SMSTC (2018/19)

Statistics

Lecture 17: MCMC Methods 2: Markov chain Monte Carlo

Valentin Popov
School of Mathematics and Statistics
University of St Andrews

www.smstc.ac.uk

Contents

17.1 Monte Carlo method

In many circumstances, we are faced with the problem of evaluating an integral that is too complex to calculate explicitly. For example, we may wish to estimate the posterior mean of a parameter, θ:

$$\mathbb{E}_\pi[f(\theta)] = \int f(\theta)\pi(\theta|\boldsymbol{x})d\theta.$$

We can use the simulation technique of *Monte Carlo Integration* to obtain an estimate of a given integral (and hence posterior expected value). The method is based upon drawing observations from the distribution of the variable of interest and simply calculating the empirical estimate of the mean. For example, given a sample of observations, $\theta^1, ..., \theta^n \sim \pi(\theta|\boldsymbol{x})$, we can estimate the integral

$$\mathbb{E}_\pi[f(\theta)] = \int f(\theta)\pi(\theta|\boldsymbol{x})d\boldsymbol{x}$$

by the ergodic average

$$\bar{f}_n = \frac{1}{n}\sum_{i=1}^{n} f(\theta^i). \tag{17.1}$$

This is Monte Carlo integration.

For independent samples, the Law of Large Numbers ensures that

$$\bar{f}_n \to \mathbb{E}_\pi[f(\theta)] \quad \text{as } n \to \infty.$$

Independent sampling from $\pi(\theta|\boldsymbol{x})$ may be difficult, however Equation (17.1) still holds if we generate our samples, not independently, but via some other method.

The above example concentrates on obtaining an estimate of the posterior mean of a distribution. However, any number of posterior summary statistics may be of interest. For example, suppose that we are interested in the posterior probability that the parameter of interest, θ has a value greater than 10. Then, we can estimate $\mathbb{P}_\pi(\theta > 10)$ by simply calculating the proportion of the sample from the posterior distribution for which the parameter value is greater than 10. Alternatively, the sample obtained can be used to plot the density of the (marginal) posterior distribution of θ. So, how do we obtain a sample from the posterior distribution, when in general, this will be very complex and often high-dimensional? One answer (and there are others) is via the use of a Markov chain.

17.2 Markov chains

A Markov chain is simply a stochastic sequence of numbers where each value in the sequence depends *only* upon the last. We might label the sequence $\theta^0, \theta^1, \theta^2$, etc. For example, we may specify,

$$\theta^{t+1} \sim N\left(\frac{\theta^t}{2}, 1\right),$$

where θ^0 is chosen from some (arbitrary) starting distribution. Then, what happens if we run the chain for a long time, more specifically, as $t \to \infty$? Is the value of θ^t dependent on θ^0? (Note: for interest as $t \to \infty$, $\theta^t \to N(0, 1.33)$).

More generally, we generate the new state of the chain, θ^{t+1} from some density, dependent only on θ^t:

$$\theta^{t+1} \sim \mathcal{K}(\theta^t, \theta) \quad \left(= \mathcal{K}(\theta|\theta^t)\right).$$

We call $\mathcal{K}$ the transition kernel for the chain. The transition kernel uniquely describes the dynamics of the chain.

In general, under certain conditions (the chain is aperiodic and irreducible), the Markov chain will converge to a *stationary* distribution (note we will always assume that these conditions are met).

Suppose that we have a Markov chain with stationary distribution $\pi(\theta)$. The Ergodic Theorem states that,

$$\bar{f}_n = \frac{1}{n}\sum_{i=1}^{n} f(\theta^t) \to \mathbb{E}_\pi(f(\theta)) \qquad \text{as } t \to \infty.$$

For our purposes $\pi(\theta) \equiv \pi(\theta|\boldsymbol{x})$ (i.e. the stationary distribution will be the posterior distribution of interest).

17.3 Markov chain Monte Carlo

Suppose that we can construct a Markov chain with stationary distribution equal to the posterior distribution of interest. In other words, at time t we update the state of the chain from θ^t to θ^{t+1}. This updating is performed in such a way that the probability distribution associated with the t^{th} observation gets closer and closer to $\pi(\theta|\boldsymbol{x})$ as t increases. We say that the distribution of the chain *converges* to π and we return to the concept of convergence later on. Thus, given that the chain has converged to the stationary distribution, realisations of the Markov chain can be regarded as a (dependent) sample from posterior distribution of interest. This sample can be used to obtain empirical (Monte Carlo) estimates of any posterior summaries of interest e.g., posterior means. This is the basic idea of Markov chain Monte Carlo (MCMC). Note that the initial values of the Markov chain, prior to convergence to the stationary distribution, are discarded as they will not be from the correct target distribution. This initial phase is called the *burn-in*, and we return to this issue later.

So how do we construct a Markov chain with a given stationary distribution? There are several standard approaches, and we cover these in the following subsections.

17.3.1 Gibbs Sampler

Suppose that we have the parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p) \in \mathbb{R}^p$, with distribution $\pi(\boldsymbol{\theta})$. The Gibbs Sampler uses the set of full conditionals of π to sample indirectly from the marginal distributions. Specifically, let $\pi(\theta_i|\boldsymbol{\theta}_{(i)})$ denote the induced full conditional of θ^i, given the values of the other components $\boldsymbol{\theta}_{(i)} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_k)$, $i = 1, \ldots, k$, $1 < k \leq p$. Then, given an arbitrary starting value $\boldsymbol{\theta}^0 = (\theta_1^0, \ldots, \theta_k^0)$, the Gibbs Sampler successively makes random drawings from the full conditional distributions $\pi(\theta_i|\boldsymbol{\theta}_{(i)})$, $i = 1, \ldots, k$, as follows:

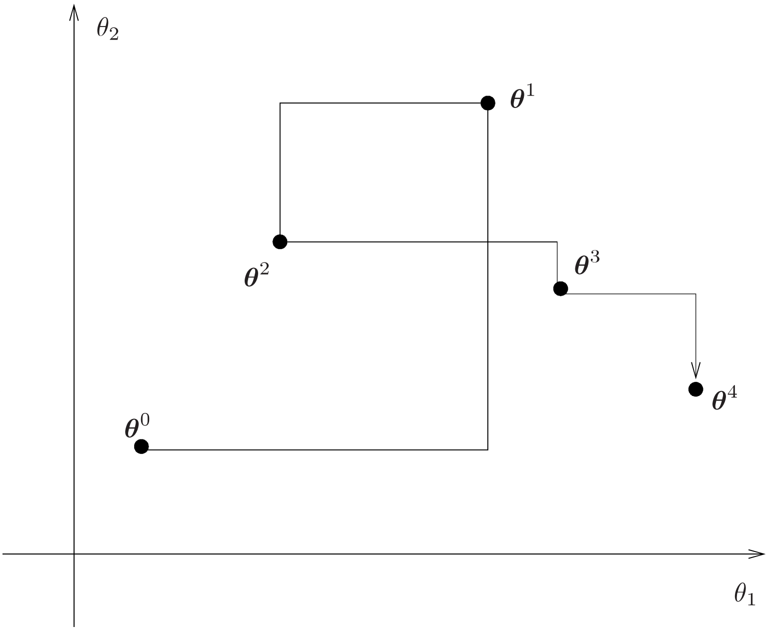
$$\begin{array}{lll}
\theta_1^1 & \text{is sampled from} & \pi(\theta_1|\boldsymbol{\theta}_{(1)}^0) \\
\theta_2^1 & \text{is sampled from} & \pi(\theta_2|\theta_1^1, \theta_3^0, \ldots, \theta_k^0) \\
. & . & . \\
. & . & . \\
. & . & . \\
\theta_i^1 & \text{is sampled from} & \pi(\theta_i|\theta_j^1,\ j < i \quad \theta_j^0,\ \text{and } j > i) \\
. & . & . \\
. & . & . \\
. & . & . \\
\theta_k^1 & \text{is sampled from} & \pi(\theta_k|\boldsymbol{\theta}_{(k)}^1).
\end{array}$$

This completes a transition from $\boldsymbol{\theta}^0$ to $\boldsymbol{\theta}^1$. Iteration of the full cycle of random variate generations from each of the full conditionals in turn, produces a sequence $\boldsymbol{\theta}^0, \boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^t, ...$ which is a realisation of a Markov chain with transition probability for going from $\boldsymbol{\theta}^t$ to $\boldsymbol{\theta}^{t+1}$ given by

$$\mathcal{K}_G(\boldsymbol{\theta}^t, \boldsymbol{\theta}^{t+1}) = \prod_{l=1}^{k} \pi(\theta_l^{t+1}|\theta_j^{t+1},\ j < l \text{ and } \theta_j^t,\ j > l), \tag{17.2}$$

and stationary distribution π.

Thus, in two dimensions a typical trajectory of the Gibbs Sampler may look something like that given in Figure 17.1.



PSfrag replacements

Figure 17.1: Typical Gibbs sampler path.

Conceptually, the Gibbs Sampler appears to be a rather straightforward algorithmic procedure. Ideally, each of the conditionals will be of the form of a standard distribution and suitable prior specification often ensures that this is the case.

Example

Suppose that $x_1, \ldots, x_n$ are independent, identically distributed (*iid*) observations from a $N(\mu, \tau^{-1})$ distribution. Then the likelihood is given by

$$f(\boldsymbol{x}|\mu, \tau) = \frac{\tau^{n/2}}{(2\pi)^{n/2}} \exp\left[-\frac{\tau}{2}\sum_{i=1}^{n}(x_i - \mu)^2.\right]$$

Now suppose that we have prior information on μ and τ, indicating that they are independent, μ being Normal (μ_0, σ_0^2) and τ being Gamma (α_0, β_0) so that

$$p(\mu, \tau) = p(\mu)p(\tau) = \frac{\exp(-(\mu - \mu_0)^2/2\sigma_0^2)}{\sqrt{2\pi\sigma_0^2}} \frac{\exp(-\beta_0\tau)\beta_0^{\alpha_0}\tau^{\alpha_0 - 1}}{\Gamma(\alpha_0)} \qquad \tau > 0 \quad \mu \in \mathbb{R}.$$

Then Bayes' Theorem gives us the posterior,

$$\begin{aligned} \pi(\mu, \tau|x_1, \ldots, x_n) &\propto f(\boldsymbol{x}|\mu, \tau)p(\mu, \tau) \\ &\propto \tau^{\frac{n}{2} + \alpha_0 - 1} \exp(-\beta_0\tau) \exp\left(-\frac{\tau S_n}{2} - \frac{(\mu - \mu_0)^2}{2\sigma_0^2}\right) \end{aligned}$$

where $S_n = \sum_{i=1}^{n}(x_i - \mu)^2$.

Note that the normalisation constant (or constant of proportionality) is unknown, and it is not easy to compute anything directly from this distribution. However, we can easily implement the Gibbs sampler and use MCMC to sample from the distribution.

Firstly, conditioning on τ, we have,

$$\pi(\mu|\tau, \boldsymbol{x}) \sim N\left(\frac{n\bar{x}\tau + \mu_0\tau_0}{n\tau + \tau_0}, (n\tau + \tau_0)^{-1}\right)$$

where $\tau_0 = (\sigma_0^2)^{-1}$.

Exercise: Check.

Similarly, conditioning on μ, we have,

$$\pi(\tau|\mu, \boldsymbol{x}) \sim \Gamma\left(\alpha_0 + \frac{n}{2}, \beta_0 + \frac{S_n}{2}\right).$$

Exercise: Check.

The Gibbs sampler for this example can be written algorithmically in the form:

Step 1. Conditional on the current parameter value, τ_t, generate a new value for μ, from the posterior conditional distribution,

$$\pi(\mu^{t+1}|\tau^t, \boldsymbol{x}) \sim N\left(\frac{n\bar{x}\tau^t + \mu_0\tau_0}{n\tau^t + \tau_0}, (n\tau^t + \tau_0)^{-1}\right).$$

Step 2. Conditional on the newly updated parameter value, μ_{t+1}, generate a new value for τ, from the posterior conditional distribution,

$$\pi(\tau^{t+1}|\mu^{t+1}, \boldsymbol{x}) \sim \Gamma\left(\alpha_0 + \frac{n}{2}, \beta_0 + \frac{\sum_{i=1}^n (x_i - \mu^{t+1})^2}{2}\right).$$

Step 3. Return to step 1. □

Computational problems arise when the posterior conditional distributions are not of standard form. It is still possible to use the Gibbs sampler, and sample from non-standard distributions, using, for example, rejection sampling. However, such algorithms are typically computationally intensive and inefficient. An alternative approach is the following.

17.3.2 Metropolis-Hastings algorithm

A general way to construct MCMC samplers is as a form of generalised rejection sampling, where values are drawn from approximate distributions and “corrected” in order that, asymptotically, they behave as random observations from the target distribution. This is the motivation for methods such as the Metropolis Hastings algorithm, which sequentially draws candidate observations from a distribution, conditional only upon the last observation, thus inducing a Markov chain. The most important aspect of such algorithms is not the Markov property, but the fact that the approximating candidate distributions are improved at each step in the simulation.

The method commonly known as the Metropolis Hastings algorithm, is based upon the observation that given a Markov chain with transition density $\mathcal{K}(\boldsymbol{\theta}, \boldsymbol{\phi})$ and exhibiting detailed balance for π i.e.,

$$\pi(\boldsymbol{\theta})\mathcal{K}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \pi(\boldsymbol{\phi})\mathcal{K}(\boldsymbol{\phi}, \boldsymbol{\theta}), \tag{17.3}$$

the chain has stationary density, $\pi(\cdot)$.

The candidate generating density (or proposal density) typically depends upon the current state of the chain, and we denote it by $q(\boldsymbol{\phi}|\boldsymbol{\theta}^t)$. The choice of proposal density is essentially arbitrary. However, in general, the induced chain will not satisfy the reversibility condition of (17.3), so we introduce an acceptance function $\alpha(\boldsymbol{\theta}^t, \boldsymbol{\phi})$. We then accept the candidate observation, and set $\boldsymbol{\theta}^{t+1} = \boldsymbol{\phi}$, with probability $\alpha(\boldsymbol{\theta}^t, \boldsymbol{\phi})$; else if the candidate observation is rejected, the chain remains at $\boldsymbol{\theta}^t$, so that $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t$.

It can be shown that the optimal form for the acceptance function, in the sense that suitable candidates are rejected least often and computational efficiency is maximised, is given by

$$\alpha(\boldsymbol{\theta}^t, \boldsymbol{\phi}) = \min\left(1, \frac{\pi(\boldsymbol{\phi})q(\boldsymbol{\theta}^t|\boldsymbol{\phi})}{\pi(\boldsymbol{\theta}^t)q(\boldsymbol{\phi}|\boldsymbol{\theta}^t)}\right),$$

so that the transition kernel is given by

$$\mathcal{P}_H(\boldsymbol{\theta}, A) = \int_A \mathcal{K}_H(\boldsymbol{\theta}, \boldsymbol{\phi})d\boldsymbol{\phi} + r(\boldsymbol{\theta})I_A(\boldsymbol{\theta}),$$

where

$$\begin{aligned} \mathcal{K}_H(\boldsymbol{\theta}, \boldsymbol{\phi}) &= q(\boldsymbol{\phi}|\boldsymbol{\theta})\alpha(\boldsymbol{\theta}, \boldsymbol{\phi}), \\ r(\boldsymbol{\theta}) &= 1 - \int_E q(\boldsymbol{\phi}|\boldsymbol{\theta})\alpha(\boldsymbol{\theta}, \boldsymbol{\phi})d\boldsymbol{\phi}, \end{aligned} \tag{17.4}$$

and $\mathcal{K}_H$ satisfies the reversibility condition of (17.3), implying that the kernel, $\mathcal{P}_H$ also preserves detailed balance for π.

Thus, the Metropolis Hastings method can be written algorithmically as follows.

Step 1. Given the current position, $\boldsymbol{\theta}^t = \boldsymbol{\theta}$, generate a new value, $\boldsymbol{\phi}$, from the distribution $q(\boldsymbol{\phi}|\boldsymbol{\theta})$.

Step 2. Calculate

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\phi}) = \min\left(1, \frac{\pi(\boldsymbol{\phi})q(\boldsymbol{\theta}|\boldsymbol{\phi})}{\pi(\boldsymbol{\theta})q(\boldsymbol{\phi}|\boldsymbol{\theta})}\right).$$

Step 3. With probability $\alpha(\boldsymbol{\theta}, \boldsymbol{\phi})$, set $\boldsymbol{\theta}^{t+1} = \boldsymbol{\phi}$, else set $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}$.

Step 4. Return to step 1.

Important notes:

17–1. We only need to know π up to proportionality, since any constants of proportionality cancel in the numerator and denominator of the calculation of α.

17–2. The performance of the MCMC algorithm is dependent on the choice of the proposal distribution q. If q is chosen poorly, then the number of rejections may be high, so that the efficiency of the procedure can be low; conversely if q is chosen such that only very small moves are proposed, it may take a very long time for the Markov chain to traverse the set of plausible posterior parameter values (see below example).

(Simple) Example

Suppose that we are interested in sampling from the standard normal distribution, and that we choose to use a proposal distribution of the form

$$q(\phi|\theta) \sim N(\theta, \sigma^2),$$

where σ^2 is to be specified. Then the acceptance probability is given by

$$\alpha(\theta, \phi) = \exp\left(-\frac{1}{2}(\phi^2 - \theta^2)\right), \tag{17.5}$$

Figure 17.2 plots the resulting output for $\sigma^2 = 0.1$, 1 and 10.

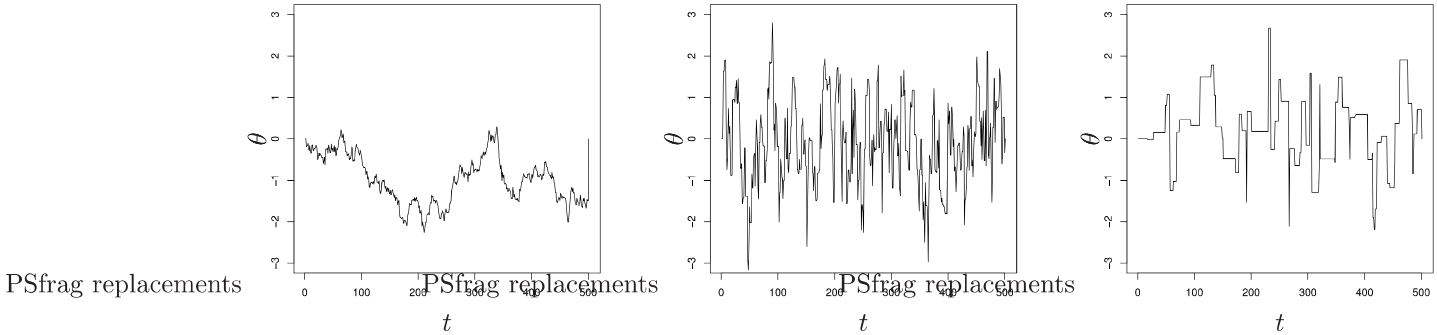


Figure 17.2: Sample paths for Metropolis Hastings algorithms with (a) $\sigma^2 = 0.1$, (b) $\sigma^2 = 1$ and (c) $\sigma^2 = 10$.

Notice that the acceptance function is independent of σ, but that the value of σ has a significant impact upon the acceptance rate of the chain. This is because the proposal with $\sigma^2 = 1$ is much closer to the target distribution, so that more sensible candidates are generated and subsequently accepted. However, the proposal with $\sigma^2 = 10$ generates candidate observations too far out in the tail to come from the target distribution and these are subsequently rejected. Conversely, the proposal with $\sigma^2 = 0.1$ generates candidates very similar to the current value that are typically accepted, but means that it takes a long time to move over the parameter space. The movement around the parameter space is often referred to as “mixing”. We return to this issue later (see Improving Performance). □

In the case where the candidate generating function is symmetric i.e., $q(\boldsymbol{\phi}|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\phi})$, the acceptance function reduces to

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\phi}) = \min\left(1, \frac{\pi(\boldsymbol{\phi})}{\pi(\boldsymbol{\theta})}\right). \tag{17.6}$$

This special case is the original Metropolis algorithm. There are a number of other special cases:

Random Walk Metropolis

If $q(\boldsymbol{\phi}|\boldsymbol{\theta}) = f(|\boldsymbol{\phi} - \boldsymbol{\theta}|)$ for some arbitrary density f, then the kernel driving the chain is a random walk, since the candidate observation is of the form $\boldsymbol{\phi}^{t+1} = \boldsymbol{\theta}^t + \boldsymbol{z}$, where $\boldsymbol{z} \sim f$. There are many common choices for f, including the uniform distribution on the unit disk, or a multivariate normal or t-distribution. Note that these are symmetric, so that the acceptance probability is of the simple form given in (17.6).

The Independence Sampler

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

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\phi}) = \min\left(1, \frac{w(\boldsymbol{\phi})}{w(\boldsymbol{\theta})}\right),$$

where $w(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})/f(\boldsymbol{\theta})$. (This is the importance weight function that would be used in importance sampling given observations generated from f.)

Single-updates and the Gibbs Sampler

The Metropolis Hastings algorithm need not update all variables in the sample space simultaneously, it can be used in stages, updating variables one-at-a-time, much like the Gibbs Sampler. This is commonly called the single-update Metropolis Hastings algorithm and, in fact, the Gibbs Sampler is a special case of the single-update Metropolis Hastings algorithm. Suppose that in the single-update Metropolis Hastings algorithm, we break each iteration of the algorithm into k steps, and let q_j denote a proposal for candidates in the jth co-ordinate direction, so that

$$q_j(\boldsymbol{\theta}, \boldsymbol{\phi}) = \begin{cases} \pi(\phi^j|\boldsymbol{\theta}^{(j)}) & \boldsymbol{\phi}^{(j)} = \boldsymbol{\theta}^{(j)}, \quad j = 1, ..., k. \\ 0 & \text{else} \end{cases}$$

With this proposal, the acceptance probability at the jth step is given by

$$\alpha_j(\boldsymbol{\theta}, \boldsymbol{\phi}) = \min\left(1, f(\boldsymbol{\theta}, \boldsymbol{\phi})\right),$$

where,

$$\begin{aligned} f(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \frac{\pi(\boldsymbol{\phi}) q_j(\boldsymbol{\phi}, \boldsymbol{\theta})}{\pi(\boldsymbol{\theta}) q_j(\boldsymbol{\theta}, \boldsymbol{\phi})} \\ &= \frac{\pi(\boldsymbol{\phi})/\pi(\phi^j|\boldsymbol{\theta}^{(j)})}{\pi(\boldsymbol{\theta})/\pi(\theta^j|\boldsymbol{\phi}^{(j)})} \\ &= \frac{\pi(\boldsymbol{\phi})/\pi(\phi^j|\boldsymbol{\phi}^{(j)})}{\pi(\boldsymbol{\theta})/\pi(\theta^j|\boldsymbol{\theta}^{(j)})}, \qquad \text{since } \boldsymbol{\phi}^{(j)} = \boldsymbol{\theta}^{(j)} \\ &= \frac{\pi(\boldsymbol{\phi}^{(j)})}{\pi(\boldsymbol{\theta}^{(j)})}, \\ &\quad \text{by definition of conditional probability for } \boldsymbol{\theta} = (\theta^j, \boldsymbol{\theta}^{(j)}) \\ &= 1, \text{ since } \boldsymbol{\phi}^{(j)} = \boldsymbol{\theta}^{(j)}. \end{aligned}$$

Thus, at each step, the only possible jumps are to parameter vectors $\boldsymbol{\phi}$, that match $\boldsymbol{\theta}$ on all components other than the jth, and these are automatically accepted.
Hence,

$$q(\boldsymbol{\phi}|\boldsymbol{\theta}) = \prod_{j=1}^{p} q_j([\boldsymbol{\phi}^{<j}, \boldsymbol{\theta}^{\geq j}], [\boldsymbol{\phi}^{\leq j}, \boldsymbol{\theta}^{>j}]),$$

where

$$\begin{aligned} \boldsymbol{\theta}^{>j} &= (\boldsymbol{\theta}^i, \ \ i > j), \\ \boldsymbol{\phi}^{<j} &= (\boldsymbol{\phi}^i, \ \ i < j) \end{aligned}$$

and, since $\alpha_j(\boldsymbol{\theta}, \boldsymbol{\phi}) = 1 \quad \forall j$, the transition distribution can, by simple manipulation, be re-written in the same form as the Gibbs transition density given in (17.2).

Comments

The Metropolis Hastings algorithm has the advantage over the Gibbs Sampler, in that it is not necessary to know all of the conditional distributions, we need only simulate from q, which we can choose arbitrarily. However, if q is poorly chosen, then the mixing of the Markov chain can be slow, so that the efficiency of the procedure can be low. Thus, the choice of q typically involves some pilot-tuning for the parameters within this distribution. Alternatively, the Gibbs sampler can be more difficult to implement, and computationally expensive, but requires no such pilot-tuning, since the proposal distribution is necessarily defined. In practice a combination of Metropolis-Hastings steps and Gibbs updates are often implemented. Parameters with standard posterior conditional distributions are updated via the Gibbs sampler; whereas parameters with non-standard posterior conditionals are updated using the Metropolis-Hastings algorithm.

17.3.3 Further issues

Here we discuss some issues associated with the practical implementation of the MCMC algorithm.

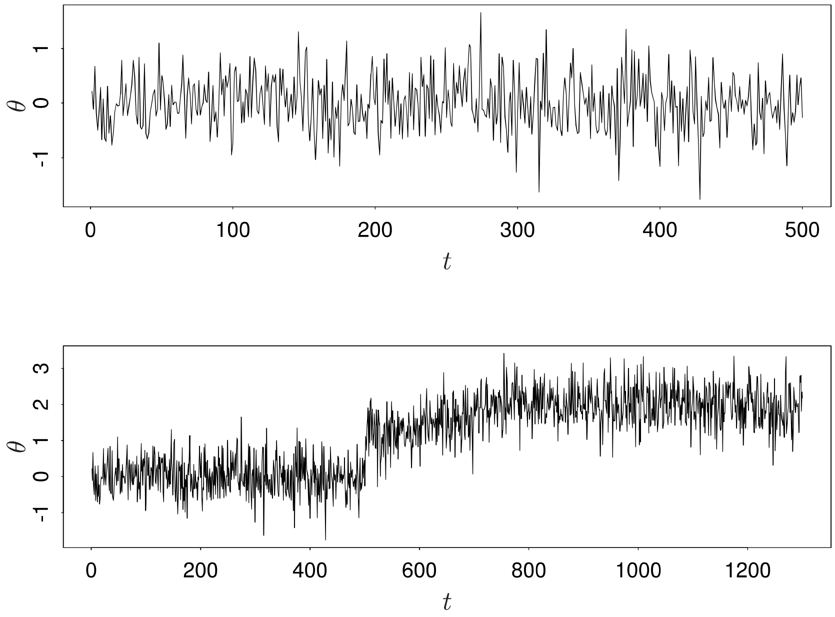
Run lengths

Recall that we use ergodic averages of realisations of the Markov chain to estimate parameters (or functions of parameters) of interest. However, the idea behind the MCMC algorithm is that the Markov chain constructed has stationary distribution equal to the correct target distribution. Thus, before using realisations of the Markov chain to estimate the parameters of interest, the Markov chain must have *converged* to the stationary distribution. In practice, we discard the observations from the start of the Markov chain and only base our inference on observations once the chain has converged to the stationary distribution. This initial period where we discard observations is called the *burn-in* period. Consequently, the posterior estimates obtained from the post burn-in period are independent of the initial values of the parameters ($\boldsymbol{\theta}^0$). However, how do we determine the length of burn-in that we need?

The simplest method to determine the length of burn-in needed is to look at the raw trace plots of the parameter values. Often, it is possible to see the individual parameters converging from their starting value to values consistent with the posterior distribution. This is a fairly efficient (and essentially common-sense) method but is not robust. For example, consider Figure 17.3 which plots the output from an MCMC sampler. The first 500 iterations appear to have already "settled down" to a constant distribution, so that we may have chosen a very short burn-in period of 100 (or 250) iterations, say. However, by plotting more iterations (see lower figure), this is clearly not the case. An alternative early technique (the so-called "thick-pen" approach) involves running multiple chains, with different (and typically over-dispersed) starting values. These are then plotted on a single graph. Taking a "thick-pen", the trace plot is followed, until the pen touches all lines of the plot. Once this occurs, the chain is assumed to have "converged".

More mathematical techniques have been developed. These often follow similar underlying ideas to the "thick-pen" test, but in a more rigorous and mathematical framework. One of the most popular is the Brooks-Gelman-Rubin technique. This is based on running multiple chains from different starting points and using an *Analysis of Variance* technique to assess whether or not each of the chains have the same distribution. (Note that this statistic can be calculated within OpenBUGS - see Lecture 18). Essentially, the underlying idea involves determining whether or not there are any differences in estimates from the different replications, allowing for natural variability due to the stochasticity of the Markov chains.

In practice, we note that convergence diagnostics can only "disprove" convergence and cannot "prove" that the stationary distribution has been reached. It is possible that a Markov chain may not have converged to the stationary distribution, but that the implemented convergence diagnostics do not indicate non-convergence. The simplest example to illustrate this is with a bimodal posterior distribution for a parameter, where only a single mode is explored by the Markov chain(s) implemented. This is one reason for running multiple chains from different starting values in an attempt to identify multiple modes. Thus, overall, running several replications from different starting points provides some reassurance when trying to check that convergence is achieved. In essence, if these multiple chains provide the same posterior estimates then this suggests that no major modes have been missed in any one simulation and that each has probably converged.



PSfrag replacements

Figure 17.3: MCMC sample paths.

Monte Carlo errors

MCMC is a simulation-based estimation technique for statistics of interest. Thus, it is subject to *Monte Carlo error*. This is essentially a measure of the expected variation in the parameter estimates if multiple replications of the MCMC algorithm are run. Monte Carlo error decreases with increasing sample size, and it is desirable to have a small Monte Carlo error relative to the scale of the parameter estimate (dependent on the precision of the estimate needed/reported).

Improving performance

The performance of the MCMC algorithm depends jointly on the target distribution of interest and the updating algorithm used. The target distribution is typically fixed (e.g. posterior distribution of interest), so that the performance of the algorithm can be changed only through the proposals used in the updating algorithm. With the exception of the Gibbs sampler, most MCMC updates require a degree of *pilot-tuning* to obtain a chain with good mixing properties. In practice, this often involves adjusting the relevant proposal variances to obtain a Metropolis-Hastings acceptance rate of 20-40% (Gelman *et al*, 1996). This can often be achieved by implementing a pilot run of the MCMC algorithm, for 1000 iterations, say, calculating the mean acceptance rate for each parameter and adjusting the proposal variance accordingly to obtain a mean acceptance rate in the given interval.

In addition, if parameters are highly correlated with each other (this is usually easy to assess from an initial pilot-run), multi-parameter updates can be used, proposing to update a number of parameters simultaneously. This is often referred to as *blocking*, since parameters are updated in "blocks". However, we note although this can overcome the problem of poor mixing due to highly correlated parameters, the specification of suitable multi-dimensional proposals can be difficult (the Multivariate Normal distribution is an obvious choice).