifies the implementation of MCMC algorithms for such models. See the example in Section 5.2 and, in particular, Fig. 3 later for an example of a DAG and its corresponding conditional independence graph, together with a discussion of their usefulness in implementing MCMC algorithms for problems of this sort.

3.8. Software

Despite the widespread use of the MCMC method within the Bayesian statistical community, surprisingly few programs are freely available for its implementation. This is partly because algorithms are generally fairly problem specific and there is no automatic mechanism for choosing the best implementation procedure for any particular problem. However, one program has overcome some of these problems and is widely used by many statistical practitioners. The program is known as BUGS; see Gilks *et al.* (1992) and Spiegelhalter *et al.* (1996a, b).

Currently restricted to the Gibbs sampler, the BUGS package can implement MCMC methodology for a wide variety of problems, including generalized linear mixed models with hierarchical, crossed, spatial or temporal random effects, latent variable models, frailty models, measurement errors in responses and covariates, censored data, constrained estimation and missing data problems.

The package provides a declarative S-PLUS-type language for specifying statistical models. The program then processes the model and data and sets up the sampling distributions required for Gibbs sampling. Finally, appropriate sampling algorithms are implemented to simulate values of the unknown quantities in the model. BUGS can handle a wide range of standard distributions, and in the cases where full conditional distributions are non-standard but log-concave the method of adaptive rejection sampling (Gilks and Wild, 1992) is used to sample these components. In the latest version (6), Metropolis–Hastings steps have also been introduced for non-log-concave conditionals, so that the package is no longer restricted to problems where the target density is log-concave. In addition, a stand-alone and menu-driven set of S-PLUS functions (CODA) is supplied with BUGS to calculate convergence diagnostics and both graphical and statistical summaries of the simulated samples.

Overall the package is extremely easy to use and is capable of implementing the Gibbs sampler algorithm for a wide range of problems. It is also freely available from the BUGS Web site at

`http: //www.mrc-bsu.cam.ac.uk/bugs/Welcome.html,`

which has led to its being an extremely popular tool, widely used among practising statisticians.

Thus, there is a wide variety of MCMC algorithms which have been developed in the literature. Although many practitioners have their favourite sampler, it is important to understand that each has its own distinct advantages and drawbacks with respect to the others. Thus, the decision about which form of sampler to use for any particular problem should always be a considered one, based on the problem in hand.

4. Alternative updating schemes

Having discussed the most common forms of transition kernel, and various issues associated with the implementation of MCMC methods, we now discuss some new forms of transition kernel, which attempt to overcome problems associated with the standard updating schemes that were discussed in Section 2.

4.1. Simulated tempering and Metropolis-coupled Markov chain Monte Carlo methods

To overcome problems associated with slow mixing Markov chains, which become stuck in local modes, alternative methods based on the introduction of transition kernels with 'flatter' stationary distributions have been proposed.

Simulated tempering (Marinari and Parisi, 1992) and Metropolis-coupled MCMC methods (Geyer, 1991) are similar methods based on the idea of using a series of kernels $\mathcal{K}_1, \ldots, \mathcal{K}_m$, with corresponding (unnormalized) stationary densities $\pi_1, \ldots, \pi_m$. We take $\pi_1 = \pi$ and then the other kernels are chosen to progress in steps between two extremes: the 'cold' distribution π_1 and the 'hot' distribution π_m. For example, Geyer and Thompson (1995) suggested taking

$$\pi_i(x) = \pi(x)^{1/i}, \qquad i = 1, \ldots, m,$$

in which case π_∞ would correspond to a uniform distribution over the entire parameter space.

Having specified these kernels, together with their corresponding equilibrium densities, the Metropolis-coupled MCMC algorithm works as follows. We run m chains $\{\mathbf{X}_i^t\}$, $i = 1, \ldots, m$, where updates in chain i use transition kernel $\mathcal{K}_i$. At each iteration, after having updated each of the chains, we then select two chains and attempt to swap the states of these two chains. Suppose that we select chains i and j, at time t; then we propose the swap $\mathbf{Y}_i^t = \mathbf{X}_j^t$ and $\mathbf{Y}_j^t = \mathbf{X}_i^t$. With probability

$$\min\left\{1, \frac{\pi_i(\mathbf{Y}_i^t)\,\pi_j(\mathbf{Y}_j^t)}{\pi_i(\mathbf{X}_i^t)\,\pi_j(\mathbf{X}_j^t)}\right\}, \tag{8}$$

we accept the swap so that $\mathbf{X}_i^t = \mathbf{Y}_i^t$ and $\mathbf{X}_j^t = \mathbf{Y}_j^t$, i.e. the chains swap states. As a basis for inference, we use only the sample path of the chain with the correct stationary density, i.e. $\{X_1^t: t = 0, 1, \ldots\}$.

This method essentially allows two types of update. One draws observations from some density $\mathcal{K}_i$, and the second is based on a proposal generated from the potential swapping of states between chains. The acceptance probability (8) ensures that this second type of update preserves the stationary density.

The advantage of the second update is that the warmer distributions each mix progressively more rapidly than the cold distribution which is of primary interest. By allowing the chains to swap states we improve the mixing rate of the cold chain, by allowing for 'jumps' between points in the state space which might otherwise be very unlikely under the transition density $\mathcal{K}_1$. Thus, the parameter space is traversed more rapidly and mixing is improved. The drawback, of course, is that we need to run m chains in parallel, whereas only the output from one is used as a basis for inference. Thus, the method may be considered computationally inefficient in this respect.

Simulated tempering is based on a similar idea but, rather than running m chains, we run only one and randomly swap between transition densities. Geyer and Thompson (1995) suggested setting $p_{i,i+1} = p_{i,i-1} = \frac{1}{2}$, $i = 2, \ldots, m-1$, and $p_{1,2} = p_{m,m-1} = 1$, where $p_{i,j}$ is the probability of proposing that at any iteration the chain should move from sampler i to sampler j. This proposal is then accepted with probability

$$\min\left\{1, \frac{\pi_j(\mathbf{X}^t) c_j p_{j,i}}{\pi_i(\mathbf{X}^t) c_i p_{i,j}}\right\}, \tag{9}$$

where the c_i, $i = 1, \ldots, m$, are approximate normalization constants for the densities π_i, $i = 1, \ldots, m$. Thus, the temperature or sampler index follows a random walk on the integers $1, \ldots, m$. It is not essential that a random walk is used, but there has been very little work on alternative distributions.

Given a sample path from the resulting chain, we base our inference on all observations which were drawn from the cold sampler, i.e. with transition density $\mathcal{K}_1$, and discard the remaining observations.

One of the main drawbacks of this method is the estimation of the normalization constants in expression (9). Several methods have been proposed for their estimation, such as reverse logistic regression via a Metropolis-coupled MCMC chain, stochastic approximation techniques and various *ad hoc* methods; see Geyer (1990) and Geyer and Thompson (1995). Alternatively, it may be possible to estimate these normalization constants from the sample path of the chain as it progresses, by noting that the prior distribution for the temperature index is also the marginal for that parameter in the joint distribution, sampled by the chain (Green, personal communication). This observation allows us to construct estimates of the c_i up to a multiplicative constant, which is all that we require in expression (9).

The other drawback with the method is the inefficiency associated with discarding all those observations generated from the 'warmer' transition densities. As explained above, the Metropolis-coupled MCMC algorithm is also wasteful, but it has the advantage that it may be implemented on a parallel processing machine, greatly reducing the run time of the algorithm. Thus, the Metropolis-coupled MCMC algorithm might generally be preferred over simulated tempering.

4.2. *Continuous time processes*

Continuous time Markov processes were motivated by the need to produce 'intelligent' and problem-specific proposal distributions for the Metropolis–Hastings algorithm. Many methods come from the physics literature, where the idea of energy gradients are used to choose the most suitable directions for updating. See Neal (1993) for an excellent review of such methods. The essential idea of continuous time Markov chains is that jumps from state to state occur at random times which are distributed exponentially. Inference from such chains is gained by recording successive states of the chain and weighting these values by the time spent in each state. However, such methods have yet to become popular in the MCMC literature.

There has, however, been some interest in the use of chains which approximate continuous time Markov chains. One approach to the problem of designing problem-specific proposal distributions for Metropolis–Hastings algorithms is to use what are known as Langevin algorithms. Such algorithms (which are derived from approximated diffusion processes) use information about the target distribution π in the form of the gradient of $\log(\pi)$, to generate a candidate observation for the next state of the chain.

To describe these algorithms, let us first describe the underlying diffusion process. Consider a k-dimensional continuous time Markov chain, whose state at time t is denoted by $\mathbf{X}^t$, and which has stationary distribution π. We describe the diffusion process in terms of a stochastic differential equation, of the form

$$\mathrm{d}\mathbf{X}^t = \mathrm{d}\mathbf{B}^t + \tfrac{1}{2}\nabla \log\{\pi(\mathbf{X})\}\,\mathrm{d}t, \tag{10}$$

where ∇ denotes the vector of partial derivatives with ith element $\partial/\partial X_i$ and $\mathrm{d}\mathbf{B}^t$ is an increment of k-dimensional Brownian motion; see Ikeda and Watanabe (1989), for example. Essentially, equation (10) says that, given $\mathbf{X}^t$, $\mathbf{X}^{t+\mathrm{d}t}$ follows a multivariate normal distribution with mean $\mathbf{X}_t + \frac{1}{2}\nabla \log\{\pi(\mathbf{X})\}\,\mathrm{d}t$ and variance matrix $\mathbf{I}_k\,\mathrm{d}t$, where $\mathbf{I}_k$ denotes the k-dimensional identity matrix.

It can be shown that this diffusion process will, under certain regularity conditions, have unique stationary distribution π; see Roberts and Tweedie (1995). The process $\mathbf{X}$ is called the *Langevin* diffusion for π and, in the context of MCMC sampling, is referred to as the unadjusted Langevin algorithm (ULA). Clearly, direct simulation of this process, in terms of infinitesimal time intervals, is impossible. Thus, we *discretize* the process and take time intervals of length $\delta > 0$. However, this discretization disturbs the convergence properties of the process, so it is necessary to 'correct' the algorithm by means of a Metropolis–Hastings rejection step, leading to the Metropolis-adjusted Langevin algorithm (MALA) as follows.

Given the current state of the process $\mathbf{X}^t$, we generate a candidate state $\mathbf{Y}$ via a multivariate normal distribution, with mean $\mathbf{X}^t + \delta\boldsymbol{\mu}(\mathbf{X}^t)$ and variance matrix $\delta\mathbf{I}_k$, where $\boldsymbol{\mu}(\mathbf{X})$ is some vector-valued function of $\mathbf{X}$. Then, we accept $\mathbf{Y}$ as the state of the process at time $t+\delta$ with probability

$$\min\left(1, \frac{\pi(\mathbf{Y})}{\pi(\mathbf{X})} \exp\left[-\tfrac{1}{2}(\boldsymbol{\mu}(\mathbf{Y}) + \boldsymbol{\mu}(\mathbf{X}^t))'(2(\mathbf{Y} - \mathbf{X}^t) + \delta\{\boldsymbol{\mu}(\mathbf{Y}) - \boldsymbol{\mu}(\mathbf{X}^t)\})\right]\right). \tag{11}$$

The MALA is somewhat similar to the random walk Metropolis algorithm but with a proposal distribution which takes account of the gradient of $\log\{\pi(\cdot)\}$, biasing the algorithm towards moves 'uphill', i.e. in the direction of the modes of π. The Metropolis–Hastings correction ensures that the process has the correct stationary density. However, the rate at which the process converges to the stationary distribution can be very slow; see Roberts and Tweedie (1995) and Roberts and Rosenthal (1996).

To improve the convergence properties of the MALA, a final modification is adopted, whereby the proposal mean is truncated to $\mathbf{X}^t + \min[D, \frac{1}{2}\nabla \log\{\pi(\mathbf{X})\}]$, for some constant $D > 0$, with a corresponding correction of the acceptance probability in expression (11). Thus, the Metropolis-adjusted truncated Langevin algorithm exhibits the desirable properties of both the random walk Metropolis algorithm and the Langevin proposal from the ULA. See Roberts and Tweedie (1995) and Neal (1993), for further discussion.

4.3. Dimension jumping algorithms

Occasionally, it is necessary to specify a model in such a way that the number of parameters in the model is, in itself, a parameter. Such problems often occur in the context of mixture modelling, where we may not know how many components are in the mixture, and so we require a Markov chain that can move between states of different dimensions, corresponding to the differing number of parameters associated with each plausible model.

Several methods have been proposed for tackling problems where it is necessary for our Markov chain to move between spaces of different dimension. Such problems commonly arise in mixture modelling (Richardson and Green, 1997), model choice (Green, 1995), image analysis, variable selection and changepoint analysis (Phillips and Smith, 1996; Grenander and Miller, 1994), for example.

MCMC methods for problems of this sort were first addressed by Grenander and Miller (1994) who proposed what were known as jump–diffusion algorithms. These were subsequently discussed by Phillips and Smith (1996) and work as follows. Let $(\mathbf{X}, p)^t$ denote a continuous time Markov process, where p dictates the dimension of $\mathbf{X}$, perhaps corresponding to the number of components in a mixture, with parameters denoted by $\mathbf{X}$, for example. This process is constructed to combine a jump component, which allows discrete transitions between models (altering p) at random times, and a diffusion component, which updates the model-specific parameters (for a particular fixed p), in between these jumps. Green (1995) proposed a more general framework, known as the reversible jump MCMC (RJMCMC) method, which encompasses this diffusion method, as well as many other common techniques.

Simply put, the RJMCMC algorithm extends the basic Metropolis–Hastings algorithm to general state spaces, so that π becomes a general measure, rather than a density, and the proposal density $q(\mathbf{x}, \mathbf{y})$ is replaced by a proposal kernel $q(\mathbf{x}, \mathrm{d}\mathbf{y})$. If we let $\mathbf{x}$ denote the state variable, which lies in some subspace $\mathcal{C}_1 \subseteq \mathcal{C}$ say, and let $\pi(\mathrm{d}\mathbf{x})$ denote the target probability measure (as opposed to target density), then we can consider a countable family of move types, which we label $m = 1, 2, \ldots$. When the current state is $\mathbf{x} \in \mathcal{C}_1 \subseteq \mathcal{C}$, both a move type m and candidate observation $\mathbf{y} \in \mathcal{C}_2 \subseteq \mathcal{C}$, say, are proposed, with joint distribution $q_m(\mathbf{x}, \mathrm{d}\mathbf{y})$. The move is then accepted with probability

$$\alpha_m(\mathbf{x}, \mathbf{y}) = \min\left\{1, \frac{\pi(\mathrm{d}\mathbf{y})\, q_m(\mathbf{y}, \mathrm{d}\mathbf{x})}{\pi(\mathrm{d}\mathbf{x})\, q_m(\mathbf{x}, \mathrm{d}\mathbf{y})}\right\}. \tag{12}$$

This probability is rigorously defined, subject to a 'dimension matching' condition on $q_m(\mathbf{x}, \mathrm{d}\mathbf{y})$ (Green, 1995) that effectively matches the degrees of freedom of joint variation of $\mathbf{x}$ and $\mathbf{y}$ as the dimension changes with the number of parameters under different models.

Green (1995) provided a 'template' for dimension-changing moves. Suppose that a move of type m is proposed, from $\mathbf{x}$ to a point $\mathbf{y}$ in a higher dimensional space. This will very often be implemented by drawing a vector of continuous random variables $\mathbf{u}$, independent of $\mathbf{x}$, and setting $\mathbf{y}$ to be a deterministic and invertible function $\mathbf{y}(\mathbf{x}, \mathbf{u})$. The reverse of the move (from $\mathbf{y}$ to $\mathbf{x}$) can be accomplished by using the inverse transformation, so that, in this direction, the proposal is deterministic, once m has been chosen. Thus, if $\pi(\mathrm{d}\mathbf{x})\, q_m(\mathbf{x}, \mathrm{d}\mathbf{y})$ has a finite density with respect to some symmetric measure on $\mathcal{C} \times \mathcal{C}$, then we can generate the pair $(m, \mathbf{y}) \sim f_m(\mathbf{x}, \mathbf{y})$, the joint density of m and $\mathbf{y}$, given the current state $\mathbf{x}$, in which case the acceptance probability in equation (12) becomes

$$\min\left\{1, \frac{p(\mathbf{y})\, r_m(\mathbf{y})}{p(\mathbf{x})\, r_m(\mathbf{x})\, q(\mathbf{u})} \left| \frac{\partial \mathbf{y}}{\partial(\mathbf{x}, \mathbf{u})} \right| \right\},$$

where ${\boldsymbol\jmath}r_m(\mathbf{x})$ is the probability of choosing move type m when in state $\mathbf{x}$, $q(\mathbf{u})$ is the density function of $\mathbf{u}$ and $p(\mathbf{x})$ denotes the posterior density of $\mathbf{x}$ under π (see Green (1995) for further details, together with technical details and conditions). The final term in the ratio above is a Jacobian arising from the change of variable from $(\mathbf{x}, \mathbf{u})$ to $\mathbf{y}$.

To provide a more intuitive description, let us consider a simple example provided by Green (1995). Suppose that we have just two subspaces C_1 and C_2 of dimension 1 and 2 respectively. Now assume that we are currently in state $(x_1, x_2) \in C_2$. A plausible dimension-changing move might be to propose $y = \frac{1}{2}(x_1 + x_2)$ as the new state. For detailed balance to hold, the reverse move must be fixed so that, given that the chain is in state x, the only plausible move is to (y_1, y_2), where $y_1 = x - u$ and $y_2 = x + u$, i.e. so that $\frac{1}{2}(y_1 + y_2) = x$. Note that this dimension matching strategy might also be thought of as a simple moment matching mechanism, which may easily be generalized to more complex problems, as we shall see in Section 5.3.

The RJMCMC method provides a general framework, encompassing many other well-known algorithms. For example, when we consider only subspaces of the same dimension, the RJMCMC algorithm reduces to the random scan Metropolis–Hastings algorithm. In the case of point processes, the RJMCMC method encompasses the method proposed by Geyer and Møller (1994). Similarly, and as we mentioned above, the jump–diffusion processes of Grenander and Miller (1994) and Phillips and Smith (1996) can also be thought of as special cases of the RJMCMC method, where the moves within subspaces are made by continuous time diffusion processes.

Having discussed the wide range of implementations of MCMC methodology, and some of the most important implementational issues regarding their practical application, in the next section we discuss how they may be applied to particular problems to illustrate how some of the practical issues arise and are dealt with in a few different contexts.

5. Applications

In this section, we examine some specific modelling problems and explain how MCMC methods can be used to fit models of each type. We begin with a simple model to highlight the general approach to MCMC-based model fitting, before discussing more complex problems and the various implementational and practical issues associated with MCMC methodology in general.

5.1. A simple changepoint model

To explain, in greater detail, exactly how to formalize statistical problems for use with MCMC algorithms, we follow Buck *et al.* (1993) who used a non-linear regression model with a single changepoint, to describe the shape of prehistoric corbelled tombs.

Data were collected concerning the shapes of corbelled tombs at sites throughout the Mediterranean. The data consist of a series of measurements of depth from the apex of these tombs, d, and the corresponding radius r; see Buck *et al.* (1993, 1996) and Fig. 2. Having obtained data from tombs at different sites, we try to model the data from each tomb separately, to ascertain whether or not different civilizations used similar technologies to construct the tombs, resulting in tombs of similar shapes and therefore models with similar fitted parameters.

We adopt the model

$$\log(r_i) = \begin{cases} \log(\alpha_1) + \beta_1 \log(d_i + \delta) + \epsilon_i & 0 \leqslant d_i < \gamma, \\ \log(\alpha_2) + \beta_2 \log(d_i + \delta) + \epsilon_i & \gamma \leqslant d_i \end{cases}$$

with the constraint that

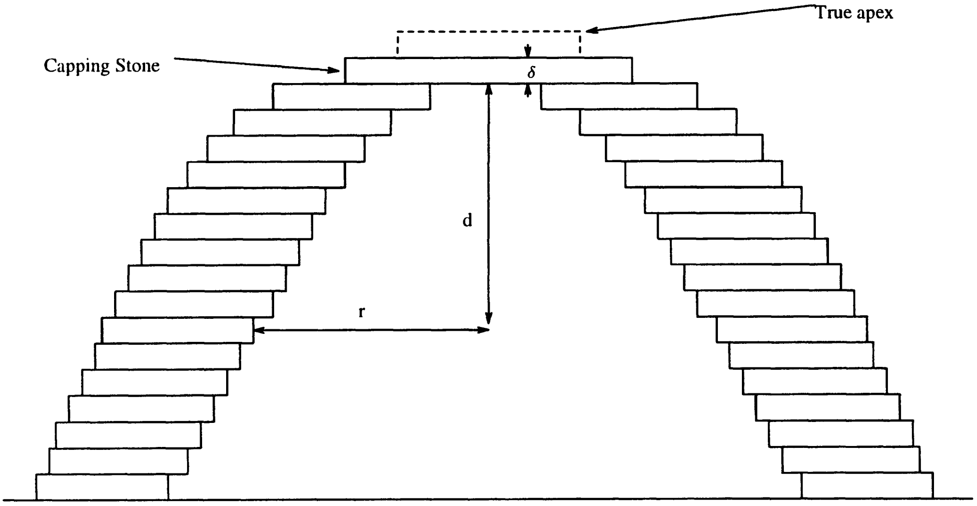

Fig. 2. Shape of the inside of an ideal corbelled tomb with a capping stone: - - - -, true apex stone

$$\beta_1 > \beta_2 > 0, \tag{13}$$

to ensure that the tomb wall is steeper above the changepoint (at depth γ) and, for continuity at $d = \gamma$, we set

$$\beta_2 = \beta_1 + \frac{\log(\alpha_1/\alpha_2)}{\log(\gamma + \delta)}.$$

Hinkley (1969, 1971) has described the maximum likelihood approach to this problem, whereas a Bayesian approach is described by Smith and Cook (1980). We assume that the errors ϵ_i are IID standard normal variates, and that the depths are ordered $d_1 < \ldots < d_n$, with $d_{i^*} < \gamma \leqslant d_{i^*+1}$ for some $1 \leqslant i^* < n$, where both γ and i^* are unknown. The likelihood is then given by

$$L = \frac{1}{\sigma^n} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{i^*} \{\ln(r_i) - \ln(\alpha_1) - \beta_1 \ln(d_i + \delta)\}^2\right]$$
$$\times \exp\left[-\frac{1}{2\sigma^2} \sum_{i=i^*+1}^{n} \left\{\ln(r_i) - \ln(\alpha_2) - \beta_1 \ln(d_i + \delta) - \frac{\ln(\alpha_1/\alpha_2)\ln(d_i + \delta)}{\ln(\gamma + \delta)}\right\}^2\right]. \tag{14}$$

Thus, we have a six-parameter model with parameters σ, γ, δ, β_1, α_1 and α_2. The likelihood surface is ill suited to analysis by conventional methods. For example, the continuous parameter γ defines the ranges of the summation of the exponent sums of squares so the likelihood surface may be only piecewise continuous with discontinuities at the data ordinates. However, in the context of the Gibbs sampler, for example, such problems disappear.

Given the likelihood function of equation (14) and, if we assume a uniform prior for σ^2, then the posterior will be proportional to this likelihood. If we let $\tau = \sigma^{-2}$, then we can see that the conditional posterior distribution is proportional to

$$\tau^{n/2} \exp(-A\tau/2)$$

where

$$A = \sum_{i=1}^{i^*} \{\ln(r_i) - \ln(\alpha_1) - \beta_1 \ln(d_i + \delta\}^2$$

$$+ \sum_{i=i^*+1}^{n} \left\{ \ln(r_i) - \ln(\alpha_2) - \beta_1 \ln(d_i + \delta) - \frac{\ln(\alpha_1/\alpha_2)\ln(d_i + \delta)}{\ln(\gamma + \delta)} \right\}^2.$$

Thus, the full conditional distribution of τ, given all the other parameters, is a gamma distribution, $\tau \sim \Gamma(n/2 + 1, A/2)$, under the assumption of a uniform prior.

Similar results can be obtained for three of the other parameters, α_1, α_2 and β_1. For example, consider the conditional posterior distribution for β_1. Ignoring all terms not involving β_1, the likelihood is proportional to

$$\exp\left[-\frac{1}{\sigma^2} \sum_{i=1}^{i^*} \beta_1 \ln(d_i + \delta)\{\ln(r_i) - \ln(\alpha_1) - \tfrac{1}{2}\beta_1 \ln(d_i + \delta)\} \right]$$

$$\times \exp\left[-\frac{1}{\sigma^2} \sum_{i=i^*+1}^{n} \beta_1 \ln(d_i + \delta)\left\{ \ln(r_i) - \ln(\alpha_2) - \tfrac{1}{2}\beta_1 \ln(d_i + \delta) - \frac{\ln(\alpha_1/\alpha_2)\ln(d_i + \delta)}{\ln(\gamma + \delta)} \right\} \right],$$

which can be manipulated to be of the form of a normal distribution with mean

$$\mu = \frac{\nu^2}{\sigma^2} \sum_{i=1}^{n} \ln(d_i + \delta)\ln(r_i) - \ln(\alpha_1) \sum_{i=1}^{i^*} \ln(d_i + \delta)$$

$$- \sum_{i=i^*+1}^{n} \left[\ln(\alpha_2)\ln(d_i + \delta)\left\{ 1 - \frac{\ln(d_i + \delta)}{\ln(\gamma + \delta)} \right\} + \frac{\ln(\alpha_1)\ln(d_i + \delta)^2}{\ln(\gamma + \delta)} \right]$$

and variance

$$\nu^2 = \sigma^2 \Big/ \sum_{i=1}^{n} \ln(d_i + \delta)^2.$$

Thus, with a uniform prior, the posterior conditional will be a truncated normal distribution, with parameters μ and ν^2. In a similar manner, the full conditional distributions for parameters α_1 and α_2 can be shown to be log-normal distributions.

The conditional posterior distributions for γ and δ are of non-standard form. The fact that the full conditionals for the majority of parameters are standard distributions suggests the use of the Gibbs sampler to sample from the posterior. Thus, we require some mechanism for sampling from the non-standard conditional distributions of δ and γ. This may be most easily performed via some form of univariate sampling method like those discussed in Section 2.1. For example, Buck *et al.* (1993) used the rejection sampling method to sample values for γ and δ. Alternatively, we could adopt a hybrid approach by introducing a Metropolis step into the sampling algorithm for these parameters, as described in Section 2.3. For example, we could choose a univariate normal proposal distribution for δ and γ, with appropriate mean and variance, and then use a single Metropolis–Hastings update step each time that either of the conditionals for these parameters would have been used in the Gibbs sampler algorithm.

One additional complication with this model is that we require that the parameters satisfy the order constraint (13). This difficulty may be overcome by sampling from the unconstrained conditionals, but rejecting sampled points which fail to satisfy this condition. In general, this may be a somewhat wasteful method and may lead to slow mixing, but for this example Buck *et al.* (1993) stated that this appears not to be the case. In the case of parameter β_1 for example, and given that its value lies in the range $[a, b]$, an alternative approach is to sample a uniform random variable u between $\Phi\{(a-u)/v\}$ and $\Phi\{(b-\mu)/v\}$ and set $\beta_1 = \Phi^{-1}(u)$, where Φ denotes the standard normal cumulative density function. This dispenses with the potentially wasteful rejection of observations from the unconstrained density, with little added computational expense. However, the cumulative density function and its inverse may not always be available for all densities from which we may need to sample and so the rejection method may often be required.

Having seen how posterior distributions may be both derived and sampled, we now discuss a second example, in which we briefly outline how a finite mixture distribution may be modelled via MCMC sampling, and focus on the interpretation of the resulting output in terms of what information it provides regarding the performance of the sampler.

5.2. Finite mixture distributions with a fixed number of components

The Bayesian implementation of finite mixture distributions has been an area of considerable interest within the literature; see Titterington *et al.* (1985), McLachlan and Basford (1988), Diebolt and Robert (1994), Nobile (1994), Robert (1995), Mengersen and Robert (1996) and Feng and McCulloch (1996), for example.

A k-component mixture model has density of the form

$$f(\mathbf{x}) = \sum_{j=1}^{k} w_j \pi_j(\mathbf{x}),$$

where the π_j, $j = 1, \ldots, k$, are densities which are known at least up to some multiplicative constant, and the proportions w_j, $j = 1, \ldots, k$, are such that $\Sigma_{j=1}^{k} w_j = 1$.

For each density π_j, we have a set of parameters $\boldsymbol{\theta}_j$ (which includes w_j) and, given data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we introduce hyperparameters denoted by z_i, $i = 1, \ldots, n$, where the $z_i \in \{1, \ldots, k\}$ are indicator variables assigning each observation to a particular component. Thus, we obtain a hierarchical model with conditional distributions of the form

$$p(z_i = j) = w_j,$$

$$\mathbf{x}_i | z_i \sim \pi(\mathbf{x}|\boldsymbol{\theta}_{z_i}).$$

For example, suppose that π_j, $j = 1, \ldots, k$, denotes a normal distribution with mean μ_j and variance σ_j^2. Then, Diebolt and Robert (1994) suggested conjugate priors for the parameters of the form

$$\mathbf{W} \sim D(\alpha_1, \ldots, \alpha_k),$$

$$\mu_j | \sigma_j \sim N(\nu_j, \sigma_j^2/n_j),$$

$$\sigma_j^{-2} \sim \Gamma(\phi_j, s_j^2),$$

where $D(\cdot)$ denotes the Dirichlet distribution and n_j denotes the number of observations assigned to component j. Note that the prior specification for the σ_j is often (and equivalently) expressed as an inverse Wishart distribution (Robert, 1995), but we retain the inverse gamma distribution here, for simplicity. We may represent the model graphically, as in Fig. 3 which provides the DAG and conditional independence graph for the model. From the conditional independence graph we can see that the posterior distribution simplifies to give the Bayesian hierarchical model

$$p(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{z}, \mathbf{W}) = p(\mathbf{W}|\boldsymbol{\alpha})\, p(\mathbf{z}|\mathbf{W})\, p(\boldsymbol{\sigma}|\boldsymbol{\phi}, \mathbf{s})\, p(\boldsymbol{\mu}|\boldsymbol{\sigma}, \boldsymbol{\nu})\, p(\mathbf{x}|\boldsymbol{\sigma}, \boldsymbol{\mu}, \mathbf{z}).$$

Thus, it is a fairly straightforward matter to fit the observed data via the Gibbs sampler.

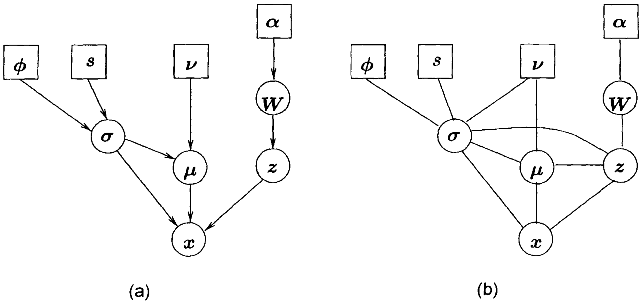

Fig. 3. Graphical representations of a mixture model: (a) DAG; (b) conditional independence graph (the square nodes represent variables which are known, such as observed data, or (hyper)prior parameters that are assumed known and fixed)

Bowmaker *et al.* (1985) analysed data on the peak sensitivity wavelengths for individual microspectrophotometric records on a set of monkeys' eyes. A subsequent analysis of the data for one particular monkey by Spiegelhalter *et al.* (1996c) made the assumption that each observation comes from one of two populations with common variance. We shall repeat the analysis of their data, imposing an order constraint on the two group means, for identifiability. (This point is further discussed in Section 5.3.)

Taking vague priors for all four parameters (μ_1, μ_2, σ and w), we obtain the output provided in Fig. 4, by running the Gibbs sampler for 5000 iterations. The trace plots for all four parameters indicate a high dependence between successive iterations, which suggests a slow mixing or convergence rate. Plots of this type, where sustained jumps are observed in the raw output, often suggest that the sampler may be improved by reparameterizing, for example, and, in this case, the behaviour is well understood. Both Spiegelhalter *et al.* (1996c) and Robert (1995) highlighted the

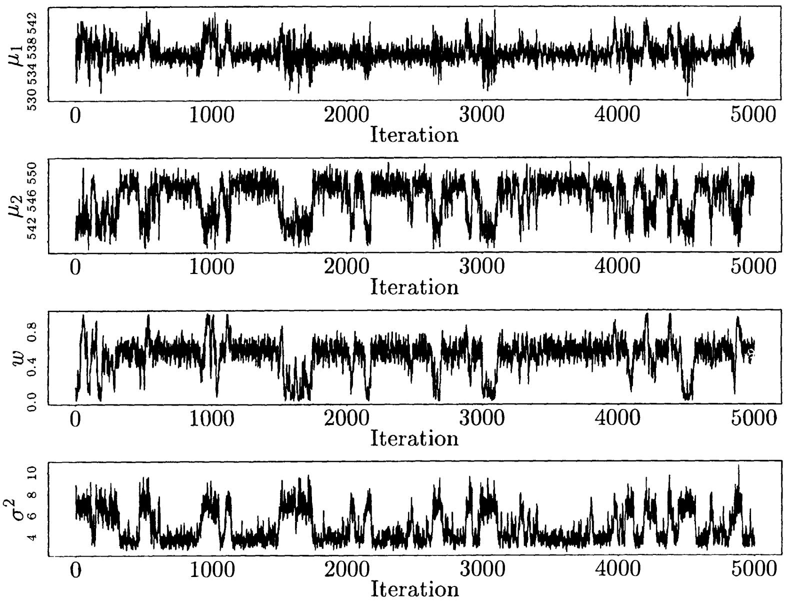

Fig. 4. Raw output from the original mixtures model

fact that there is the possibility that at some point during the simulation a large proportion of (or even all) observations may be assigned to a single component, which creates a *trapping state* (or almost reducibility) of the chain. Essentially, when only a few observations are assigned to a particular component, the fit of this component to those observations is particularly good, creating a local mode in the posterior, from which it is difficult for the Gibbs sampler to escape.

This problem may be overcome by reparameterizing. Here, we set

$$f(x) = w\,N(\mu, \sigma^2) + (1 - w)\,N(\mu + \theta, \sigma^2),$$

with the restriction that $\theta > 0$, to ensure identifiability. This reparameterization creates some dependence between the two components and reduces the influence of this local mode, so that the sampler may move more freely through these formerly trapping states. This reparameterization was suggested by Robert (1995), who provides further discussion of this problem.

Fig. 5 shows the corresponding raw trace plots for the reparameterized model. This plot shows little of the dependence between successive iterations that was present in Fig. 4, suggesting that the Gibbs sampler's performance is improved by reparameterizing. None of the trace plots of Fig. 5 appear to have settled to a stable value. This suggests that convergence may not yet have been achieved and is confirmed by using some of the convergence diagnostic techniques discussed in Section 3.1. For example, having run three independent replications of the chain, the method of Heidelberger and Welch (1983) indicates problems with the convergence of all four parameters. Similarly, the method of Raftery and Lewis (1992) suggests problems with the convergence of parameters μ_1, μ_2 and σ. Finally, the original method of Gelman and Rubin (1992) provides no evidence of a lack of convergence, but a plot of the numerator and denominator of their $\hat{R}$-value (as suggested by Brooks and Gelman (1997)) clearly indicates that none of the parameters has converged even after 10 000 iterations. In this case, it does not seem sensible to continue to run the chain (see Brooks and Roberts (1996)); rather it might be more sensible to take another look at

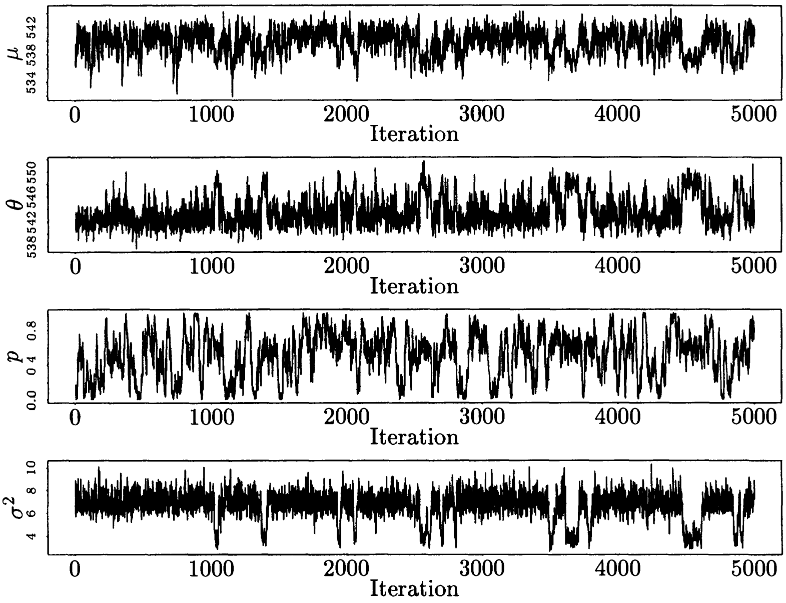

Fig. 5. Raw output from the reparameterized mixtures model

either the model or the model fitting process to examine what might be causing these convergence problems. For example, convergence may be improved if empty components are disallowed, so that both components always have at least one element.

We next consider another, related, example in which we allow the number of components, k, to vary as an additional parameter in the model. This allows us to demonstrate how the RJMCMC algorithm may be implemented for problems of this sort and to discuss some of the practical issues associated with RJMCMC algorithms.

5.3. Finite mixture distributions with an unknown number of components

An interesting extension to the normal mixtures example can be obtained if we also allow the number of components to vary. This case is discussed by Richardson and Green (1997), who used an RJMCMC algorithm to fit the normal mixture model of Section 5.2 with a variable number of components. Here, we assume the following priors: $p(\sigma_j^{-2}) \sim \Gamma(\phi, s^2)$, $p(\mu_j) \sim N(\nu_j, \kappa_j^2)$, $p(k) \sim \mathrm{Po}(\lambda)$ and $p(\mathbf{p}|k) \sim D(\alpha, \ldots, \alpha)$.

The RJMCMC implementation follows the Gibbs sampler above for the first few steps at each iteration but also introduces a new step allowing for new components to be formed or old components to be deleted. Fig. 6 provides the corresponding graphical representation of the new model, from which the conditional independence graph and hence the necessary conditional distributions can be derived. The RJMCMC algorithm can be implemented via the following steps at each iteration:

step 1—update the p_j, $j = 1, \ldots, k$;
step 2—update the μ_j, $j = 1, \ldots, k$;
step 3—update the σ_j^2, $j = 1, \ldots, k$;
step 4—update the z_i, $i = 1, \ldots, n$;
step 5—split one component in two, or combine two into one, i.e. update k.

Steps 1–4 can be performed as in the previous example, but the split or combine move involves altering the value of k and requires reversible jump methodology, since it involves altering the

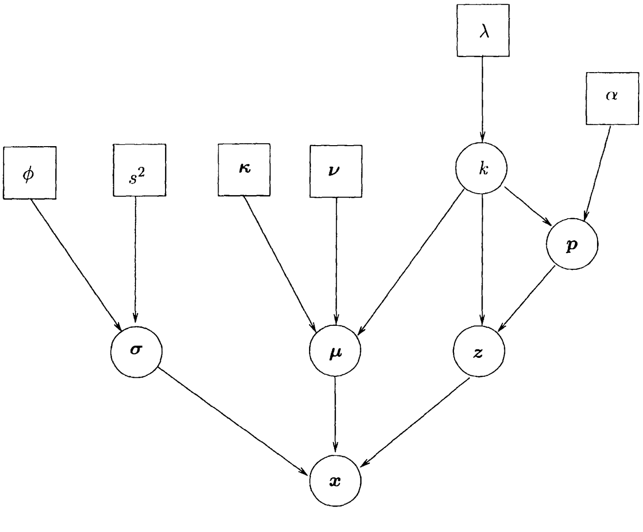

Fig. 6. Graphical representation of a mixture model with an unknown number of components

number of parameters in the model. As discussed in Section 4.3, we need to specify proposal distributions for these moves in such a way as to observe the dimension matching requirement of the algorithm. We can do this as follows.

We begin by randomly selecting either a split or a combination move, with probabilities q_k and $1 - q_k$ respectively. If we choose to combine two adjacent components, j_1 and j_2 say, to form component j^*, then we reduce k by 1, set $z_i = j^*$ for all $i \in \{i\colon z_i = j_1 \text{ or } z_i = j_2\}$ and pick values for p_{j^*}, μ_{j^*} and $\sigma^2_{j^*}$. New values are selected for the two components' parameters, by matching the zeroth, first and second moments of the new component with those of the two components that it replaces, i.e.

$$\begin{aligned} p_{j^*} &= p_{j_1} + p_{j_2}, \\ p_{j^*}\mu_{j^*} &= p_{j_1}\mu_{j_1} + p_{j_2}\mu_{j_2}, \\ p_{j^*}(\mu^2_{j^*} + \sigma^2_{j^*}) &= p_{j_1}(\mu^2_{j_1} + \sigma^2_{j_1}) + p_{j_1}(\mu^2_{j_2} + \sigma^2_{j_2}). \end{aligned}$$

Thus, the choices are deterministic, once j_1 and j_2 have been selected.

So that the dimension matching requirement can be observed, the splitting move is essentially defined to be the reverse of the combination move described above. Richardson and Green (1997) proposed the following method. Randomly select a component j^*, which we shall split to form two new components j_1 and j_2. This is done by reversing the moment matching algorithm above, and selecting three random variables, u_1, u_2, $u_3 \in (0, 1)$ (perhaps uniformly or with beta distributions, as suggested by Richardson and Green (1997)). We then set the p- and μ-values for the new components to be

$$\begin{aligned} p_{j_1} &= p_{j^*}u_1 \quad \text{and} \quad p_{j_2} = p_{j^*}(1 - u_1), \\ \mu_{j_1} &= \mu_{j^*} - u_2\sigma\sqrt{(p_{j_2}/p_{j_1})} \quad \text{and} \quad \mu_{j_2} = \mu_{j^*} - u_2\sigma\sqrt{(p_{j_1}/p_{j_2})}, \\ \sigma^2_{j_1} &= u_3(1 - u_2)\sigma^2_{j^*}p_{j^*}/p_{j_1} \quad \text{and} \quad \sigma^2_{j_2} = (1 - u_3)(1 - u_2)\sigma^2_{j^*}p_{j^*}/p_{j_2}. \end{aligned}$$

Then, we check that two new components are adjacent, i.e. that there is no μ_i such that $\min(\mu_{j_1}, \mu_{j_2}) \leqslant \mu_i \leqslant \max(\mu_{j_1}, \mu_{j_2})$. If this condition is not satisfied, we automatically reject this move, since the split–combine pair could not possibly be reversible. If the proposed move is not rejected, then we use a step of the Gibbs sampler to reassign all those observations that we previously assigned to component j^*, setting $z_i = j_1$ or $z_i = j_2$, for all such observations.

Having proposed a particular split or combination, we then decide whether or not to accept the move proposed. Richardson and Green (1997) provided the acceptance probabilities that are associated with proposals of each type, but they can be quite difficult to derive, involving various density ratios, a likelihood and a Jacobian term. However, given the acceptance probabilities for univariate normal mixtures, the corresponding acceptance probabilities for similar mixture models (gamma mixtures, for example) can be obtained by substituting the appropriate density function in the corresponding expressions. For more difficult problems, symbolic computation tools are available which can simplify the problem and, in addition, Green (1995) has provided acceptance probabilities which can be used for several quite general problems, thereby avoiding the problem altogether in these cases.

Richardson and Green (1997) provided an analysis of three data sets via the method above. We shall not provide an additional example here but instead briefly discuss a few of the practical issues associated with this methodology.

In the fixed k example, we imposed an order constraint on the component means for identifiability, i.e. so that component 1 was constrained to have the lower mean. A similar problem occurs in this case, which we call 'label switching'. If we take a two-component mixture, with means μ_1

and μ_2, it is quite possible that the posterior distributions of these parameters may overlap, and this creates a problem in analysing the output. If the means are well separated, then labelling posterior realizations by ordering their means should result in an identical labelling with the population. However, as the separation between the means decreases, label switching occurs, where observations are assigned the incorrect label, i.e. according to our labelling criterion they belong to one component, when in fact they belong to another. One way around this problem is to order not on the means but on the variances (in the heterogeneous case) or on the weights. In general the labelling mechanism can be performed after the simulation is complete and should generally be guided by the context of the problem in hand. Of course, this is not a problem if, in terms of posterior inference, we are interested only in parameters which are invariant to this labelling, such as the number of components in the mixture.

On a more practical note, if we run the RJMCMC algorithm with the five steps outlined above, this results in a slow mixing chain, for exactly the same reasons as with the fixed k example. Occasionally, empty components will arise and the sampler will retain these for extended periods of time, preventing the sampler from mixing efficiently. To circumvent this problem, Richardson and Green (1997) introduced an additional step whereby empty components can be deleted from the model. Such moves must be reversible, so that empty components may also be created, but the introduction of this additional step greatly improves the mixing dynamics of the sampler.

This leads to an allied point of how we might formally determine how well the sampler is mixing and, indeed, how long the sampler should be run before it achieves stationarity. Traditional convergence assessment techniques are difficult to apply in this situation, since both the number of parameters and their interpretations are changing with time. One parameter that is not affected by jumps between different models is the k-parameter. To assess the convergence of this parameter alone is actually a very simple matter, since there are plenty of diagnostic methods for univariate parameters of this sort. Using such methods, one might first assess convergence of k and then, once k appears to have reached stationarity, look at convergence of the parameters for fixed k-values. However, it is likely that, even in a long run, some values of k will not be visited very often and thus the assessment of convergence within such k is almost impossible. Thus, the best approach to the assessment of convergence of such samplers is unclear; see Brooks (1997).

6. Discussion

It is clear that existing MCMC methods provide a powerful statistical tool and have revolutionized practical Bayesian statistics over the past few years. However, the implementation of many of these methods requires some expertise, and it is hoped that this paper, and associated references, cover many of the issues that potential practitioners should be aware of.

No review would be complete without some discussion of what the future holds for MCMC methods. Though considerable work is still being performed on implementational issues for example, perhaps the most exciting recent development is in the area of exact simulation.

One of the biggest drawbacks associated with MCMC methods is the problem of ascertaining the proximity of the distribution of any given Markov chain output to the target distribution. We have discussed how convergence diagnostic methods can be used to provide some reassurance that inference based on any given sample is reliable, but often different methods give contradictory or unclear conclusions. Thus, there is great interest in the construction of samplers for which this problem does not arise. Such samplers are known as *exact samplers* and some suggestions for such methods have recently been developed.

The original method was proposed by Propp and Wilson (1996) who used the idea of Markov chain *coupling* to generate a single observation from the target distribution. The method assumes that we have a finite state space, with n distinct states and transition probabilityp_{ij} of going from state i to state j, and that the stationary distribution of a Markov chain with these transition probabilities is π. Each transition from state $\mathbf{X}^t$ to $\mathbf{X}^{t+1}$ is assumed to be based on a single random vector $\mathbf{u}^t$, uniquely defining a random map from the state space to itself, so that the next state of the chain can be written as a deterministic function of the present state and this random observation, i.e.

$$\mathbf{X}^{t+1} = \phi(\mathbf{X}^t, u^t),$$

for some function ϕ. The method proceeds as follows.

We begin by selecting a sequence of random 'seeds' $\{u^t\colon t = 0, -1, -2, \ldots\}$ and running a set of n chains (one started from each of the n distinct states) for N iterations, beginning at time $-N$ and ending at time 0. For each point in the state space, we run a single Markov chain using that point as a starting value at time $-N$ and examine the states of these chains at time 0. At some point during the simulation, some of the sample paths may have met and coalesced. If all the chains have coalesced, then the set of final states at time 0 will consist of only a single point i^*, say. In this case, the final state of each of the chains at time 0 will have no dependence on their starting points. Thus, it follows that the distribution of the state i^* will be the target distribution, and we will have obtained a single observation from π. If all chains have not coalesced by time 0, then we restart the chains at time $-2N$ and run them again until time 0, retaining the same u^t-sequence. We continue to double the length of the simulation, until all chains have coalesced by time 0.

The advantage of this approach is that, if all the chains coalesce, then the distribution of their common state at time 0 (and therefore all subsequent states) will be *exactly* the target distribution. Thus, the need for convergence diagnostics and the like is removed. An obvious disadvantage of this approach is the computational burden associated with running n parallel chains, which may be prohibitively high in many practical applications. This problem may be overcome in the case where the state space has a partial ordering. In this case, the state space will have two (or more) extreme states and, if the simulation itself is monotonic, we can run chains from only these states in the knowledge that, when these chains converge, so also will all chains started at intermediate states. Thus, we greatly reduce the number of chains that we need to simulate. Several examples of this are presented in the literature; see Kendall (1996), Haggstrom *et al.* (1996) and Møller (1997).

The implementation of Propp and Wilson's (1996) exact simulation idea has subsequently been extended to non-finite state spaces and several researchers have provided further studies of these methods; see Haggstrom *et al.* (1996), Cai (1997), Foss and Tweedie (1997) and Murdoch and Green (1997). Though current exact simulation methods tend to be limited in their applicability, the idea of exact sampling promises to become an active area of future research, and it is clear that practical simulation procedures of this sort will eventually become available.

Acknowledgements

The author would like to thank various referees, anonymous and otherwise, who have offered comments and criticism of earlier drafts of this paper. Particular thanks go to Peter Green whose constructive criticisms have greatly improved the paper.

References

Almond, R. (1995) *Graphical Belief Modelling*. London: Chapman and Hall.

Amit, Y. (1991) On rates of convergence of stochastic relaxation for Gaussian and non-Gaussian distributions. *J. Multiv. Anal.*, **38**, 82–99.

Amit, Y. and Grenander, U. (1991) Comparing sweep strategies for stochastic relaxation. *J. Multiv. Anal.*, **37**, 197–222.

Applegate, D., Kannan, R. and Polson, N. G. (1990) Random polynomial time algorithms for sampling from joint distributions. *Technical Report 500*. Carnegie Mellon University, Pittsburgh.

Asmussen, S., Glynn, P. W. and Thorisson, H. (1992) Stationarity detection in the initial transient problem. *ACM Trans. Modllng Comput. Simuln*, **2**, 130–157.

Berzuini, C., Best, N. G., Gilks, W. R. and Larizza, C. (1997) Dynamic graphical models and Markov chain Monte Carlo methods. *J. Am. Statist. Ass.*, to be published.

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

Besag, J. E. and York, J. C. (1989) Bayesian restoration of images. In *Analysis of Statistical Information* (ed. T. Matsunawa), pp. 491–507. Tokyo: Institute of Statistical Mathematics.

Bowmaker, J. K., Jacobs, G. H., Spiegelhalter, D. J. and Mollon, J. D. (1985) Two types of trichromatic squirrel monkey share a pigment in the red-green region. *Vis. Res.*, **25**, 1937–1946.

Brooks, S. P. (1996) Quantitative convergence diagnosis for MCMC via CUSUMS. *Technical Report*. University of Bristol, Bristol.

———(1997) Discussion on On Bayesian analysis of mixtures with an unknown number of components (by S. Richardson and P. J. Green). *J. R. Statist. Soc.* B, **59**, 774–775.

———(1998) MCMC convergence diagnosis via multivariate bounds on log-concave densities. *Ann. Statist.*, to be published.

Brooks, S. P., Dellaportas, P. and Roberts, G. O. (1997) A total variation method for diagnosing convergence of MCMC algorithms. *J. Comput. Graph. Statist.*, **6**, 251–265.

Brooks, S. P. and Gelman, A. (1997) Alternative methods for monitoring convergence of iterative simulations. *J. Comput. Graph. Statist.*, to be published.

Brooks, S. P. and Morgan, B. J. T. (1994) Automatic starting point selection for function optimisation. *Statist. Comput.*, **4**, 173–177.

Brooks, S. P. and Roberts, G. O. (1996) Discussion on Convergence of Markov chain Monte Carlo algorithms (by N. G. Polson). In *Bayesian Statistics 5* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith). Oxford: Oxford University Press.

Buck, C. E., Cavanagh, W. G. and Litton, C. D. (1996) *Bayesian Approach to Interpreting Archaeological Data*. Chichester: Wiley.

Buck, C. E., Litton, C. D. and Stephens, D. A. (1993) Detecting a change in the shape of a prehistoric corbelled tomb. *Statistician*, **42**, 483–490.

Cai, H. (1997) A note on an exact sampling algorithm and Metropolis-Hastings Markov chains. *Technical Report*. University of Missouri, St Louis.

Chan, K. S. (1993) Asymptotic behaviour of the Gibbs sampler. *J. Am. Statist. Ass.*, **88**, 320–328.

Chib, S. and Greenberg, E. (1995) Understanding the Metropolis-Hastings algorithm. *Am. Statistn*, **49**, 327–335.

Cowles, M. K. and Carlin, B. P. (1996) Markov chain Monte Carlo convergence diagnostics: a comparative review. *J. Am. Statist. Ass.*, **91**, 883–904.

Cowles, M. K. and Rosenthal, J. S. (1996) A simulation approach to convergence rates for Markov chain Monte Carlo. *Technical Report*. Harvard School of Public Health, Boston.

Damien, P., Wakefield, J. and Walker, S. (1997) Gibbs sampling for Bayesian nonconjugate and hierarchical models using auxiliary variables. *Technical Report*. University of Michigan Business School, Ann Arbor.

Diaconis, P. and Stroock, D. (1991) Geometric bounds for eigenvalues of Markov chains. *Ann. Appl. Probab.*, **1**, 36–61.

Diebolt, J. and Robert, C. P. (1994) Estimation of finite mixture distributions through Bayesian sampling. *J. R. Statist. Soc.* B, **56**, 363–375.

Edwards, D. (1995) *Introduction to Graphical Modelling*. New York: Springer.

Edwards, R. G. and Sokal, A. D. (1988) Generalisation of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm. *Phys. Rev. Lett.*, **38**, 2009–2012.

Feng, Z. D. and McCulloch, C. E. (1996) Using bootstrap likelihood ratios in finite mixture models. *J. R. Statist. Soc.* B, **58**, 609–617.

Foss, S. G. and Tweedie, R. L. (1997) Perfect simulation and backward coupling. *Technical Report*. Institute of Mathematics, Novosibirsk.

Gafarian, A. V., Ancker, C. J. and Morisaku, T. (1978) Evaluation of commonly used rules for detecting "steady state" in computer simulation. *Nav. Res. Logist. Q.*, **25**, 511–529.

Garren, S. and Smith, R. L. (1995) Estimating the second largest eigenvalue of a Markov transition matrix. *Technical Report*. University of Cambridge, Cambridge.

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.

Gelfand, A. E., Sahu, S. K. and Carlin, B. P. (1995) Efficient parameterizations for normal linear mixed models. *Biometrika*, **82**, 479–488.

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

Gelman, A. and Rubin, D. (1992) Inference from iterative simulation using multiple sequences. *Statist. Sci.*, **7**, 457–511.

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattn Anal. Mach. Intell.*, **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, A. F. M. Smith, A. P. Dawid and J. O. Berger), pp. 169–193. Oxford: Oxford University Press.

Geyer, C. J. (1990) Reweighting Monte Carlo mixtures. *Technical Report*. University of Minnesota, Minneapolis.

———(1991) Markov chain Monte Carlo likelihood. In *Computing Science and Statistics: Proc. 23rd Symp. Interface* (ed. E. M. Keramidas), pp. 156–163. Fairfax Station: Interface Foundation.

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

Geyer, C. J. and Møller, J. (1994) Simulation procedures and likelihood inference for spatial point processes. *Scand. J. Statist.*, **21**, 359–373.

Geyer, C. J. and Thompson, E. A. (1995) Annealing Markov chain Monte Carlo with applications to ancestral inference. *J. Am. Statist. Ass.*, **90**, 909–920.

Gilks, W. R. and Roberts, G. O. (1996) Improving MCMC mixing. In *Markov Chain Monte Carlo in Practice* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter). London: Chapman and Hall.

Gilks, W. R., Thomas, D. and Spiegelhalter, D. J. (1992) Software for the Gibbs sampler. *Comput. Sci. Statist.*, **24**, 439–448.

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

Green, P. J. (1995) Reversible jump MCMC computation and Bayesian model determination. *Biometrika*, **82**, 711–732.

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

Haggstrom, O., van Lieshout, M. N. M. and Møller, J. (1996) Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. *Technical Report*. Department of Mathematics, Aalborg University, Aalborg.

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

Heidelberger, P. and Welch, P. D. (1983) Simulation run length control in the presence of an initial transient. *Ops Res.*, **31**, 1109–1144.

Higdon, D. (1996) Auxiliary variable methods for Markov chain Monte Carlo with applications. *Technical Report*. Institute of Statistics and Decision Sciences, Duke University, Durham.

Hills, S. E. and Smith, A. F. M. (1992) Parameterization issues in Bayesian inference. In *Bayesian Statistics 4* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith). Oxford: Oxford University Press.

Hinkley, D. V. (1969) Inference about the intersection of two-phase regression. *Biometrika*, **56**, 495–504.

———(1971) Inference in two-phase regression. *J. Am. Statist. Ass.*, **66**, 736–743.

Ikeda, N. and Watanabe, S. (1989) *Stochastic Differential Equations and Diffusion Processes*. Amsterdam: Elsevier.

Kass, R. E., Carlin, B. P., Gelman, A. and Neal, R. M. (1997) MCMC in practice: a roundtable discussion. *Am. Statistn*, to be published.

Kendall, W. S. (1996) Perfect simulation for the area-interaction point process. *Technical Report*. University of Warwick, Coventry.

Kimbler, D. L. and Knight, B. D. (1987) A survey of current methods for the elimination of initialization bias in digital simulation. In *Proc. 20th A. Simulation Symp.*, pp. 133–152.

Lauritzen, S. L. (1996) *Graphical Models*. Oxford: Oxford University Press.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988) Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J. R. Statist. Soc.* B, **50**, 157–224.

Lawler, G. F. and Sokal, A. D. (1988) Bounds on the L^2 spectrum for Markov chains and their applications. *Trans. Am. Math. Soc.*, **309**, 557–580.

Liu, C., Liu, J. and Rubin, D. B. (1993) A control variable for assessment of the convergence of the Gibbs sampler. *Proc. Statist. Comput. Sect. Am. Statist. Ass.*, 74–78.

Liu, J. S. (1994) Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Technical Report*. Harvard University, Cambridge.

Madigan, D. and York, J. (1995) Bayesian graphical models for discrete data. *Int. Statist. Rev.*, **63**, 215–232.

Marinari, E. and Parisi, G. (1992) Simulated tempering: a new Monte Carlo scheme. *Europhys. Lett.*, **19**, 451–458.

McLachlan, G. J. and Basford, K. E. (1988) *Mixture Models: Inference and Applications to Clustering*. New York: Dekker.

Mengersen, K. and Robert, C. (1996) Testing for mixtures: a Bayesian entropy approach. In *Bayesian Statistics 5* (eds J. M. Bernardo, A. F. M. Smith, A. P. Dawid and J. O. Berger). Oxford: Oxford University Press.

Mengersen, K. L. and Tweedie, R. L. (1995) Rates of convergence of the Hastings and Metropolis algorithms. *Ann. Statist.*, **24**, 101–121.

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

Meyn, S. P. and Tweedie, R. L. (1993) *Markov Chains and Stochastic Stability*. New York: Springer.

———(1994) Computable bounds for geometric convergence rates of Markov chains. *Ann. Appl. Probab.*, **4**, 981–1011.

Mira, A. and Tierney, L. (1997) On the issue of auxiliary variables in Markov chain Monte Carlo sampling. *Technical*

Report. University of Minnesota, Minneapolis.

Møller, J. (1997) Markov chain Monte Carlo and spatial point processes. In *Stochastic Geometry, Likelihood and Computation* (eds W. S. Kendall and M. N. M. van Lieshout). New York: Chapman and Hall.

Murdoch, D. J. and Green, P. J. (1997) Exact sampling from a continuous state space. *Technical Report*. Queen's University, Kingston.

Mykland, P., Tierney, L. and Yu, B. (1995) Regeneration in Markov chain samplers. *J. Am. Statist. Ass.*, **90**, 233–241.

Neal, R. M. (1993) Probabilistic inference using Markov chain Monte Carlo methods. *Technical Report CRG-TR-93-1*. Department of Computer Science, University of Toronto, Toronto.

Nobile, A. (1994) Bayesian analysis of finite mixture distributions. *PhD Thesis*. Carnegie Mellon University, Pittsburgh.

Peskun, P. H. (1973) Optimum Monte Carlo sampling using Markov chains. *Biometrika*, **60**, 607–612.

Phillips, D. B. and Smith, A. F. M. (1996) Bayesian model comparison via jump diffusions. In *Practical Markov Chain Monte Carlo* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter). London: Chapman and Hall.

Polson, N. G. (1996) Convergence of Markov chain Monte Carlo algorithms. In *Bayesian Statistics 5* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith). Oxford: Oxford University Press.

Propp, J. G. and Wilson, D. B. (1996) Exact sampling with coupled Markov chains and applications to statistical mechanics. *Rand. Struct. Alg.*, **9**, 223–252.

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

Richardson, S. and Green, P. J. (1997) On Bayesian analysis of mixtures with an unknown number of components (with discussion). *J. R. Statist. Soc.* B, **59**, 731–792.

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.

Robert, C. (1995) Mixtures of distributions: inference and estimation. In *Practical Markov Chain Monte Carlo* (eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter). London: Chapman and Hall.

———(1996) Convergence assessments for Markov chain Monte Carlo methods. *Statist. Sci.*, **10**, 231–253.

Robert, C. P. and Mengersen, K. L. (1995) Reparameterisation issues in mixture modelling and their bearing on the Gibbs sampler. *Technical Report*. Laboratoire de Statistique, Paris.

Roberts, G. O. (1994) Methods for estimating L^2 convergence of Markov chain Monte Carlo. In *Bayesian Statistics and Econometrics: Essays in Honor of Arnold Zellner* (eds D. Berry, K. Chaloner and J. Geweke). Amsterdam: North-Holland.

Roberts, G. O. and Polson, N. G. (1994) On the geometric convergence of the Gibbs sampler. *J. R. Statist. Soc.* B, **56**, 377–384.

Roberts, G. O. and Rosenthal, J. S. (1996) Quantitative bounds for convergence rates of continuous time Markov chains. *Technical Report*. University of Cambridge, Cambridge.

———(1997) Convergence of slice sampler Markov chains. *Technical Report*. University of Cambridge, Cambridge.

Roberts, G. O. and Sahu, S. K. (1997) Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *J. R. Statist. Soc.* B, **59**, 291–317.

Roberts, G. O. and Smith, A. F. M. (1994) Simple conditions for the convergence of the Gibbs sampler and Metropolis Hastings algorithms. *Stoch. Process. Applic.*, **49**, 207–216.

Roberts, G. O. and Tweedie, R. L. (1995) Exponential convergence of Langevin diffusions and their discrete approximations. *Technical Report*. University of Cambridge, Cambridge.

———(1996) Geometric convergence and central limit theorems, for multidimensional Hastings and Metropolis algorithms. *Biometrika*, **83**, 95–110.

Rosenthal, J. S. (1995a) Minorization conditions and convergence rates for Markov chain Monte Carlo. *J. Am. Statist. Ass.*, **90**, 558–566.

———(1995b) Rates of convergence for Gibbs sampling for variance component models. *Ann. Statist.*, **23**, 740–761.

———(1995c) Rates of convergence for data augmentation on finite sample spaces. *Ann. Appl. Probab.*, **3**, 319–339.

Schervish, M. J. and Carlin, B. P. (1992) On the convergence of successive substitution sampling. *J. Comput. Graph. Statist.*, **1**, 111–127.

Sinclair, A. J. and Jerrum, M. R. (1988) Conductance and the rapid mixing property for Markov chains: the approximation of the permanent resolved. In *Proc. 20th A. Symp. Theory of Computing*.

Smith, A. F. M. (1991) Bayesian computational methods. *Phil. Trans. R. Soc. Lond.* A, **337**, 369–386.

Smith, A. F. M. and Cook, D. G. (1980) Straight lines with a change-point: a Bayesian analysis of some renal transplant data. *Appl. Statist.*, **29**, 180–189.

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**, 3–23.

Smith, R. L. and Tierney, L. (1996) Exact transition probabilities for the independence Metropolis sampler. *Technical Report*. University of Cambridge, Cambridge.

Spiegelhalter, D. J., Thomas, A. and Best, N. G. (1996a) Computation on Bayesian graphical models. In *Bayesian Statistics 5* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith). Oxford: Oxford University Press.

Spiegelhalter, D. J., Thomas, A., Best, N. G. and Gilks, W. R. (1996b) *BUGS: Bayesian Inference using Gibbs Sampling, Version 0.50*. Cambridge: Medical Research Council Biostatistics Unit.

———(1996c) *BUGS Examples*, vol. 2. Cambridge: Medical Research Council Biostatistics Unit.

Swendsen, R. H. and Wang, J. S. (1987) Non-universal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.*, **58**, 86–88.

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

Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985) *Statistical Analysis of Finite Mixture Distributions*. Chichester: Wiley.

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.

Whittaker, J. (1990) *Graphical Models in Applied Multivariate Statistics*. Chichester: Wiley.

Yu, B. (1995) Estimating L^1 error of kernel estimator: monitoring convergence of Markov samplers. *Technical Report*. Department of Statistics, University of California, Berkeley.

Yu, B. and Mykland, P. (1997) Looking at Markov sampler through Cusum path plots: a simple diagnostic idea. *Statist. Comput.*, to be published.

Zellner, A. and Min, C. (1995) Gibbs sampler convergence criteria. *J. Am. Statist. Ass.*, **90**, 921–927.