Chapter 16
Monte Carlo Markov Chains

Abstract Monte Carlo Markov Chains (MCMC) are a powerful method to analyze scientific data that has become popular with the availability of modern-day computing resources. The basic idea behind an MCMC is to determine the probability distribution function of quantities of interest, such as model parameters, by repeatedly querying datasets used for their measurement. The resulting sequence of values form a Markov chain that can be analyzed to find best-fit values and confidence intervals. The modern-day data analyst will find that MCMCs are an essential tool that permits tasks that are simply not possible with other methods, such as the simultaneous estimate of parameters for multi-parametric models of virtually any level of complexity, even in the presence of correlation among the parameters.

16.1 Introduction to Monte Carlo Markov chains

A typical data analysis problem is the fit of data to a model with adjustable parameters. Chapter 8 presented the maximum likelihood method to determine the best-fit values and confidence intervals for the parameters. For the linear regression to a two-variable dataset, in which the independent variable is assumed to be known and the dependent variable has errors associated with its measurements, we found an analytic solution for the best-fit parameters and its uncertainties (Sect. 8.3). Even the case of a multiple linear regressions is considerably more complex to solve analytically (Chap. 9) and most fits to non-linear functions do not have analytic solutions at all.

When an analytic solution is not available, the χ^2_{min} method to search for best-fit parameters and their confidence intervals is still applicable, as described in Sect. 10.3. The main complication is the computational cost of sampling the parameter space in search of χ^2_{min} and surfaces of constant $\chi^2_{min} + \Delta\chi^2$. Consider, for example, a model with 10 free parameters: even a very coarse sampling of 10 values for each parameter will result in 10^{10} evaluations of the likelihood, or χ^2, to cover the entire parameter space. Moreover, it is not always possible to improve the situation by searching for just a few interesting parameters at a time, e.g., fixing the value of the background while searching for the flux of the source. In fact, there may

M. Bonamente, *Statistics and Analysis of Scientific Data*, Graduate Texts in Physics, DOI 10.1007/978-1-4939-6572-4_16

be correlation among parameters and this requires that the parameters be estimated simultaneously.

The Monte Carlo Markov chain (MCMC) methods presented in this chapter provide a way to bypass altogether the need for a uniform sampling of parameter space. This is achieved by constructing a Markov chain that only samples the interesting region of parameters space, i.e., the region near the maximum of the likelihood. The method is so versatile and computationally efficient that MCMC techniques have become the leading analysis method in many fields of data analysis.

16.2 Requirements and Goals of a Monte Carlo Markov Chain

A Monte Carlo Markov chain makes use of a dataset Z and a model with m adjustable parameters, $\theta = (\theta_1, \ldots, \theta_m)$, for which it is possible to calculate the likelihood

$$\mathscr{L} = P(Z/\theta). \tag{16.1}$$

Usually, the calculation of the likelihood is the most intensive task for an MCMC. It necessary to be able to evaluate the likelihood for all possible parameter values.

According to Bayesian statistic, one is allowed to have a *prior* knowledge on the parameters, even before they are measured (see Sect. 1.7). The prior knowledge may come from experiments that were conducted beforehand, or from any other type a priori belief on the parameters. The prior probability distribution will be referred to as $p(\boldsymbol{\theta})$.

The information we seek is the probability distribution of the model parameters *after* the measurements are made, i.e., the posterior distribution $P(\theta/Z)$. According to Bayes' theorem, the posterior distribution is given by

$$P(\theta/Z) = \frac{P(\theta)P(Z/\theta)}{P(Z)} = \frac{P(\theta)\cdot\mathcal{L}}{P(Z)}, \tag{16.2}$$

where the quantity $P(Z) = \int P(Z/\theta)P(\theta)d\theta$ is a normalization constant.

Taken at face value, (16.2) appears to be very complicated, as it requires a multi-dimensional integration of the term $P(Z)$. The alternative provided by a Monte Carlo Markov chain is the construction of a sequence of *dependent* samples for the parameters θ in the form of a Markov chain. Such Markov chain is constructed in such a way that each parameter value appears in the chain in proportion to this posterior distribution. With this method, it will be shown that the value of the normalization constant $P(Z)$ becomes unimportant, thus alleviating significantly the computational burden. The goal of a Monte Carlo Markov chain is therefore that of creating a sequence of parameter values that has as its stationary distribution the

posterior distribution of the parameters. After the chain is run for a large number of iterations, the posterior distribution is obtained via the sample distribution of the parameters in the chain.

There are several algorithms to sample the parameter space that satisfy the requirement of having the posterior distribution of the parameters $P(\theta/Z)$ as the stationary distribution of the chain. A very common algorithm that can be used in most applications is that of Metropolis and Hastings [19, 32]. It is surprisingly easy to implement, and therefore constitutes a reference for any MCMC implementation. Another algorithm is that of Gibbs, but its use is limited by certain specific requirements on the distribution function of the parameters.Both algorithms presented in this chapter provide a way to sample values of the parameters and describe a way to accept them into the Markov chain.

16.3 The Metropolis–Hastings Algorithm

The Metroplis–Hastings algorithm [19, 32] was devised well before personal computers became of widespread use. In this section we first describe the algorithm and then prove that the resulting Markov chain has the desired stationary distribution. The method has the following steps.

1. The Metropolis–Hastings algorithm starts with an arbitrary choice of the initial values of the model parameters, $\theta_0 = (\theta_1^0, \ldots, \theta_m^0)$. This initial set of parameters is automatically accepted into the chain. As will be explained later, some of the initial links in the MCMC will later be discarded to offset the arbitrary choice of the starting point.
2. A *candidate* for the next link of the chain, θ', is then drawn from a *proposal (or auxiliary) distribution* $q(\theta'/\theta_n)$, where θ_n is the current link in the chain. The distribution $q(\theta'/\theta_n)$ is the probability of drawing a given candidate θ', given that the chain is in state θ_n. There is a large amount of freedom in the choice of the auxiliary distribution, which can depend on the current state of the chain θ_n, according to the Markovian property, but not on its prior history. One of the simplest choices for a proposal distribution is an m-dimensional uniform distribution of fixed width in the neighborhood of the current parameter. A uniform prior is very simple to implement, and it is the default choice in many applications. More complex candidate distributions can be implemented using, e.g., the method of simulation of variables described in Sect. 4.8.
3. A *prior distribution* $p(\theta)$ has to be assumed before a decision can be made whether the candidate is accepted into the chain or rejected. The Metropolis–Hastings algorithm gives freedom on the choice of the prior distribution as well. A typical choice of prior is another uniform distribution between two hard limits, enforcing a prior knowledge that a given parameter may not exceed certain boundaries. Sometimes the boundaries are set by nature of the parameter itself, e.g., certain parameter may only be positive numbers, or in a fixed interval range.

Other priors may be more restrictive. Consider the case of the measurement of the slope of the curve in the Hubble experiment presented on page 157. It is clear that, after a preliminary examination of the data, the slope parameter b will not be a negative number, and will not be larger than, say, $b = 2$. Therefore one can safely assume a prior on this parameter equal to $p(b) = 1/2$, for $0 \leq b \leq 2$. Much work on priors has been done by Jeffreys [23], in search of mathematical functions that express the lack of prior knowledge, known as *Jeffreys priors*. For many applications, though, simple uniform prior distributions are typically sufficient.

4. After drawing a random candidate θ', we must decide whether to accept it into the chain or reject it. This choice is made according to the following *acceptance probability*, which is the heart of the Metropolis–Hastings algorithm:

$$\alpha(\theta'/\theta_n) = \min\left\{\frac{\pi(\theta')q(\theta_n/\theta')}{\pi(\theta_n)q(\theta'/\theta_n)}, 1\right\}, \tag{16.3}$$

The acceptance probability $\alpha(\theta'/\theta_n)$ determines the probability of going from θ_n to the new candidate state θ', where $q(\theta'/\theta_n)$ is the proposal distribution, and $\pi(\theta') = P(\theta/Z)$ is the intended stationary distribution of the chain. Equation (16.3) means that the probability of going to a new value in the chain, θ', is proportional to the ratio of the posterior distribution of the candidate to that of the previous link. The acceptance probability can also be re-written by making use of Bayes' theorem (16.2), as

$$\alpha(\theta'/\theta_n) = \min\left\{\frac{p(\theta')P(Z/\theta')q(\theta_n/\theta')}{p(\theta_n)P(Z/\theta_n)q(\theta'/\theta_n)}, 1\right\} \tag{16.4}$$

In this form, the acceptance probability can be calculated based on known quantites. The term $p(\theta_n)q(\theta'/\theta_n)$ at the denominator represents the probability of occurrence of a given candidate θ'; in fact, the first term is the prior probability of the n-th link in the chain, and the second term is the probability of generating the candidate, once the chain is at that state. The other term, $\mathcal{L} = P(Z/\theta_n)$, is the likelihood of the current link in the chain. At the numerator, all terms have reverse order of conditioning between the current link and the candidate. Therefore all quantities in (16.4) are known, since $p(\theta_n)$ and $q(\theta'/\theta_n)$ (and their conjugates) are chosen by the analyst and the likelihood can be calculated for all model parameters.

Acceptance probability means that the candidate is accepted in the chain in proportion to the value of $\alpha(\theta'/\theta_n)$. Two cases are possible:

- $\alpha = 1$: This means that the candidate will be accepted in the chain, since the probability of acceptance is 100 %. The candidate becomes the next link in the chain, $\theta_{n+1} = \theta'$. The min operator guarantees that the probability is never greater than 1, which would not be meaningful.

- $\alpha < 1$: This means that the candidate can only be accepted in the chain with a probability α. To enforce this probability of acceptance, it is sufficient to draw a random number $0 \leq u \leq 1$ and then accept or reject the candidate according to the following criterion:

$$\begin{cases} \text{if } \alpha \geq u \Rightarrow \text{candidate is accepted, } \theta_{n+1} = \theta' \\ \text{if } \alpha < u \Rightarrow \text{candidate is rejected, } \theta_{n+1} = \theta_n \,. \end{cases} \tag{16.5}$$

  It is important to notice that if the candidate is rejected, then the chain doesn't move from its current location and a new link equal to the previous one is added to the chain. This means that at each time step in the chain a new link is added, either by repeating the last link (if the candidate is rejected) or by adding a different link (if the candidate is accepted).

The logic of the Metropolis–Hastings algorithm can be easily understood in the case of uniform priors and auxiliary distributions. In that case, the candidate is accepted in proportion to just the ratio of the likelihoods, since all other terms in (16.3) cancel out:

$$\alpha(\theta'/\theta_n) = \min\left\{\frac{\mathscr{L}(\theta')}{\mathscr{L}(\theta_n)}, 1\right\}. \tag{16.6}$$

If the candidate has a higher likelihood than the current link, it is automatically accepted. If the likelihood of the candidate is lower than the likelihood of the current link, then it is accepted in proportion to the ratio of the likelihoods of the candidate and of the current link. The possibility of accepting a parameter of *lower* likelihood permits a sampling of the parameter space, instead of a simple search for the point of maximum likelihood which would only result in a point estimate.

We now show that use of the Metropolis–Hastings algorithm creates a Markov chain that has $\pi(\theta_n) = P(\theta_n/Z)$ as its stationary distribution. For this purpose, we will show that the posterior distribution of the parameters satisfies the relationship

$$\pi(\theta_n) = \sum_j \pi(\theta_j) p_{jn}, \tag{16.7}$$

where p_{jn} are the transition probabilities of the Markov chain and the index j runs over all possible states.

Proof (Justification of the Metropolis–Hastings Algorithm) To prove that the Metropolis–Hastings algorithm leads to a Markov chain with the desired stationary distribution, consider the time-reversed chain:

$$\begin{aligned} &\text{original chain:} && X_0 \to X_1 \to X_2 \to \ldots \to X_n \to \ldots \\ &\text{time-reversed chain:} && X_0 \leftarrow X_1 \leftarrow \ldots X_n \leftarrow X_{n+1} \leftarrow \ldots . \end{aligned}$$

The time-reversed chain is defined by the transition probability $p_{ij}^{\star}$:

$$p_{ij}^{\star} = P(X_n = \varepsilon_j / X_{n+1} = \varepsilon_i) = \frac{P(X_n = \varepsilon_j, X_{n+1} = \varepsilon_i)}{P(X_{n+1} = \varepsilon_i)} = \frac{P(X_{n+1} = \varepsilon_i / X_n = \varepsilon_j) P(X_n = \varepsilon_j)}{P(X_{n+1} = \varepsilon_i)},$$

leading to the following relationship with the transition probability of the original chain:

$$\Rightarrow p_{ij}^{\star} = \frac{p_{ji} \pi(\theta_j)}{\pi(\theta_i)} \tag{16.8}$$

If the original chain is *time-reversible*, then $p_{ij}^{\star} = p_{ij}$, and the time-reversed process is also a Markov chain. In this case, the stationary distribution will follow the relationship

$$\pi(\theta_i) \cdot p_{ij} = p_{ji} \cdot \pi(\theta_j) \tag{16.9}$$

known as the equation of *detailed balance*. The detailed balance is the hallmark of a time-reversible Markov chain, stating that the probability to move forward and backwards is the same, once the stationary distribution is reached. Therefore, if the transition probability of the Metropolis–Hastings algorithm satisfies this equation, with $\pi(\theta) = P(\theta/Z)$, then the chain is time reversible, and with the desired stationary distribution. Moreover, Theorem 15.4 can be used to prove that this distribution is unique.

The Metropolis–Hastings algorithm enforces a specific transition probability between states θ_i and θ_j,

$$p_{ij} = q(\theta_j / \theta_i) \alpha(\theta_j / \theta_i) \qquad \text{if } \theta_i \neq \theta_j \tag{16.10}$$

where q is the probability of generating the candidate (or proposal distribution), and α the probability of accepting it. One can also show that the probability of remaining at the same state θ_i is

$$p_{ii} = 1 - \sum_{j \neq i} q(\theta_j / \theta_i) \alpha(\theta_j / \theta_i).$$

where the sum is over all possible states.

According to the transition probability described by (16.3),

$$\alpha(\theta_j/\theta_i) = \min\left\{\frac{p(\theta_j)P(Z/\theta_j)q(\theta_i/\theta_j)}{p(\theta_i)P(Z/\theta_i)q(\theta_j/\theta_i)}, 1\right\} = \min\left\{\frac{\pi(\theta_j)q(\theta_i/\theta_j)}{\pi(\theta_i)q(\theta_j/\theta_i)}, 1\right\}$$

in which we have substituted $\pi(\theta_i) \equiv p(\theta_i/Z) = P(Z/\theta_i)p(\theta_i)/p(Z)$ as the posterior distribution. Notice that the probability $p(Z)$ cancels out, therefore its value does not play a role in the construction of the chain.

It is clear that, if $\alpha(\theta_j/\theta_i) < 1$, then $\alpha(\theta_i/\theta_j) = 1$, thanks to the min operation. Assume, without loss of generality, that $\alpha(\theta_i, \theta_j) < 1$:

$$\alpha(\theta_j/\theta_i) = \frac{\pi(\theta_j)q(\theta_i/\theta_j)}{\pi(\theta_i)q(\theta_j/\theta_i)}$$

$$\Rightarrow \alpha(\theta_j/\theta_i)\pi(\theta_i)q(\theta_j/\theta_i) = \pi(\theta_j)q(\theta_i/\theta_j)\cdot\alpha(\theta_i/\theta_j)$$

Now, since we assumed $\alpha(\theta_j/\theta_i) < 1$, the operation of min becomes redundant. Using (16.10) the previous equation simplifies to

$$p_{ij}\cdot\pi(\theta_i) = p_{ji}\cdot\pi(\theta_j)$$

which shows that the Metropolis–Hastings algorithm satisfies the detailed balance equation; it thus generates a time-reversible Markov chain, with stationary distribution equal to the posterior distribution. □

Example 16.1 The data from Hubble's experiment (page 157) can be used to run a Monte Carlo Markov chain to obtain the posterior distribution of the parameters a and b. This fit was also performed using a maximum likelihood method (see page 159) in which the common uncertainty in the dependent variable, $\log v$, was estimated according to the method described in Sect. 8.5.

Using these data, a chain is constructed using uniform priors on the two fit parameters a and b:

$$\begin{cases} p(a) = \dfrac{10}{7} & \text{for } 0.2 \le b \le 0.9 \\ p(b) = 10 & \text{for } 0.15 \le a \le 0.25. \end{cases}$$

The proposal distributions are also uniform distributions, respectively, of fixed width 0.1 and 0.02 for a and b, and centered at the current value of the parameters:

$$\begin{cases} p(\theta_{n+1}/a_n) = 5 & \text{for } a_n - 0.1 \le \theta_{n+1} \le a_n + 0.1 \\ p(\theta_{n+1}/b_n) = 25 & \text{for } b_n - 0.02 \le \theta_{n+1} \le b_n + 0.02 \end{cases}$$

in which a_n and b_n are, respectively, the n-th links of the chain, and θ_{n+1} represent the candidate for the $(n+1)$-th link of the chain, for each parameter.

In practice, once the choice of a uniform distribution with fixed width is made, the actual value of the prior and proposals distributions are not used explicitly. In fact, the acceptance probability becomes simply a function of the ratio of the likelihoods, or of the χ^2's:

$$\alpha(\theta'/\theta_n) = \min\left\{\frac{\mathscr{L}(\theta')}{\mathscr{L}(\theta_n)}, 1\right\} = \min\left\{e^{\frac{\chi^2(\theta_n)-\chi^2(\theta')}{2}}, 1\right\}$$

◇

where $\chi^2(\theta_n)$ and $\chi^2(\theta')$ are the minimum χ^2's calculated, respectively, using the n-th link of the chain and the candidate parameters (Fig. 16.1).

A few steps of the chain are reported in Table 16.1. Where two consecutive links in the chain are identical, it is an indication that the candidate parameter drawn at that iteration was rejected, and the previous link was therefore repeated. Figure 16.2 shows the sample distributions of the two fit parameters from a chain with 100,000 links. A wider prior on parameter a would make it possible to explore further the tails of the distribution.

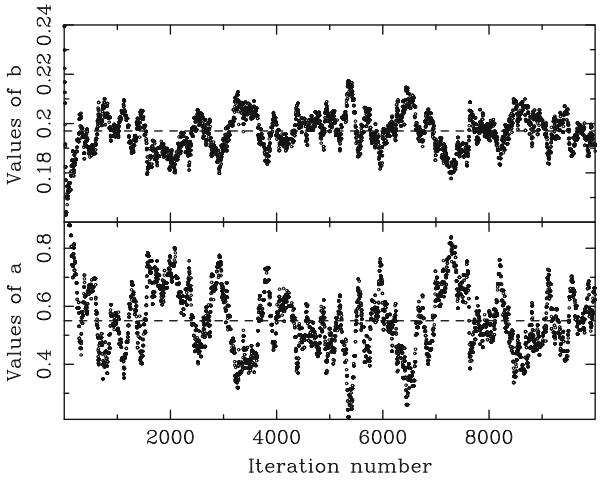

Fig. 16.1 MCMC for parameters a, b of linear model fit to the data in Table 8.1. The chain was run for 10,000 iterations, using uniform priors on both parameters (between 0.15 and 0.25 for a, and 0.2 and 0.9 for b). The chain started at $a = 0.90$ and $b = 0.25$. The proposal distributions were also uniform, with width of, respectively, 0.2 for a and 0.04 for b, centered at the current value of the chain

Table 16.1 Sample of MCMC chain for the Hubble data

n	a	b	$\chi^2(\theta_n)$				
1	0.90000	0.25000	3909.55420	136	0.80627	0.18064	11.47313
2	0.94116	0.24395	3563.63110	137	0.77326	0.18284	10.63887
3	0.96799	0.23951	3299.28149	138	0.77326	0.18284	10.63887
4	0.96799	0.23951	3299.28149	139	0.77326	0.18284	10.63887
5	0.96799	0.23951	3299.28149	140	0.77326	0.18284	10.63887
6	0.96799	0.23951	3299.28149	.....			
7	0.97868	0.22983	2503.21655	1141	0.42730	0.20502	8.90305
8	0.97868	0.22983	2503.21655	1142	0.42730	0.20502	8.90305
9	0.96878	0.22243	1885.28088	1143	0.42174	0.20494	8.68957
10	1.01867	0.21679	1714.54456	1144	0.42174	0.20494	8.68957
.....				1145	0.42174	0.20494	8.68957
21	1.08576	0.19086	563.56506	1146	0.42174	0.20494	8.68957
22	1.06243	0.19165	536.47919	1147	0.42174	0.20494	8.68957
23	1.06243	0.19165	536.47919	1148	0.42174	0.20494	8.68957
24	1.06559	0.18244	254.36528	1149	0.42174	0.20494	8.68957
25	1.06559	0.18244	254.36528	1150	0.43579	0.20323	8.65683
26	1.06559	0.18244	254.36528	.....			
27	1.06559	0.18244	254.36528	9991	0.66217	0.19189	12.43171
28	1.06559	0.18244	254.36528	9992	0.62210	0.19118	8.52254
29	1.04862	0.17702	118.84048	9993	0.62210	0.19118	8.52254
30	1.04862	0.17702	118.84048	9994	0.62210	0.19118	8.52254
.....				9995	0.62210	0.19118	8.52254
131	0.84436	0.17885	13.11242	9996	0.62210	0.19118	8.52254
132	0.84436	0.17885	13.11242	9997	0.62210	0.19118	8.52254
133	0.84436	0.17885	13.11242	9998	0.62210	0.19118	8.52254
134	0.80627	0.18064	11.47313	9999	0.64059	0.18879	11.11325
135	0.80627	0.18064	11.47313	10,000	0.64059	0.18879	11.11325

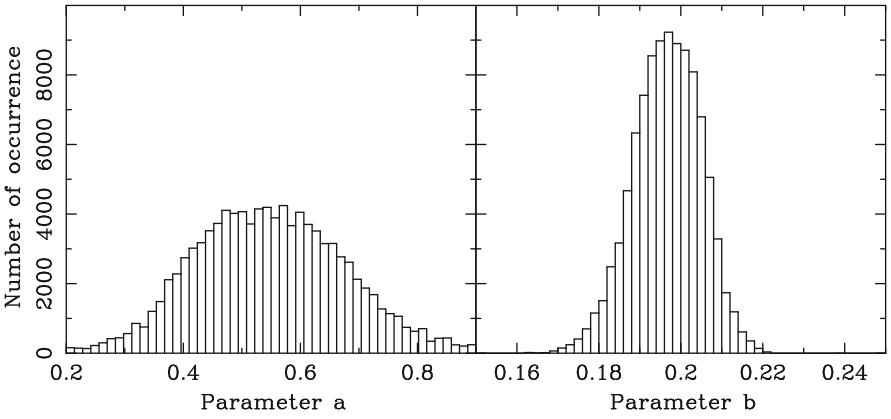

Fig. 16.2 Sample distribution function for parameters a and b, constructed using a histogram plot of 100,000 samples of a MCMC ran with the same parameters as Fig. 16.1

16.4 The Gibbs Sampler

The Gibbs sampler is another algorithm that creates a Markov chain having as stationary distribution the posterior distribution of the parameters. This algorithm is based on the availability of the *full conditional distribution*, defined as

$$\pi_i(\theta_i) = \pi(\theta_i|\theta_j, j \neq i) \tag{16.11}$$

The full conditional distribution is the (posterior) distribution of a given parameter, given that the values of all other parameters are known. If the full conditional distributions are known and can be sampled from, then a simple algorithm can be implemented:

1. Start the chain at a given value of the parameters, $\theta_0 = (\theta_0^1, \ldots, \theta_0^m)$.
2. Obtain a new value in the chain through successive generations:

$$\theta_1^1 \text{ drawn from } \pi(\theta_1|\theta_0^2, \theta_0^3, \ldots)$$

$$\theta_1^2 \text{ drawn from } \pi(\theta_2|\theta_1^1, \theta_0^3, \ldots)$$

$$\ldots$$

$$\theta_1^m \text{ drawn from } \pi(\theta_m|\theta_1^1, \theta_1^2, \ldots, \theta_1^{m-1})$$

3. Iterate until convergence to stationary distribution is reached.

The justification of this method can be found in [15]. In the case of data fitting with a dataset Z and a model with m adjustable parameters, usually it is not possible to know the full conditional distributions, thus this method is not as common as the Metropolis–Hastings algorithm. The great advantage of the Gibbs sampler is the fact that the acceptance is 100 %, since there is no rejection of candidates for the Markov chain, unlike the case of the Metropolis–Hastings algorithm.

Example 16.2 This example reproduces an application presented by Carlin et al. [8], and illustrates a possible application in which the knowledge of the full conditional distribution results in the possibility of implementing a Gibbs sampler.

Consider the case in which a Poisson dataset of n numbers, y_i, $i = 1, \ldots, n$, is fit to a step-function model:

$$\begin{cases} y = \lambda \text{ if } i \leq \text{m} \\ y = \mu \text{ if } i > \text{m} \end{cases} \tag{16.12}$$

The model therefore has three parameters, the values λ, μ, and the point of discontinuity, m. This situation could be a set of measurements of a quantity that may suddenly change its value at an unknown time, say the voltage in a given portion of an electric circuit after a switch has been opened or closed.

Assume that the priors on the parameters are, respectively, a gamma distributions for λ and μ, $p(\lambda) = G(\alpha, \beta)$ and $p(\mu) = G(\gamma, \delta)$, and a uniform distribution for m, $p(m) = 1/n$ (see Sect. 7.2 for definition of the gamma distribution). According to Bayes' theorem, the posterior distribution is proportional to the product of the likelihood and the priors:

$$\pi(\lambda, \mu, m) \propto P(y_1, \ldots, y_n/\lambda, \mu, m) \cdot p(\lambda)p(\mu)p(m). \tag{16.13}$$

The posterior is therefore given by

$$\pi(\lambda, \mu, m) \propto \prod_{i=1}^{m} e^{-\lambda}\lambda^{y_i} \prod_{i=m+1}^{n} e^{-\mu}\mu^{y_i} \cdot \lambda^{\alpha-1}e^{-\beta\lambda} \cdot \mu^{\gamma-1}e^{-\delta\mu} \cdot \frac{1}{n}$$

$$\Rightarrow \pi(\lambda, \mu, m) \propto \lambda^{\left(\alpha+\sum_{i=1}^{m} y_i-1\right)}e^{-(\beta+m)\lambda} \cdot \mu^{\left(\gamma+\sum_{i=m+1}^{n} y_i-1\right)}e^{-(\delta+n-m)\mu}.$$

The equation above indicates that the conditional posteriors, obtained by fixing all parameters except one, are given by

$$\begin{cases} \pi_\lambda(\lambda) = G\left(\alpha + \sum_{i=1}^{m} y_i, \beta + m\right) \\ \pi_\mu(\mu) = G\left(\gamma + \sum_{i=m+1}^{n} y_i, \delta + n - m\right) \\ \pi_m(m) = \dfrac{\lambda^{\left(\alpha+\sum_{i=1}^{m} y_i-1\right)}e^{-(\beta+m)\lambda} \cdot \mu^{\left(\gamma+\sum_{i=m+1}^{n} y_i-1\right)}e^{-(\delta+n-m)\mu}}{\sum_{i=1}^{n}\left(\lambda^{\left(\alpha+\sum_{i=1}^{m} y_i-1\right)}e^{-(\beta+m)\lambda} \cdot \mu^{\left(\gamma+\sum_{i=m+1}^{n} y_i-1\right)}e^{-(\delta+n-m)\mu}\right)}. \end{cases} \tag{16.14}$$

This is therefore an example of a case where the conditional posterior distributions are known, and therefore the Gibbs algorithm is applicable. The only complication is the simulation of the three conditional distributions, which can be achieved using the methods described in Sect. 4.8. ◇

16.5 Tests of Convergence

It is necessary to test that the MCMC has reached convergence to the stationary distribution before inference on the posterior distribution can be made. Convergence indicates that the chain has started to sample the posterior distribution, so that the MCMC samples are representative of the distribution of interest, and are not biased by such choices as the starting point of the chain. The period of time required for the chain to reach convergence goes under the name of *burn-in* period, and varies from chain to chain according to a variety of factors, such as the choice of prior and proposal distributions. We therefore must identify and remove such initial period

from the chain prior to further analysis. The *Geweke z-score test* and the *Gelman-Rubin test* are two of the most common tests used to identify the burn-in period.

Another important consideration is that the chain must be run for a sufficient number of iterations, so that the sample distribution becomes a good approximation of the true posterior distribution. It is clear that the larger the number of iterations after the burn-in period, the more accurate will be the estimates of the parameters of the posterior distribution. In practice it is convenient to know the minimum *stopping time* that enables to estimate the posterior distribution with the required precision. The *Raftery-Lewis test* is designed to give an approximate estimate of both the burn-in time and the minimum required stopping time.

Typical considerations concerning the burn-in period and the stopping time of a chain can be illustrated with the example of three chains based on the data from Table 10.1. The chains were run, respectively, with a uniform proposal distribution of 1, 10, and 100 for both parameters of the linear model, starting at the same point (Figs. 16.3, 16.4 and 16.5). The chain with a narrower proposal distribution requires a longer time to reach the stationary value of the parameters, in part because at each time interval the candidate can be chosen in just a limited neighborhood of the

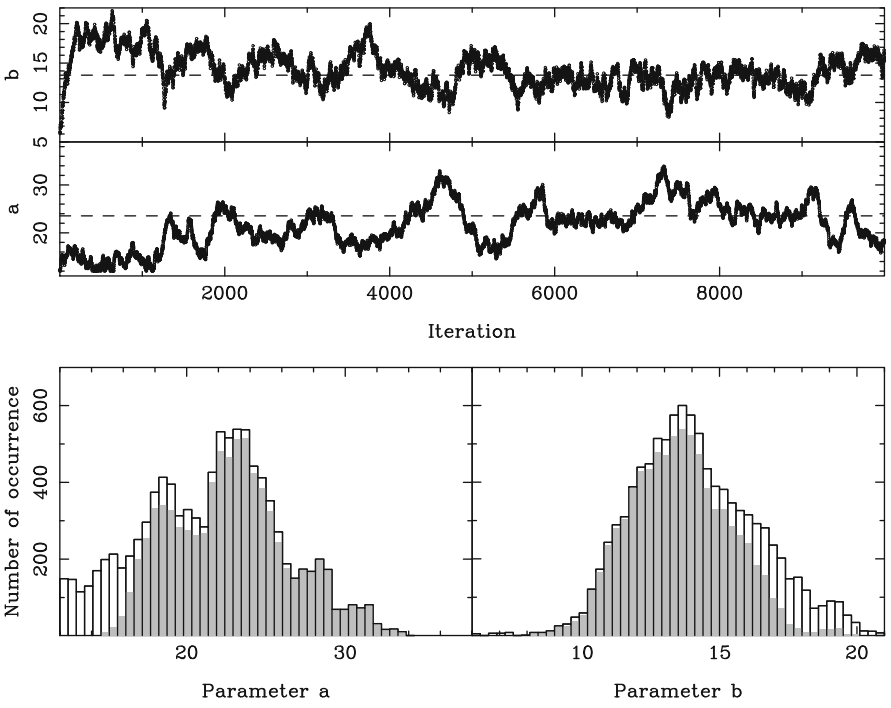

Fig. 16.3 MCMC for parameters a, b of linear model fit to the data in Table 10.1, using a uniform proposal distribution with width of 1 for both parameters. The chain started at $a = 12$ and $b = 6$. In *grey* is the sample distribution obtained by removing the initial 2000 iterations, the ones that are most affected by the arbitrary choice of starting point

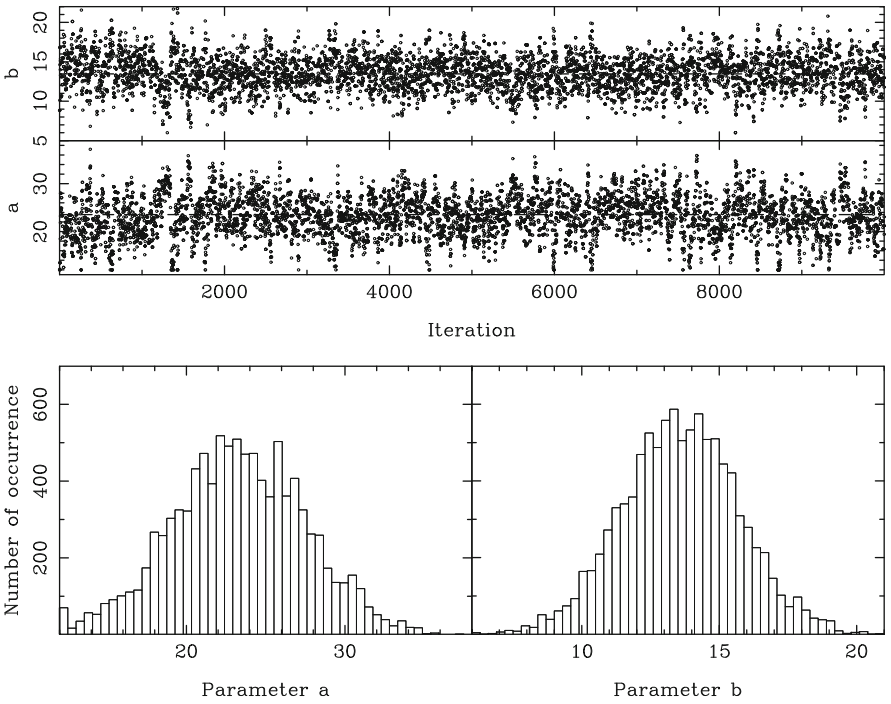

Fig. 16.4 MCMC for parameters a, b of linear model fit to the data in Table 10.1, using a uniform proposal distribution with width of 10 for both parameters. The chain started at same values as in Fig. 16.3

previous link. Moreover, the sampling of parameter space is less uniform because the chain requires longer time to span the entire parameter range. The intermediate value for the proposal distribution results in an almost immediate convergence, and the sampling of parameter space is clearly more uniform. An increase in the size of the proposal distribution, however, may eventually lead to slow convergence and poor sampling, as indicated by the chain with the largest value of the proposal width. In this case, candidates are drawn from regions of parameter space that have very low likelihood, or large χ^2, and therefore the chain has a tendency to remain at the same location for extended periods of time, with low acceptance rate. The result is a chain with poor coverage of parameter space and poorly determined sample distribution for their parameters. A smoother distribution is preferable, because it leads to a more accurate determination of the median, and of confidence ranges on the parameters.

Another consideration is that elements in the chain are more or less correlated to one another, according to the choice of the proposal distribution, and other choices in the construction of the chain. Links in the chains are always correlated by construction, since the next link in the chain typically depends on the current state of the chain. In principle a Markov chain can be constructed that does not

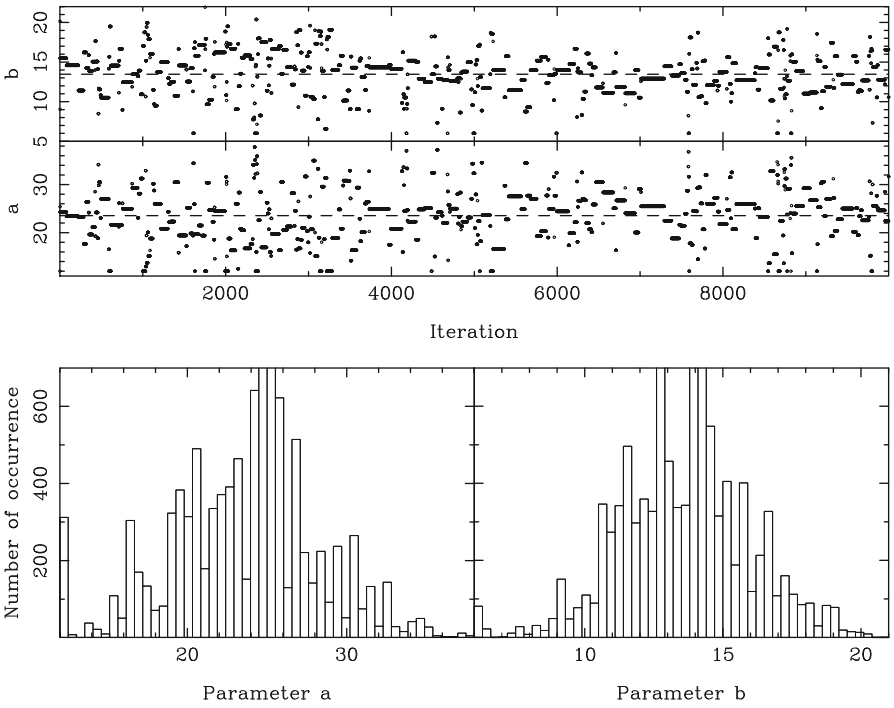

Fig. 16.5 MCMC for parameters a, b of linear model fit to the data in Table 10.1, using a uniform proposal distribution with width of 50 for both parameters. The chain started at same values as in Fig. 16.3

depend on the current state of the chain, but in most cases it is convenient to make full use of the Markovian property that allows to make use of the current state of the chain. The chains in Figs. 16.3, 16.4 and 16.5 illustrate the fact that the degree of correlation varies with the proposal distribution choice. For example, the chain with the narrowest proposal distribution appears more correlated than that with the intermediate choice for the width; also, the chain with the largest width appears to have periods with the highest degree of correlation, when the chain does not move for hundreds of iterations. This shows that the degree of correlation is a nonlinear function of the proposal distribution width, and that fine-tuning is always required to obtain a chain with good *mixing* properties. The degree of correlation among elements of the chain will become important when we desire to estimate the variance of the mean from a specific segment of the chain, since the formulas derived earlier in Chap. 4 apply only to independent samples.

Testing for convergence and stopping time of the chain are critical tasks for a Monte Carlo Markov chain. The tests discussed below are some of the more common analytic tools and can be implemented with relative ease.

16.5.1 The Geweke Z Score Test

A simple test of convergence is provided by the difference of the mean of two segments of the chain. Under the null hypothesis that the chain is sampling the same distribution during both segments, the sample means are expected to be drawn from the same parent mean. Consider segment A at the beginning of the chain, and segment B at the end of the chain; for simplicity, consider one parameter ψ at a time. If the chain is of length N, the prescription is to use an initial segment of $N_A = 0.1N$ elements, and a final segment with $N_B = 0.5N$ links, although those choices are somewhat arbitrary, and segments of different length can also be used.

The mean of each parameter in the two segments A and B is calculated as

$$\begin{cases} \overline{\psi_A} = \dfrac{1}{N_A} \displaystyle\sum_{j=1}^{N_A} \psi_j \\ \overline{\psi_B} = \dfrac{1}{N_B} \displaystyle\sum_{j=N-N_B+1}^{N} \psi_j. \end{cases} \tag{16.15}$$

To compare the two sample means, it is also necessary to estimate their sample variances $\sigma^2_{\overline{\psi_A}}$ and $\sigma^2_{\overline{\psi_B}}$. This task is complicated significantly by the fact that one *cannot* just use (2.11), because of the correlation between links of the chain. One possibility to overcome this difficulty is to *thin* the chain by using only every n-th iteration, so that the thinned chain better approximates independent samples.

The test statistic is the Z *score* of the difference between the means of the two segments:

$$Z_G = \frac{\overline{\psi_B} - \overline{\psi_A}}{\sqrt{\sigma^2_{\overline{\psi_A}} + \sigma^2_{\overline{\psi_B}}}}. \tag{16.16}$$

Under the assumption that the two means follow the same distribution and that they are uncorrelated, the Z-score is distributed as a standard Gaussian, $Z_G \sim N(0, 1)$. For this reason the two segments of the chain are typically separated by a large number of iterations. An application of the Geweke Z score is to step the start of segment A forward in time, until the Z_G scores don't exceed approximately ± 3, which correspond to a $\pm 3\sigma$ deviation in the means of the two segments. The burn-in period that needs to be excised is that before the Z scores stabilize around the expected values. An example of the use of this test statistic is provided in Fig. 16.6, in which Z_G was calculated from the chain with proposal width 10. An initial segment of length 20 % of the total chain length is compared to the final 40 % of the chain, by stepping the beginning of the initial segment until it overlaps with the final segment. By using all links in the chain to estimate the variance of the

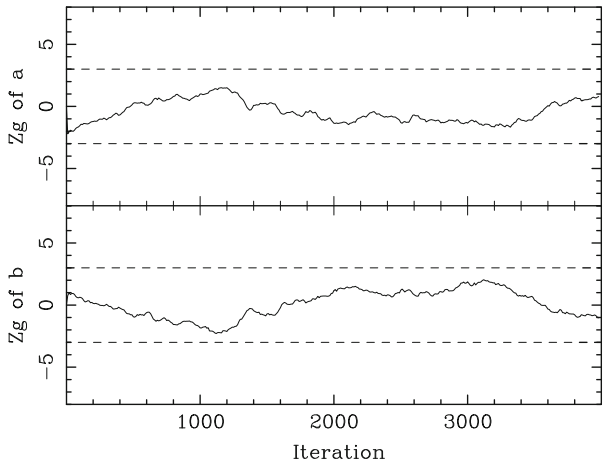

Fig. 16.6 Geweke Z scores with segment *A* and segment *B*, respectively, 20 and 40 % of the total chain length. The results correspond to the chain run with a proposal width of 10. The Z scores are calculated by using only every other 10-th iteration

mean, the variance would be underestimated because of the correlation among links, leading to erroneously large values of Z_G. If the chain is thinned by a factor of 10, then the estimate of the variance using (2.11) is more accurate, and the resulting Z scores show that the chains converge nearly immediately, as is also clear by a visual inspection of the chain from Fig. 16.4.

The effect of the starting point in the evaluation of the burn-in period is shown in Fig. 16.3, in which it is apparent that it takes about 2000 iterations for the chain to forget the initial position, and to start sampling the posterior distribution, centered at the dashed lines. A larger proposal distribution, as in Fig. 16.4, makes it easier to reach the posterior distribution more rapidly, to the point that no burn-in period is visible in this case. In the presence of a burn-in period, the sample distribution must be constructed by excising the initial portion of the chain, as shown in the grey histogram plot of Fig. 16.3.

16.5.2 The Gelman–Rubin Test

The Gelman–Rubin test investigates the effect of initial conditions on the convergence properties of the MCMC and makes use of m parallel chains starting from different initial points. Initially, the m chain will be far apart because of the different starting points. As the chains start sampling the stationary distribution, they will have the same statistical properties.

The test is based on two estimates of the variance, or variability, of the chains: the *within-chain* variance for each of the m chains W, and the *between-chain* variance B. At the beginning of the chain, W will underestimate the true variance of the model parameters, because the chains have not had time to sample all possible values. On the other hand, B will initially overestimate the variance, because of the different starting points. The test devised by Gelman and Rubin [17] defines the ratio of the within-to-between variance as a test to measure convergence of the chains, to identify an initial burn-in period that should be removed because of the lingering effect of initial conditions.

For each parameter, consider m chains of N iterations each, where $\bar{\psi}_i$ is the mean of each chain $i = 1, \ldots, m$ and $\bar{\bar{\psi}}$ the mean of the means:

$$\begin{cases} \bar{\psi}_i = \dfrac{1}{N} \displaystyle\sum_{j=1}^{N} \psi_j \\ \bar{\bar{\psi}} = \dfrac{1}{m} \displaystyle\sum_{i=1}^{m} \bar{\psi}_i. \end{cases} \tag{16.17}$$

The between-chain variance B is defined as the average of the variances of the m chains,

$$B = \frac{N}{m-1} \sum_{i=1}^{m} (\bar{\psi}_i - \bar{\bar{\psi}})^2 \tag{16.18}$$

Notice that, in (16.18), B/N is the variance of the means $\bar{\psi}_i$. The within-chain variance W is defined by

$$\begin{aligned} s_i^2 &= \frac{1}{N-1} \sum_{j=1}^{N} (\psi_j - \bar{\psi}_i)^2 \\ W &= \frac{1}{m} \sum_{i=1}^{m} s_i^2. \end{aligned} \tag{16.19}$$

The quantity $\hat{\sigma^2_{\psi}}$, defined as

$$\hat{\sigma^2_{\psi}} = \left(\frac{N-1}{N} \right) W + \frac{1}{N} B \tag{16.20}$$

is intended to be an unbiased estimator of the variance of the parameter ψ under the hypothesis that the stationary distribution is being sampled. At the beginning of a chain—before the stationary distribution is reached—$\hat{\sigma^2_{\psi}}$ overestimates the variance, because of the different initial starting points. It was suggested by Brooks and Gelman [6] to add an additional term to this estimate of the variance, to account

for the variability in the estimate of the means, so that the estimate of the within-chain variance to use becomes

$$\hat{V} = \hat{\sigma_{\psi}^2} + \frac{B}{mN}. \tag{16.21}$$

Convergence can be monitored by use of the following statistic:

$$\sqrt{\hat{R}} \equiv \sqrt{\frac{\hat{V}}{W}}, \tag{16.22}$$

which should converge to 1 when the stationary distribution in all chains has been reached. A common use of this statistic is to repeat the calculation of the Gelman–Rubin statistic after excising an increasingly longer initial portion of the chain, until approximately

$$\sqrt{\hat{R}} \leq 1.2. \tag{16.23}$$

A procedure to test for convergence of the chain using the Gelman–Rubin statistic is to divide the chain into segments of length b, such that the N iterations are divided into N/b batches. For each segment starting at iteration $i = k \times b, k = 0, \ldots, N/b - 1$, we can calculate the value $\hat{R}$ and claim convergence of the chains when (16.23) is satisfied. Figure 16.7 shows results of this test run on $m = 2$ chains

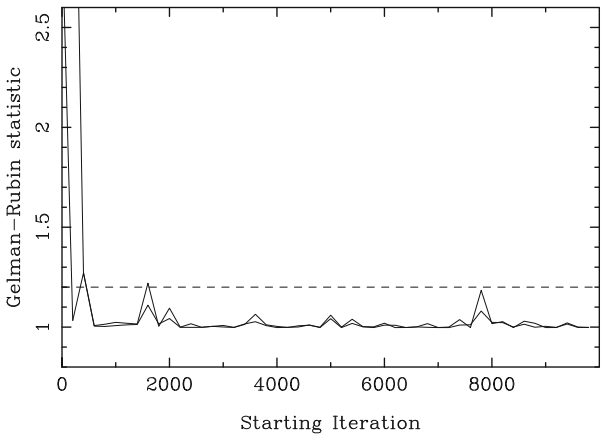

Fig. 16.7 Gelman–Rubin statistic $\hat{R}$ calculated from $m = 2$ chains with the same distributions as in Fig. 16.3, one starting at $a = 12$, $b = 6$, and the other at $a = 300$ and $b = 300$. The chain rapidly converges to its stationary distribution, and appears to forget about the starting point after approximately 500 iterations. The values of $\hat{R}$ were calculated in segments of length $b = 200$, starting at iteration $i = 0$

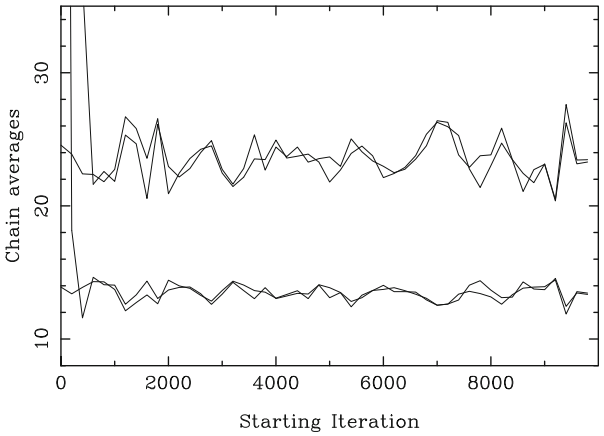

Fig. 16.8 Plot of the average of the parameters a and b for the two chains used in Fig. 16.7 (*top lines* are for parameter a, *bottom lines* for b). For both parameters, the two chains sample the same posterior distribution after approximately 500 iterations

based on the data of Table 10.1, starting at different values: one chain starting at a value that is close to the posterior mean of the parameters, and one starting at values that were intentionally chosen to be much larger than the parent values. After approximately 500 or so iterations, the within-chain and between-chain estimates of the variance become comparable, and the value of $\hat{R}$ approaches the value of one.

Another related tool that aids in assessing convergence is the plot of the mean of the parameters, shown in Fig. 16.8: it takes approximately 500 iterations for the chain starting with high initial values, to begin sampling the stationary distribution. It is clear that, from that point on, both chains hover around similar values of the parameters. One should also check that, individually, both $\hat{V}$ and W also stabilize to a common value as function of starting point of the batch, and not just their ratio $\hat{R}$. In fact, under the hypothesis of convergence, both within-chain and between-chain variances should converge to a common value.

Similar procedures to monitor convergence using the Gelman–Rubin may instead use batches of increasing length, starting from one of length b, and increasing to $2b$, etc., optionally discarding the first half of each batch. Moreover, thinning can be implemented when calculating means, to reduce the effect of correlation among the samples. In all cases, the goal is to show that eventually the value of $\hat{R}$ stabilizes around unity.

16.5.3 The Raftery–Lewis Test

An ideal test for the convergence of MCMCs is one that determines the length of the burn-in period, and how long should the chain be run to achieve a given precision in

the estimate of the model parameters. The Raftery–Lewis test provides estimates of both quantities, based on just a short test run of the chain. The test was developed by Raftery and Lewis [37], and it uses the comparison of the short sample chain with an uncorrelated chain to make inferences on the convergence properties of the chain. In this section we describe the application of the test and refer the reader interested in its justification to [37].

The starting point to use the Raftery–Lewis test is to determine what inferences we want to make from the Markov chain. Typically we want to estimate confidence intervals at a given significance for each parameter, which means we need to estimate two values θ_{min} and θ_{max} for each parameter θ such that their interval contains a probability $1-q$ (e.g., respectively, $q = 0.32, 0.10$ or 0.01 for confidence level 68, 90 or 99 %),

$$1 - q = \int_{\theta_{min}}^{\theta_{max}} \pi(\theta) d\theta.$$

Consider, for example, the case of a 90 % confidence interval: the two parameter values θ_{min} and θ_{max} are respectively the $q = 0.95$ and the $q = 0.05$ quantiles, so that the interval $(\theta_{min}, \theta_{max})$ will contain 90 % of the posterior probability for that parameter.

One can think of each quantile as a statistic, meaning that we can only approximately estimate their values $\hat{\theta}_{min}$ and $\hat{\theta}_{max}$. The Raftery–Lewis test lets us estimate any quantile $\hat{\theta}_q$ such that it satisfies $P(\theta \leq \hat{\theta}_q) = 1 - q$ to within $\pm r$, with probability s (say 95 % probability, $s = 0.95$). We have therefore introduced two additional probabilities, r and s, which should not be confused with the quantile q. Consider, for example, that the requirement is to estimate the $q = 0.05$ quantile, with a precision of $r = 0.01$ and a probability of achieving this precision of $s = 0.95$. This corresponds to accepting that the 90 % confidence interval resulting from such estimate of the $q = 0.05$ quantile (and of the $q = 0.95$ as well) may in reality be a 88 % or a 92 % confidence interval, 95 % of the time.

The Raftery–Lewis test uses the information provided by the sample chain, together with the desired quantile q and the tolerances r and s, and returns the number of burn-in iterations, and the required number of iterations N. A justification for this test can be found in [37], and the test can be simply run using widely available software such as the *gibbsit* code or the the *CODA* software [28, 34]. Note that the required number of iterations are a function of the quantile to be estimated, with estimation of smaller quantiles typically requiring longer iterations.

Summary of Key Concepts for this Chapter

☐ *Monte Carlo Markov chain (MCMC)*: A numerical method to implement a Markov chain, with the goal of estimating the posterior distribution of model parameters via

$$P(\theta/Z) \propto P(\theta)\mathcal{L}.$$

☐ *Metropolis–Hastings algorithm*: A commonly used method to draw and accept or reject candidates for the MCMC. It is based on an acceptance probability that simplifies to a ratio of likelihoods,

$$\alpha(\theta^{'}/\theta_n) = \min\left\{\frac{\mathcal{L}(\theta^{'})}{\mathcal{L}(\theta_n)}, 1\right\}$$

when the priors and proposal distributions are uniform.

☐ *The Gibbs Sampler*: An alternative algorithm to create an MCMC that makes use of the full conditional distribution of each parameter.

☐ *Convergence tests*: Tests to ensure that the MCMC is sampling the intended posterior distribution. They typically require to excise a *burn-in* time when the MCMC has not yet reached the stationary distribution.

☐ *Geweke z-score test*: A simple test of convergence that makes use of z-scores of two segments of the chain.

☐ *Gelman–Rubin test*: A convergence test that requires multiple parallel chains and makes use of *between-chain* and *within-chain* variances.

☐ *Raftery–Lewis test*: A convergence test that compares a sample of the MCMC to an uncorrelated chain to determine burn-in time and required length of the chain.

Problems

16.1 Prove that, in the presence of positive correlation among MCMC samples, the variance of the sample mean is larger than that of an independent chain.

16.2 Using the data of $\log m$ and velocity from Table 8.1 of Hubble's experiment, construct a Monte Carlo Markov chain for the fit to a linear model with 10,000 iterations. Use uniform distributions for the prior and proposal distributions of the two model parameters a and b, the latter with widths of 0.1 and 0.02, respectively, for a and b in the neighborhood of the current value. You can start your chain at values of $a = 0.2$ and $b = 0.9$. After completion of the chain, plot the sample distribution of the two model parameters.

16.3 A one-parameter chain is constructed such that in two intervals A and B the following values are accepted into the chain:

$$A : 10, 11, 13, 11, 10$$
$$B : 7, 8, 1, 11, 10, 8;$$

where A is an initial interval, and B an interval at the end of the chain. Not knowing how the chain was constructed, use the Geweke z score to determine whether the chain *might* have converged.

16.4 Using the data of Table 10.1, construct a Monte Carlo Markov chain for the parameters of the linear model, with 10,000 iterations. Use uniform distributions for the prior and proposal distributions, the latter with a width of 10 for both parameters. Start the chain at $a = 12$ and $b = 6$. After completion of the chain, plot the sample distribution of the two model parameters.

16.5 Consider the following portions of two one-parameter chains, run in parallel and starting from different initial positions:

$$7, 8, 1, 11, 10, 8$$
$$11, 11, 8, 10, 9, 12.$$

Using two segments of length $b = 3$, calculate the Gelman–Rubin statistic $\sqrt{\hat{R}}$ for both segments under the hypothesis of uncorrelated samples.

16.6 Consider the step-function model described in Example 16.2, and a dataset consisting of n measurements. Assuming that the priors on the parameters λ, μ and m are uniform, show that the full conditional distributions are given by

$$\begin{cases} \pi_\lambda(\lambda) = G\left(\sum_{i=1}^{m} y_i + 1, m\right) \\ \pi_\mu(\mu) = G\left(\sum_{i=m+1}^{n} y_i + 1, n - m\right) \\ \pi_m(m) = \dfrac{e^{-m\lambda}\lambda^{\sum_{i=1}^{m} y_i} e^{-(n-m)\mu}\mu^{\sum_{i=m+1}^{n} y_i}}{\sum_{l=1}^{n} e^{-l\lambda}\lambda^{\sum_{i=1}^{l} y_i} e^{-(n-l)\mu}\mu^{\sum_{i=l+1}^{n} y_i}}, \end{cases} \tag{16.24}$$

where G represents the gamma distribution.

16.7 Consider the step-function model described in Example 16.2, and a dataset consisting of the following five measurements:

$$0, 1, 3, 4, 2.$$

Start a Metropolis–Hastings MCMC at $\lambda = 0$, $\mu = 2$ and $m = 1$, and use uniform priors on all three parameters. Assume for simplicity that all parameters can only

be integer, and use uniform proposal distributions that span the ranges $\Delta\lambda = \pm 2$, $\Delta\mu = \pm 2$ and $\Delta m = \pm 2$, and that the following numbers are drawn in the first three iterations:

Iteration	$\Delta\lambda$	$\Delta\mu$	Δm	α
1	$+1$	-1	$+1$	0.5
2	$+1$	$+2$	$+1$	0.7
3	-1	-2	$+1$	0.1

With this information, calculate the first four links of the Metropolis–Hastings MCMC.

16.8 Consider a Monte Carlo Markov chain constructed with a Metropolis–Hastings algorithm, using uniform prior and proposal distribution. At a given iteration, the chain is at the point of maximum likelihood or, equivalently, minimum χ^2. Calculate the probability of acceptance of a candidate that has, respectively, $\Delta\chi^2 = 1$, 2, and 10.