

## Chapter 9 Output Analysis

---

### 9.1 Introduction

Let  $X_0 = x^{(0)}, X_1 = x^{(1)} \dots X_N = x^{(N)}$  be a realisation of a homogeneous and reversible Markov chain, output by some MCMC algorithm which was set up to sample a distribution  $Q(x)$  on states  $x \in \Omega$ . For example  $\{x^{(n)}\}_{n=0}^N$  might be output from one of the following MCMC algorithm's:

1. the algorithm sampling the posterior for the matrix reconstruction problem from row and column sums,
2. the algorithm sampling the normal distribution.

Suppose  $X \sim Q(\cdot)$  is a random variable taking values in  $\Omega$ . A function  $f(X)$  is called a statistic. We look to the output for answers to questions like “What is the expected value of  $f(X)$ ” — i.e., give a value for

$$\langle f(X) \rangle \equiv \sum_{x \in \Omega} f(x) \Pr(x).$$

We can estimate this from the output sample set  $\{x^{(n)}\}_{n=0}^N$ . The quantity

$$\bar{f}_N(\{x^{(n)}\}_{n=0}^N) \equiv \frac{1}{N} \sum_{n=1}^N f(x^{(n)})$$

is called an estimator of  $\langle f(X) \rangle$ . Since the samples  $\{x^{(n)}\}_{n=0}^N$  are realisations of random variables  $\{X_n\}_{n=0}^N$ , our estimate,  $\bar{f}_N(\{X_n\}_{n=0}^N)$ , is itself a random variable. We will get a different estimate value at each “trial”, that is, each time we gather a realisation  $\{x^{(n)}\}_{n=0}^N$  of the r.v.  $\{X_n\}_{n=0}^N$ .

**Example 1** If the  $\{x^{(n)}\}_{n=0}^N$  were sampled independently from a normal distribution mean  $\mu$ , variance  $\sigma^2$ ,  $f(x) = x$  and  $\bar{x} = \sum_n x^{(n)}/N$  is an estimator for  $\mu$ . In this case  $\bar{x}$  is a normally distributed random variable with mean  $\mu$  and variance  $\sigma^2/N$ .

**Example 2** Using the output in case (2) above, find a statistic giving the probability for the variable at pixel  $(j, k)$  to take the value  $x_{jk} = -1$ .

**Answer** If  $\mathcal{A}$  and not- $\mathcal{A}$  are two possible outcomes of a trial then

$$\Pr(\text{event } \mathcal{A}) \simeq \frac{\text{\#times } \mathcal{A} \text{ occurred}}{\text{number of trials}}$$

estimates the probability for the event  $\mathcal{A}$ . Let  $f(x) = I[x_{jk} = -1]$ , i.e.,  $f(x)$  is the indicator function,  $I[\text{assertion}] = 0/1$  if **assertion** is true/false. Now

$$\bar{f}_N = \frac{1}{N} \sum_{n=1}^N I(x_{jk}^{(n)} = -1) \tag{9.1}$$

$$= \frac{\text{\#times } x_{jk} = -1 \text{ in the sampled states } \{x^{(n)}\}_{n=0}^N}{\text{number of samples}} \tag{9.2}$$

so our statistic  $f = I[x_{jk} = -1]$  gives an estimator  $\bar{f}_N$  for  $\Pr(x_{jk} = -1)$

How good an estimator for  $\langle f(X) \rangle$  is  $\bar{f}_N$ ? There are two issues:

1. Systematic error due to initialization bias — i.e., does  $\bar{f}_N$  estimate  $\langle f(X) \rangle$  at all? We may prove that  $\pi^{(n)}$  tends to some desired unique equilibrium distribution  $Q(x)$ , however we have not said how large  $n$  needs to be for  $\pi^{(n)} \approx Q(x)$ . Multi-modality in  $Q$  and bugs in our computer program can be a problem here.
2. Autocorrelation in equilibrium — i.e., given  $\pi^{(n)} = Q(x)$ , for some  $n$ , how accurate is  $\bar{f}_N$  as an estimator for  $\langle f(X) \rangle$ ? Samples  $x^{(n)}, x^{(n+1)}$  are highly correlated. How many samples should we take to achieve a given level of accuracy?

## 9.2 Autocorrelation in equilibrium

Recall that the variance of a quantity is a measure of the variability of the quantity about its mean. Thus if  $x \sim Q(x)$ , and  $\mu_f \equiv \langle f(X) \rangle$ ,

$$\text{var}(f) = \langle f(X)^2 - \mu_f^2 \rangle$$

measures the variance of  $f$  in samples  $x$  distributed like  $Q(x)$ . Consider the quantity  $\bar{f}_N$ .  $\text{var}(\bar{f}_N)$  is a measure of the variability of our estimator for  $\mu_f$ .

We can quantify this statement. It is known that, under quite general conditions which we will not detail (see e.g. Geyer), for  $N$  sufficiently large,

$$\bar{f}_N \sim \text{Normal}(\langle f \rangle, \text{var}(\bar{f}_N)) \quad (9.3)$$

i.e., when  $N$  is large, our estimate  $\bar{f}_N$  is normally distributed with mean  $\langle f \rangle$  and some variance  $\text{var}(\bar{f}_N)$ . Equation 9.3 is usually written as a theorem concerning the limit

$$\lim_{N \rightarrow \infty} \sqrt{N}(\bar{f}_N - \langle f \rangle) \xrightarrow{\mathcal{D}} \text{Normal}(0, c) \quad (9.4)$$

with  $c$  some positive constant, independent of  $N$ . The limit “ $\mathcal{D}$ ” means that the distribution of the random variable on the left tends to the distribution on the right, i.e., normal, mean zero. Equation 9.4 is a statement of a “central limit theorem” concerning the distribution of  $\bar{f}_N$ .

If  $\{x^{(n)}\}_{n=0}^N$  were *independent* samples and  $\bar{f}_N \equiv \frac{1}{N} \sum_{n=1}^N f(x^{(n)})$  then

$$\text{var}(\bar{f}_N) = \text{var}(f)/N \quad (9.5)$$

However  $\{x^{(n)}\}_{n=0}^N$  is a sequence of *correlated* samples from a Markov chain. We will see that

$$\text{var}(\bar{f}_N) = \frac{\tau_f \text{var}(f)}{N} \quad (9.6)$$

where  $\tau_f$  is a number characteristic of the transition matrix of the Markov chain used to generate the sequence  $\{x^{(n)}\}_{n=0}^N$ . It follows from Equation 9.6 that  $c = \tau_f \text{var}(f)$  independent of  $N$ , in Equation 9.4.

**Interpretation:** In Equation 9.5 the variance of  $\bar{f}_N$  goes down like  $1/N$ , where  $N$  is the number of independent samples. In Equation 9.6 the variance of  $\bar{f}_N$  goes down like  $\tau_f/N$ . Hence  $\tau_f$  is the number of correlated samples with the same variance-reducing power as one independent sample. The quantity  $\tau_f$  is called the **integrated autocorrelation time** (IACT, in physics literature) or **autocovariance time** (statistics literature).

Let  $\sigma_{\bar{f}_N} \equiv \sqrt{\text{var}(\bar{f}_N)}$  denote the standard deviation of our estimate  $\bar{f}_N$ . We are assuming  $N$  is large enough that the CLT we quoted has set in and  $\bar{f}_N$  is normally distributed. We use all of this information when we put error-bars on  $\bar{f}_N$ : we report something like “we measured  $f$  and obtained a mean of  $\bar{f}_N \pm 2\sigma_{\bar{f}_N}$  at 95% confidence”.

For a given equilibrium distribution  $Q(x)$ , we would like to design a chain for which  $\tau_f$  is as small as possible, so that we get accurate estimates (small  $\sigma_{\bar{f}_N}$ ) without needing large sample sizes  $N$ .

### 9.3 Calculating the integrated autocorrelation time

The covariance  $\text{cov}(f, g)$  of two quantities  $f$  and  $g$  is a measure of their correlation. Let  $\mathcal{M} = \{X_n\}_{n=0}^{N-1}$  be a sequence of  $N$  stationary random variables in some given homogeneous Markov chain with equilibrium distribution  $Q$ , i.e.,  $X_0 \sim Q$ , so that all the r.v. in the sequence are distributed according to the equilibrium distribution. In terms of some statistic  $f(X)$ , let

$$C_{ff}(s) \equiv \text{cov}(f(X_n), f(X_{n+s})) \quad (9.7)$$

$$\equiv \langle f(X_n)f(X_{n+s}) \rangle - \mu_f^2 \quad (9.8)$$

be the autocovariance function (ACF) at lag  $s$ , i.e.,  $C_{ff}(s)$  is the covariance between the values taken by  $f$  for two r.v.  $X_n$  and  $X_{n+s}$  in the chain separated by  $s$  updates. Since the chain is homogeneous and its distribution is stationary,  $C_{ff}(s)$  depends only on  $s$  and not on  $n$ . For a Markov chain, this gets larger the closer the two states are together. If we define a normalised autocovariance function,  $\rho_{ff}(s)$  via

$$\begin{aligned} \rho_{ff}(s) &= C_{ff}(s)/C_{ff}(0) \\ &= C_{ff}(s)/\text{var}(f) \end{aligned}$$

then  $\rho_{ff}(0) = 1$  — so  $f(X_n)$  is perfectly correlated with itself! We expect  $\rho_{ff}(s) \rightarrow 0$  monotonically as  $s \rightarrow \infty$  (see Figure 9.1). Note that some authors refer to  $C_{ff}$  as the “autocorrelation function” although we reserve this usage for the case where the means have not been subtracted. We will

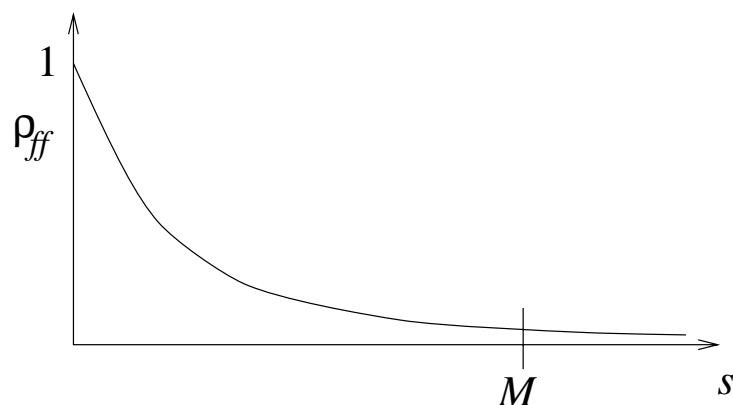


Figure 9.1 The normalised autocovariance function  $\rho_{ff}(s)$ .

assume that for some  $M$  sufficiently large  $\rho_{ff}(s) \simeq 0$  when  $s \geq M$ . We will assume also that

$N \gg M$ , so that the first  $x^{(0)}$  and last  $x^{(N)}$  samples are totally uncorrelated. We will show that under those assumptions  $\text{var}(\bar{f}_N) = \tau_f \text{var}(f)/N$ , as asserted in Equation 9.6.

$$\text{var}(\bar{f}_N) = \langle \bar{f}_N^2 \rangle - \langle \bar{f}_N \rangle^2 \quad (9.9)$$

$$= \left\langle \left( \frac{1}{N} \sum_{n=1}^N f(X_n) \right) \left( \frac{1}{N} \sum_{m=1}^N f(X_m) \right) \right\rangle - \left\langle \frac{1}{N} \sum_{n=1}^N f(X_n) \right\rangle^2 \quad (9.10)$$

$$= \frac{1}{N^2} \sum_{n=1}^N \sum_{m=1}^N \langle f(X_m) f(X_n) \rangle - \langle f^2 \rangle \quad (9.11)$$

since  $\langle f(X_n) \rangle = \langle f \rangle$ , independent of  $n$ , if the chain is homogeneous and stationary. Now, if  $M \ll N$ ,

$$\sum_{n=1}^N \sum_{m=1}^N \langle f(X_m) f(X_n) \rangle \approx \sum_{n=1}^N \left[ \langle f(X_n) f(X_n) \rangle + 2 \sum_{s=1}^{N-M} \langle f(X_n) f(x^{(n+s)}) \rangle \right] \quad (9.12)$$

$$= \sum_{n=1}^N \left[ C_{ff}(0) + \langle f \rangle^2 + 2 \sum_{s=1}^{N-M} C_{ff}(s) + \langle f \rangle^2 \right] \quad (9.13)$$

and so,

$$\text{var}(\bar{f}_N) = \frac{1}{N^2} \sum_{n=1}^N \text{var}(f) + 2 \sum_{s=1}^{N-M} C_{ff}(s) \quad (9.14)$$

$$\simeq \frac{\text{var}(f)}{N^2} \sum_{n=1}^N \left[ 1 + 2 \sum_{s=1}^M \rho_{ff}(s) \right] \quad (9.15)$$

$$\simeq \frac{\text{var}(f)\tau_f}{N} \quad (9.16)$$

as asserted in the previous section, with

$$\tau_f \equiv 1 + 2 \sum_{s=1}^{\infty} \rho_{ff}(s).$$

$\tau_f$  is the integral of the normalized autocovariance function  $\rho_{ff}(s)$ , because  $\rho_{ff}(s) = \rho_{ff}(-s)$ , and  $\rho_{ff}(0) = 1$ . To estimate  $\tau_f$  from the output  $\{x^{(n)}\}_{n=1}^N$  of an MCMC algorithm, we estimate  $C_{ff}(s)$  by

$$\bar{C}_{ff}(s) = \frac{1}{N} \sum_{n=1}^N f(x^{(n)}) f(x^{(n+s)}) - \frac{1}{N^2} \left[ \sum_{n=1}^N f(x^{(n)}) \right]^2$$

and then compute an estimate  $\bar{\rho}_{ff}(s)$  for  $\rho_{ff}(s)$  as above. A problem remains.

Since  $N$  is finite, the estimates  $\bar{\rho}_{ff}(s)$  are noisy. We expect  $\rho_{ff}(s) \simeq 0$  when  $s \geq M$ , so the signal goes to zero, and at large  $s$ ,  $s \geq M$  say,  $\bar{\rho}_{ff}(s)$  is pure noise. If we were to form our estimate  $\bar{\tau}_f$  for  $\tau_f$  by summing over all  $s$ , including  $s \geq M$ , we would be adding noise and no signal to our estimate for  $\tau_f$ . We must truncate the sum over  $\bar{\rho}_{ff}(s)$  at  $s = M$ . We must decide from the data at what lag  $s$  noise begins to dominate. There are several approaches. The method outlined in Geyer is practical, as well as being theoretically well motivated. Well tested software exists to estimate all the quantities defined in this section. See for example code available through the R package.

**Example 3** We will illustrate output analysis on the output of the MCMC algorithm given on page 7-23, for sampling the normal distribution with a mean  $\mu = 3$  and standard deviation  $\sigma = 1$ .

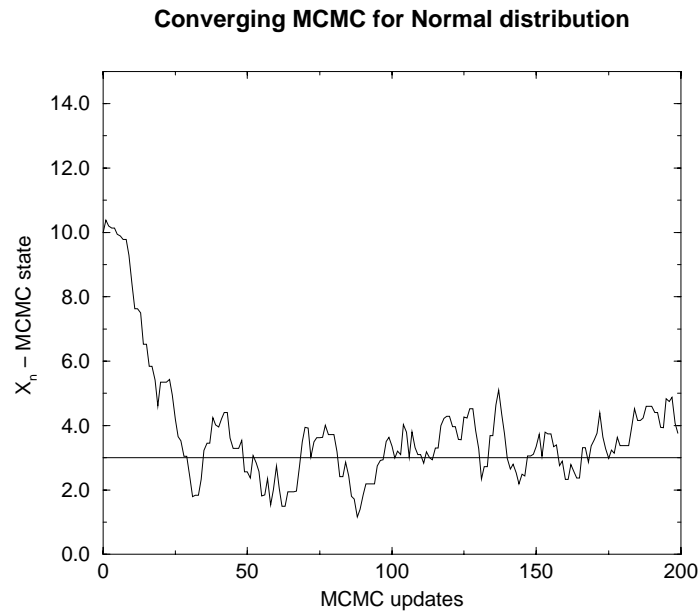


Figure 9.2 The converging sequence of MCMC updates.

A realisation of length 10000 updates was obtained. The first 200 updates (see Figure 9.2) were discarded, to allow time for  $\pi^{(n)}$ , the distribution of the Markov chain r.v. at step  $n$ , to converge to its equilibrium distribution, which, in this case, is the normal distribution. The full output is graphed in Figure 9.3. So we have a sample sequence  $\{x^{(n)}\}_{n=200}^{n=10000}$  of length  $N = 9800$ . The mean of the  $x^{(n)}$  was  $\bar{x} \simeq 3.026$  (i.e.,  $f(x) = x$  above so that  $\mu = \langle f \rangle$  and  $\bar{x} \equiv \bar{f}_N$ ). The variance (i.e.,  $\sigma^2 = \text{var}(x)$  above) was estimated at  $\bar{\sigma}^2 \simeq 0.97$ . In order to determine the variance of  $\bar{x}$  the normalised ACF  $\bar{\rho}_{xx}(s)$  was estimated from the same output data, as above. This estimated ACF is plotted in Figure 9.4.  $\tau_x$  was calculated using the estimator,

$$\bar{\tau}_x = 1 + 2 \sum_{s=1}^M \bar{\rho}_{xx}(s)$$

with  $M = 23$  (see Figure 9.4). This gave an estimate of  $\bar{\tau} \simeq 14.5$  for  $\tau$ . This means that it takes around fifteen updates of the MCMC to give one effective independent sample. Now we estimate

$$\text{var}(\bar{x}) = \frac{\bar{\sigma}^2 \bar{\tau}_x}{N} \simeq 0.0014$$

So we might reasonably report

$$\bar{x} = 3.03 \pm 0.08$$

at 95% confidence (where  $0.08 \simeq 2\sqrt{0.0014}$ ).

**Exercise 9.3.1** How could we check the result for  $\text{var}(\bar{x})$ , using 10 independent runs of length 10000 ?

**Answer** Taking the output from each of the 10 runs we form 10 independent estimates  $\bar{x}_1, \bar{x}_2 \dots \bar{x}_{10}$ . According to the CLT we quoted in Equation 9.4 these estimates should be scattered about  $\mu$  with a normal distribution with variance equal around about  $\text{var}(\bar{x}) = 0.0014$ .

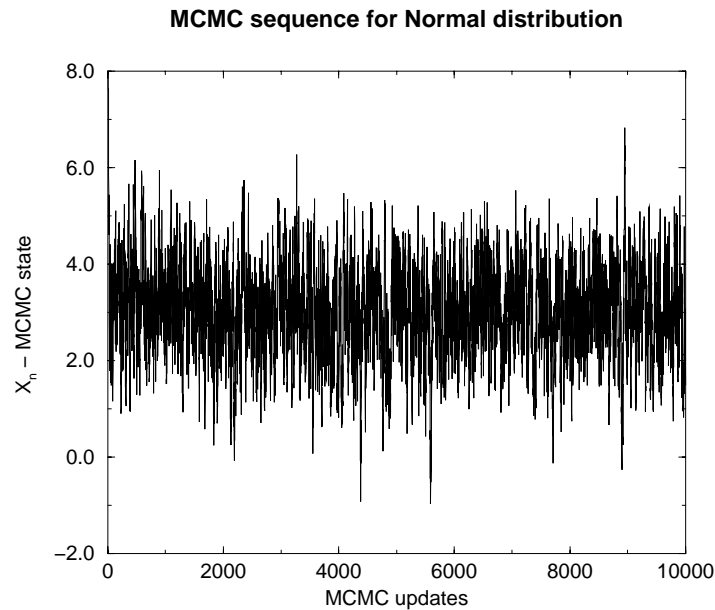


Figure 9.3 The sequence of 10000 MCMC updates output.

Note: If its so easy to estimate the variance of  $\bar{x}$  by repeating the run, why bother with the calculation of  $\tau_f$ ? Because it is only by using all your data you get all the accuracy available. Also, plotting  $\bar{\rho}_f(s)$  is a sensible way to check for convergence.

In practice we are interested in getting  $\text{var}(\bar{f}_N)$  down as small as possible (i.e., getting accurate estimates). We can do this by taking more samples. We can also look at the efficiency of our algorithm. We are free to choose the candidate generation probability  $g(x'|x)$ , and this has a big effect on the rate at which correlations die out along the chain. Good algorithms have smaller IACT  $\tau_f$ , since this leads to smaller values of  $\text{var}(\bar{f}_N)$  for given sample size  $N$ . Now  $\tau_f$  is measured in MCMC updates. An update takes a certain amount of CPU time. So really we want  $\tau_f$  small in CPU seconds. If some fancy algorithm had a small  $\tau_f$ -value, but each update took alot of time to compute, the accuracy of estimates would still only improve slowly. This is why simple algorithms (i.e., simple choices for the candidate generation probability  $g(x'|x)$ ) are hard to beat.

**Example 4** In the MCMC algorithm sampling the normal distribution, we generated candidate states using  $G(dx'|x) = dx'/2\sigma$ , i.e., the new state was chosen uniformly on the interval  $[x-a, x+a]$  with  $a = \sigma$ . Can we reduce  $\tau_f$  by using some other value of  $a$ ?

The core loop of our sampler (page 7-23) is essentially unchanged

```

.
.
a=input('enter jumpsize ');           %enter 'a' used in generation step
for k=1:N
    xp=x+(2*a*rand(1,1)-a);           %generate candidate with jumpsize 'a'
    ratio=exp(-((xp-mu)^2-(x-mu)^2)/(2*sigma^2));
    if rand(1,1)<ratio
        x=xp;
    end;

```

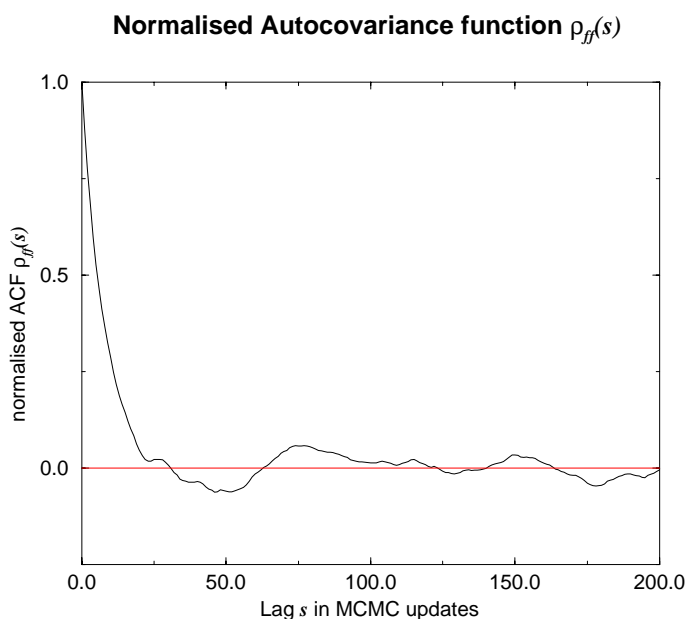


Figure 9.4 The normalised autocovariance function of the MCMC.

```

Xn(k)=x;
end;
.
.

```

This MCMC algorithm simulates a Markov chain with the normal distribution as its equilibrium distribution for  $a$  any non-zero real number. Suppose the mean  $\mu = 3$  and standard deviation  $\sigma = 1$  as before. For each of a sequence of  $a$ -values we run the MCMC and analyse the output, computing  $\tau_x(a)$ . In Figure 9.5,  $\tau_x$  is plotted against  $a$ . The IACT is minimised by a value of jumpsize of around  $a = 3.5$ . This tells us that the chain outputs the greatest number of effectively independent samples per  $N$  updates when the jumpsize is around 3.5.

**Interpretation** Referring to Figure 9.6, when  $a \ll \sigma$ , the candidate state  $x' \simeq x$ , and in the acceptance ratio  $\exp(-(x' - \mu)^2/2\sigma^2 + (x - \mu)^2/2\sigma^2) \simeq 1$ , and hence  $\alpha \simeq 1$ . Most updates are accepted, but since the jump from  $x$  to  $x'$  is small the chain moves slowly through  $\Omega$  and the r.v.  $X_n$  and  $X_{n+1}$  are highly correlated. On the other hand when  $a \gg \sigma$ ,  $x'$  will often be selected well outside the interval  $[\mu - 3\sigma, \mu + 3\sigma]$  where all the probability mass is concentrated. The ratio of the new and old probability densities will be small, and the acceptance probability  $\alpha$  will be small. Although the chain makes large hops when it does move, there are too many rejections:  $X_n$  and  $X_{n+1}$  are highly correlated, because they are often equal !

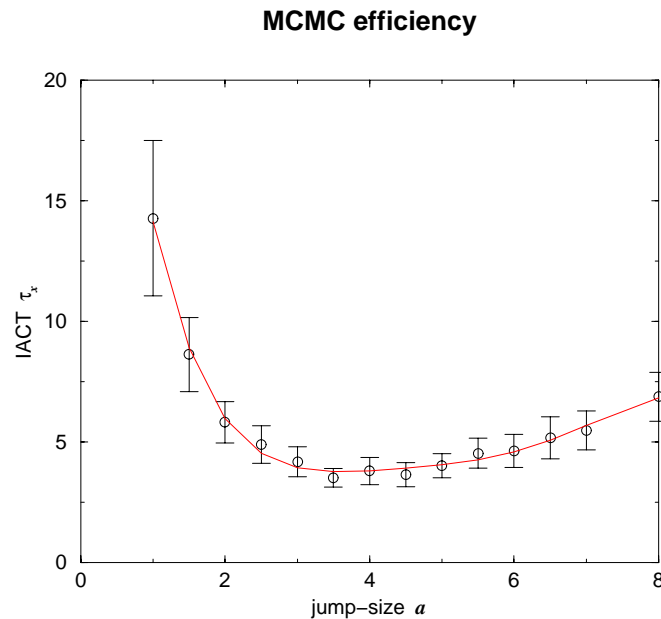


Figure 9.5 The normalised autocovariance function  $\rho_{ff}(s)$ .

## 9.4 Initialisation bias

Suppose we have an MCMC algorithm with intended equilibrium distribution  $Q(x)$ . We have seen (Figure 9.2) that the initial samples generated by MCMC are not representative of the equilibrium distribution. How long should we simulate before beginning to take samples? There is no general solution to the problem. A novel technique called “Exact simulation” offers a complete solution for certain MCMC algorithms and certain equilibrium distributions (Propp and Wilson 1995), including several of importance in statistical mechanics.

The usual approach is to monitor the behavior of some statistic (for example,  $q(x^{(n)})$ , where  $Q(x) = \exp(-q(x))/\mathcal{Z}$ ) and drop samples from the part of the run where  $q(x^{(n)})$  is converging to its equilibrium range. Referring to the MCMC output in Figure 9.7, our output sample would consist of  $x^{(n)}$  for  $n \geq k$  only. We are assuming that, for  $n \geq k$   $X_n \sim Q$ , i.e.,  $\pi^{(n)} = Q$  for  $n \geq k$ . If the total run length  $N \gg k$  this has a negligible effect on estimates, i.e.,

$$\bar{f}_{N-k} = \frac{1}{N-k} \sum_{n=k}^N f(x^{(n)})$$

when the first  $k$  samples are dropped.

## 9.5 Sticking and multimodality

We do not know when the chain has reached equilibrium. Considering Figure 9.8 it is clear that any judgement based on output may be premature. It may be that the generation probability  $g(x'|x)$  can take the chain to any state in the state space  $\Omega$ , but sections of  $\Omega$  communicate only through

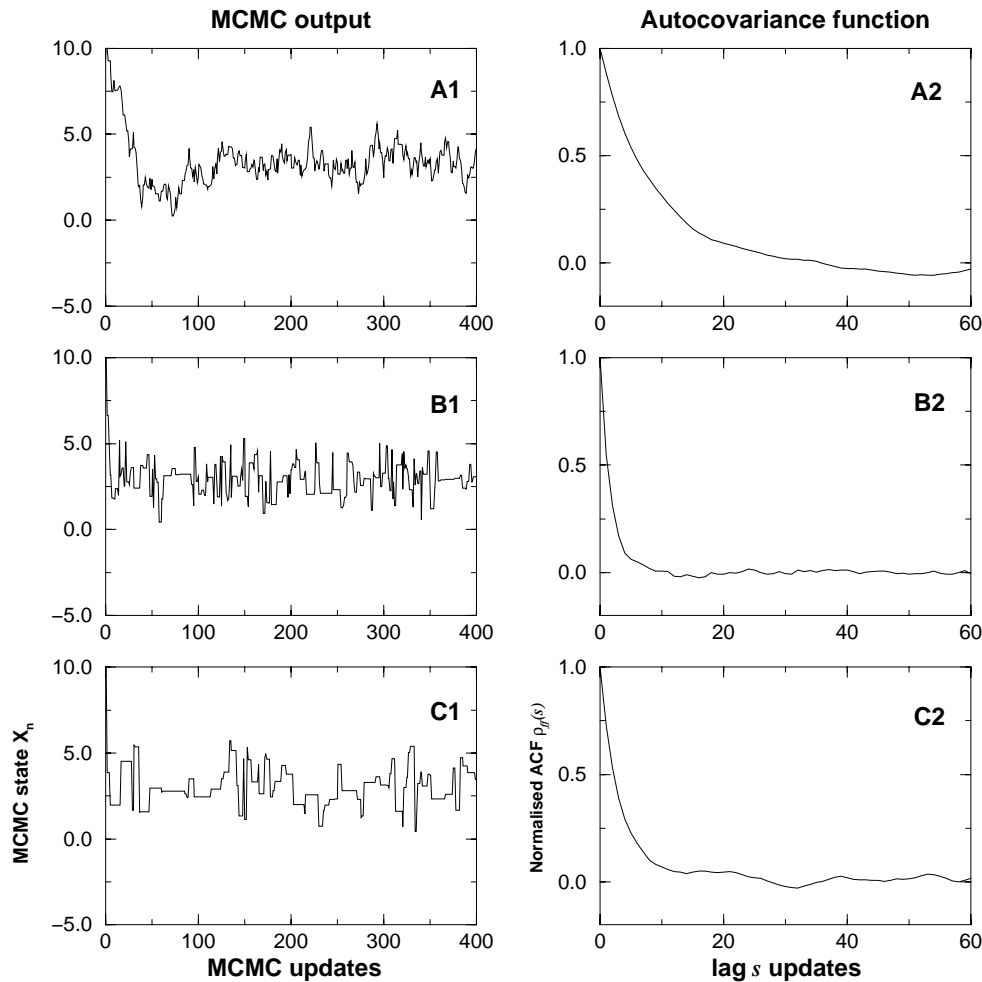


Figure 9.6 (A1) MCMC output for  $a = 1$  (A2) normalised autocovariance function  $\rho_{xx}(s)$  for  $a = 1$ . Although there is a high probability of acceptance, the sample path shows strong correlation from one update to the next. (B1) MCMC output for  $a = 3.5$  (B2)  $\rho_{xx}(s)$  for  $a = 3.5$ . The normalized autocovariance  $\rho_{xx}(s)$  between  $X_n$  and  $X_{n+s}$  now falls off rapidly with increasing lag  $s$ . (C1) output for  $a = 8$  The stepped appearance of the output reflects the low acceptance rate (C2)  $\rho_{xx}(s)$  for  $a = 8$ , the correlation between states in the chain dies off slowly.

states of very low acceptance probability. The chain may be stuck in the vicinity of one mode if  $Q(x)$  is a multimodal distribution.

**Example 5** Suppose  $Q(X \in dx)$  is the sum of two normal distributions with means  $\mu_1$  and  $\mu_2$  ( $\mu_2 > \mu_1$ ) and each having the same standard deviation  $\sigma$ . So, if  $Q(X \in dx) = q(x)dx$ , the density is

$$q(x) = \frac{1}{2\sqrt{2\pi\sigma^2}} \left( e^{-(x-\mu_1)/2\sigma^2} + e^{-(x-\mu_2)/2\sigma^2} \right).$$

We will suppose that the two normal distributions are “separated” in the sense that  $\mu_2 - \mu_1 \gg \sigma$ . See Figure 9.9. Now, for our MCMC algorithm with equilibrium  $Q(x)$  we might choose the following:

Let  $X_n = x$ .  $X_{n+1}$  is determined in the following way.

1. choose  $x'$  uniformly in an interval  $[x - a, x + a]$  centered at  $x$ . Thus  $x' \sim dx'/2a$  and the density  $1/2a$  is (trivially) symmetric in  $x$  and  $x'$ .

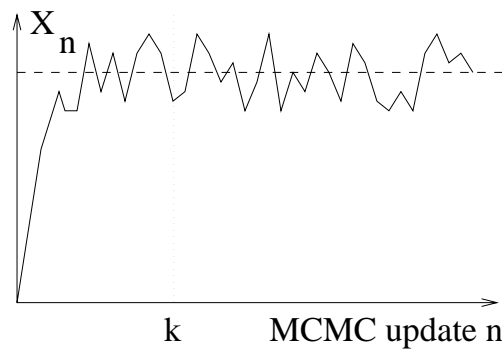


Figure 9.7 A MCMC algorithm started at an unrepresentative state converges to its equilibrium range.

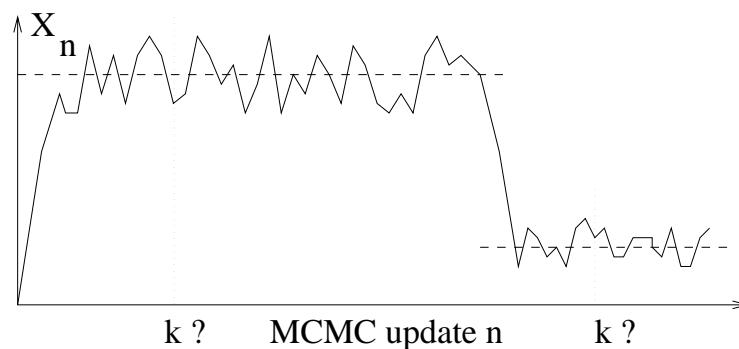


Figure 9.8 The chain may appear to be in equilibrium, when it is in fact in a metastable state.

2. With probability

$$\alpha = \min \left\{ 1, \frac{\exp(-(x' - \mu_1)/2\sigma^2) + \exp(-(x' - \mu_2)/2\sigma^2)}{\exp(-(x - \mu_1)/2\sigma^2) + \exp(-(x - \mu_2)/2\sigma^2)} \right\}$$

set  $X_{n+1} = x'$ . Otherwise we set  $X_{n+1} = x$ .

This algorithm has a straightforward Matlab implementation. The choices  $\mu_1 = -4, \mu_2 = 4$  and  $\sigma = 1$  give a wide region of low probability between lumps. Sample output for these parameter values and a range of jumpsizes  $a$  is shown in Figure 9.10, along with autocovariance functions. Ergodicity is effectively lost for  $a$  less than around 4. The very slowly falling autocovariance function for  $a = 4$ , middle row right, indicates this. However at  $a = 1$  the output is giving the illusion of an ergodicity that is not present - there is no sign of states at values  $x = 4$  even though they hold half the probability mass. The MCMC is stuck in the mode around  $x = -4$ .

## 9.6 Good habits

We have seen that there is no guaranteed adequate output analysis. In particular, practical convergence cannot be demonstrated in general. The following procedure is however often sufficient.

1. Take a single long run, of length  $N$ , dropping states from the start of the run, and estimate  $\tau_f$ , for  $f(X)$  a statistic you care about.

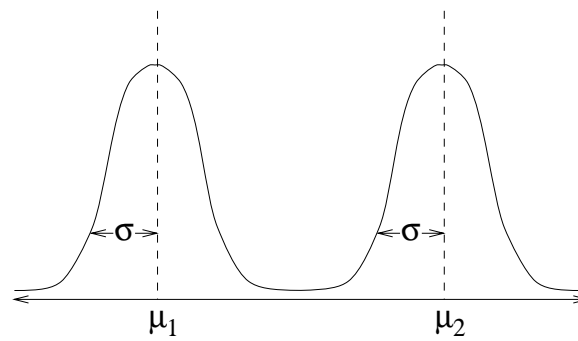


Figure 9.9 A distribution in which two regions of higher probability are separated by a region of low probability.

2. Check that  $N \gg \tau_f$ . This is the only meaningful definition of a “long” run.
3. Plot the output  $x^{(n)}$ , or  $f(x^{(n)})$  against  $n$ . Does it show any obvious trend? Does it seem to be “in equilibrium”?
4. Plot the normalised autocovariance function  $\rho_{ff}(s)$  against lag  $s$ . It should fall off smoothly to zero, and then be distributed more or less evenly, with noise, about the x-axis.
5. Read Geyer “MCMC in practice”, Statistical Science article, for some checks on the asymptotic variance of the autocorrelation function as  $s \rightarrow$  large.

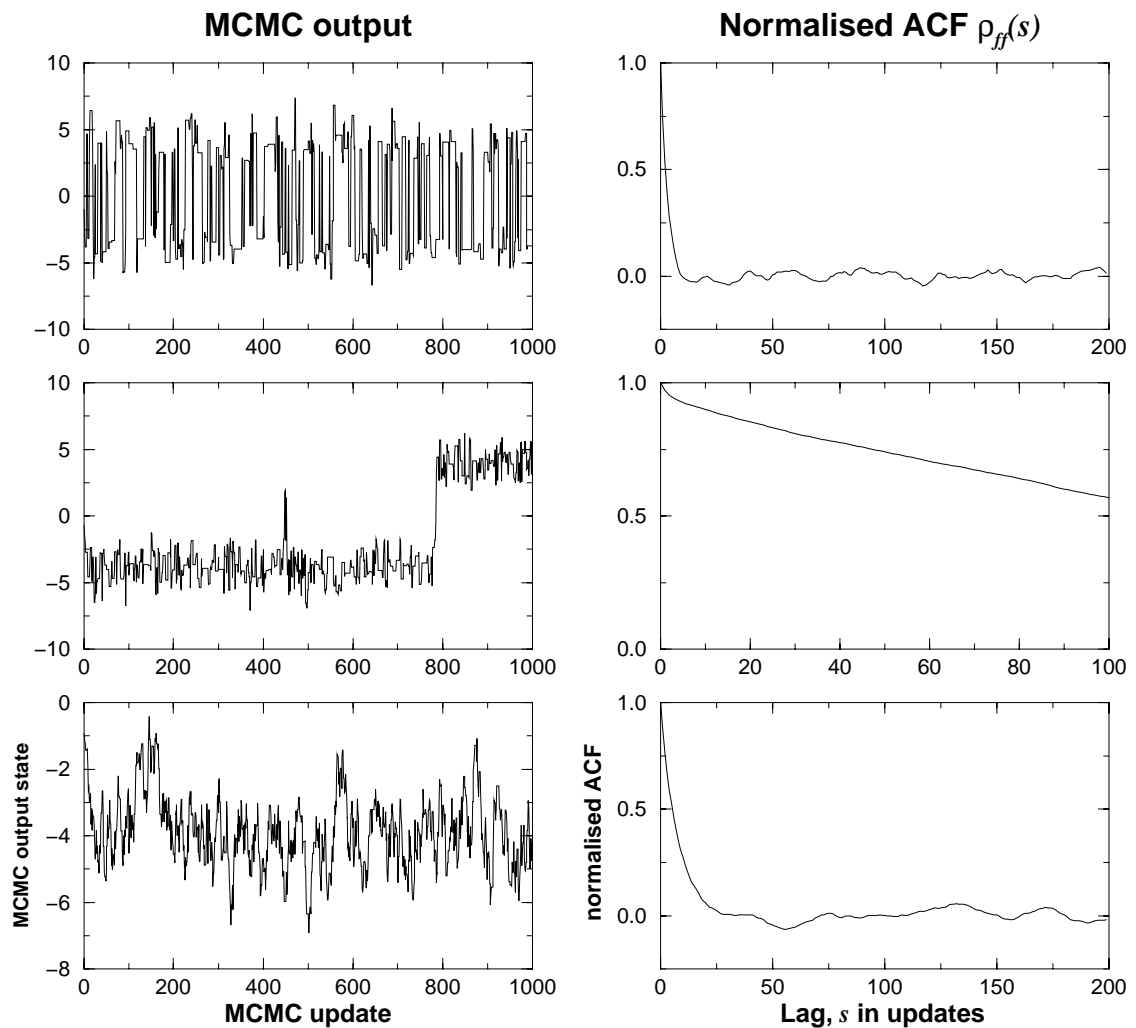


Figure 9.10 Illustration of initialisation bias and sticking: Output for MCMC sampling the multi-modal distribution of Figure 9.9. Top row,  $a = 8$  (ergodic), middle row  $a = 4$  (very slow mixing, strong initialisation bias), bottom row  $a = 1$  (stuck).