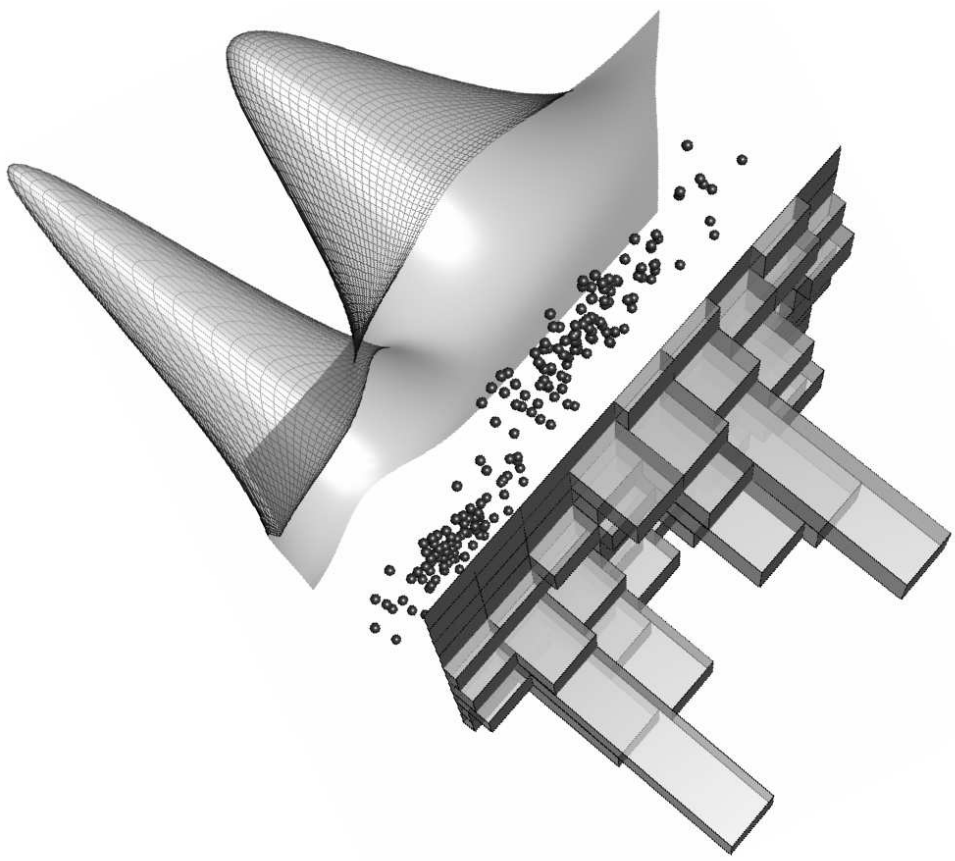
Adam M. Johansen (lectured by Ritabrata Dutta)

ST407 Monte Carlo Methods

Lecture Notes 2020/21

2/10/2020



Warwick Statistics

Contents

Preface

These notes provide some material to support the ST407 Monte Carlo Methods module.

It is not the intention that these notes should serve as either a replacement for the lectures themselves or an exhaustive summary of what is presented in those lectures. They should serve to complement the lectures, providing additional mathematical details in some areas and some material which will not be covered in lectures.

Acknowledgement

These notes have evolved from those used a course written by myself and Dr. Ludger Evers at the University of Bristol in 2007-08. Any corrections or queries should be sent to `Ritabrata.Dutta@warwick.ac.uk`.

1. Introduction

1.1 What are Monte Carlo Methods?

This lecture course is concerned with a class of simulation-based computational techniques which are used widely throughout Statistics, Physics and many other disciplines, which we shall refer to as Monte Carlo methods. These techniques are sometimes referred to as *stochastic simulation* (Ripley (1987) for example uses only this term), a more literal but less common description.

Examples of Monte Carlo methods include:

- **stochastic integration:** the use of a simulation-based method to approximately evaluate an integral
- **Monte Carlo testing:** the use of simulation in order to approximately compute a p-value
- **optimisation:** the use of a simulation-based method to approximate the minimum or maximum of a function

and many others. These random algorithms can be very broadly applicable, but do generally introduce some approximation error.

One formal definition of Monte Carlo methods was given by Halton (1970) which defined a Monte Carlo method as "representing the solution of a problem as a parameter of a hypothetical population, and using a random sequence of numbers to construct a sample of the population, from which statistical estimates of the parameter can be obtained."

1.2 Introductory examples

1.2.1 Monte Carlo Integration

Example 1.1 (Computing π in the rain). Assume we want to compute an Monte Carlo estimate of π using a simple experiment.

Assume that we are able to produce "uniform rain" on the square $[-1,1]\times[-1,1]$, such that the probability of a raindrop falling into a region $\mathcal{R} \subset [-1,1]^2$ is proportional to the area of $\mathcal{R}$, but independent of the position of $\mathcal{R}$. It is easy to see that this is the case iff the two coordinates X, Y are i.i.d. realisations of uniform distributions on the interval $[-1,1]$ (in short $X, Y \overset{\text{iid}}{\sim} \mathsf{U}[-1,1]$).

Now consider the probability that a raindrop falls into the unit circle (see Figure 1.1). It is

$$\mathbb{P}(\text{drop within circle}) = \frac{\text{area of the unit circle}}{\text{area of the square}} = \frac{\iint_{\{x^2+y^2\leq 1\}} 1\, dxdy}{\iint_{\{-1\leq x,y\leq 1\}} 1\, dxdy} = \frac{\pi}{2\cdot 2} = \frac{\pi}{4}$$

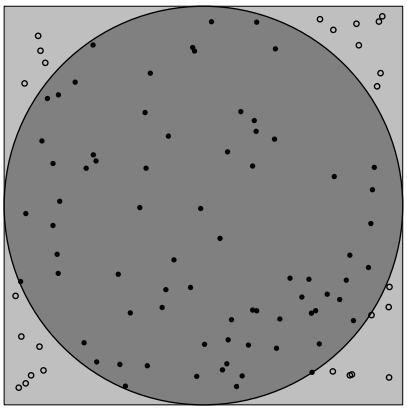


Figure 1.1. Illustration of the estimation π using uniform rain

In other words,

$$\pi = 4 \cdot \mathbb{P}(\text{drop within circle}),$$

i.e. there is an expression for the desired quantity π as a function of a probability.
Of course we cannot compute $\mathbb{P}(\text{drop within circle})$ without knowing π, however we can estimate the probability using our raindrop experiment. If we observe n raindrops, then the number of raindrops Z that fall inside the circle is a binomial random variable:

$$Z \sim \text{Bin}(n, p), \qquad \text{with } p = \mathbb{P}(\text{drop within circle}).$$

Thus we can estimate p by its maximum-likelihood estimate

$$\hat{p} = \frac{Z}{n},$$

and we can estimate π by

$$\hat{\pi} = 4\hat{p} = 4 \cdot \frac{Z}{n}.$$

Assume we have observed, as in Figure 1.1, that 77 of the 100 raindrops were inside the circle. In this case, our estimate of π is

$$\hat{\pi} = \frac{4 \cdot 77}{100} = 3.08,$$

which is quite poor.
However the *strong law of large numbers* guarantees that our estimate $\hat{\pi}$ converges almost surely to π. Figure 1.2 shows the estimate obtained after n iterations as a function of n for $n = 1, \ldots, 2000$. You can see that the estimate improves as n increases.
We can assess the quality of our estimate by computing a confidence interval for π. As we have $X \sim \text{Bin}(100, p)$, we can obtain a 95% confidence interval for p using a Normal approximation:

$$\left[0.77 - 1.96 \cdot \sqrt{\frac{0.77 \cdot (1 - 0.77)}{100}},\ 0.77 + 1.96 \cdot \sqrt{\frac{0.77 \cdot (1 - 0.77)}{100}}\right] = [0.6875,\ 0.8525],$$

As our estimate of π is four times the estimate of p, we now also have a confidence interval for π:

$$[2.750,\ 3.410],$$

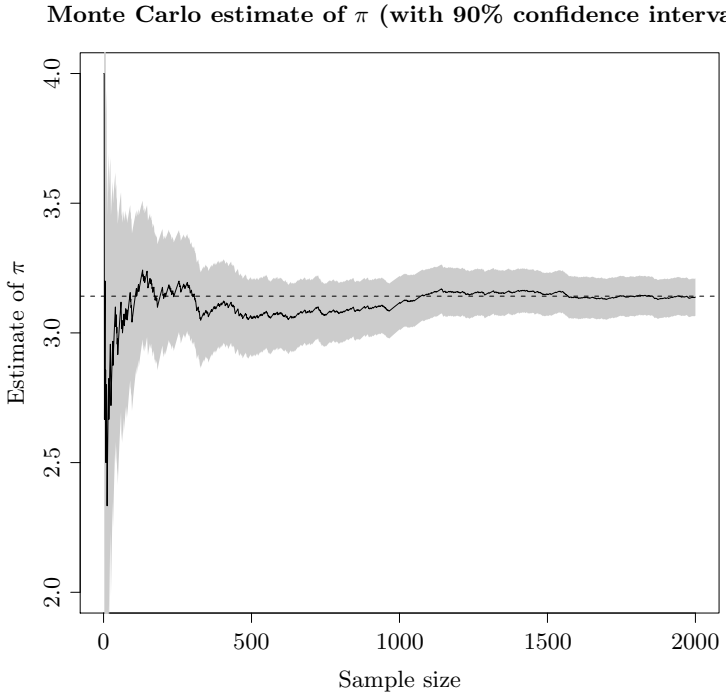


Figure 1.2. Estimate of π resulting from the raindrop experiment

in the sense that if we repeated this entire experiment many times and computed an interval in this way, we would expect it to contain the true value of π approximately 95% of the time. In greater generality, let $\hat{\pi}_n = 4\hat{p}_n$ denote the estimate after having observed n raindrops. A $(1-2\alpha)$ confidence interval for p is then

$$\left[\hat{p}_n - z_{1-\alpha}\sqrt{\frac{\hat{p}_n(1-\hat{p}_n)}{n}}, \hat{p}_n + z_{1-\alpha}\sqrt{\frac{\hat{p}_n(1-\hat{p}_n)}{n}}\right],$$

thus a $(1-2\alpha)$ confidence interval for π is

$$\left[\hat{\pi}_n - z_{1-\alpha}\sqrt{\frac{\hat{\pi}_n(4-\hat{\pi}_n)}{n}}, \hat{\pi}_n + z_{1-\alpha}\sqrt{\frac{\hat{\pi}_n(4-\hat{\pi}_n)}{n}}\right] \qquad \triangleleft$$

Recall the main steps of this process:

- We have written the quantity of interest (in our case π) as an expectation (a probability is a special case of an expectation as $\mathbb{P}(A) = \mathbb{E}(\mathbb{I}_A)$ where $\mathbb{I}_A(x)$ is the *indicator function* which takes value 1 if $x \in A$ and 0 otherwise).
- We have replaced this algebraic representation of the quantity of interest with a sample approximation. The strong law of large numbers guaranteed that the sample approximation converges to the algebraic representation, and thus to the quantity of interest. Furthermore, the central limit theorem allows us to assess the speed of convergence.

It is of course of interest whether the Monte Carlo methods offer more favourable rates of convergence than other numerical methods. We will investigate this in the case of Monte Carlo integration using the following simple example.

Example 1.2 (Monte Carlo Integration). Assume we want to evaluate the integral

$$\int_0^1 f(x)\,dx \qquad \text{with} \qquad f(x) = \frac{1}{27}\cdot\left(-65536x^8 + 262144x^7 - 409600x^6 + 311296x^5 - 114688x^4 + 16384x^3\right)$$

using a Monte Carlo approach. As f is a polynomial we can obtain the result analytically, it is $\frac{4096}{8505} = \frac{2^{12}}{3^5 \cdot 5 \cdot 7} \approx 0.4816$. Figure 1.3 shows the function for $x \in [0,1]$. Its graph is fully contained in the unit square $[0,1]^2$.

Once more, we can resort to a raindrop experiment. Assume we can produce uniform rain on the unit square. The probability that a raindrop falls below the curve is equal to the area below the curve, which of course equals the integral we want to evaluate (the area of the unit square is 1, so we don't need to rescale the result).

A more formal justification for this is, using the fact that $f(x) = \int_0^{f(x)} 1 \, dt$,

$$\int_0^1 f(x) \, dx = \int_0^1 \int_0^{f(x)} 1 \, dt \, dx = \iint\limits_{\{(x,t):t \le f(x)\}} 1 dt \, dx = \frac{\iint\limits_{\{(x,t):t \le f(x)\}} 1 \, dt \, dx}{\iint\limits_{\{0 \le x,t \le 1\}} 1 dt \, dx}$$

The numerator is nothing other than the dark grey area under the curve, and the denominator is the area of the unit square (shaded in light grey in Figure 1.3). Thus the expression on the right hand side is the probability that a raindrop falls below the curve.

We have thus re-expressed our quantity of interest as a probability in a statistical model. Figure 1.3 shows the result obtained when observing 100 raindrops. 52 of them are below the curve, yielding a Monte-Carlo estimate of the integral of 0.52.

If after n raindrops a proportion $\hat{p}_n$ is found to lie below the curve, a $(1-2\alpha)$ confidence interval for the value of the integral is

$$\left[\hat{p}_n - z_{1-\alpha}\sqrt{\frac{\hat{p}_n(1-p_n)}{n}}, \hat{p}_n + z_{1-\alpha}\sqrt{\frac{\hat{p}_n(1-p_n)}{n}}\right]$$

Thus the speed of convergence of our (rather crude) Monte Carlo method is $\mathcal{O}_{\mathbb{P}}(n^{-1/2})$.

Remark 1.1 (Order of Convergence). Big-O notation is often used to describe the asymptotic rate of convergence of both deterministic sequences of real numbers and stochastic sequence of real-valued random variables. In this course we will make occasional use of two ideas:

– The real sequence A_n is said to be $\mathcal{O}(B_n)$ (or *of order* B_n), for some other real-sequence B_n, if there exist constants $M \in \mathbb{R}$ and $N \in \mathbb{N}$ such that:

$$\forall n > N, \qquad \left|\frac{A_n}{B_n}\right| < M.$$

– If A_n is instead a sequence of real-valued random variables, it is said to be $\mathcal{O}_{\mathbb{P}}(B_n)$ (or *of order* B_n *in probability*) for some sequence of real random variables (or constants, these being degenerate random variables which take a particular value with probability one), if for every $\epsilon > 0$ there exists $\delta_\epsilon, N_\epsilon$ such that:

$$\forall n > N_\epsilon, \qquad \mathbb{P}\left(\left|\frac{A_n}{B_n}\right| > \delta_\epsilon\right) < \epsilon.$$

◁

When using Riemann sums (as in Figure 1.4) to approximate the integral from example 1.2 the error is of order $\mathcal{O}(n^{-1})$.[1], Although the order of convergence can be improved when using the trapezoid rule and further improved by using Simpson's rule or other more sophisticated techniques, all of these methods suffer from *the curse of dimensionality*: their convergence rates worsen as the dimension of the problem increases.

[1] The error made for each "bar" can be upper bounded by $\frac{\Delta^2}{2} \max |f'(x)|$. Let n denote the number evaluations of f (and thus the number of "bars"). As Δ is proportional to $\frac{1}{n}$, the error made for each bar is $\mathcal{O}(n^{-2})$. As there are n "bars", the total error is $\mathcal{O}(n^{-1})$.

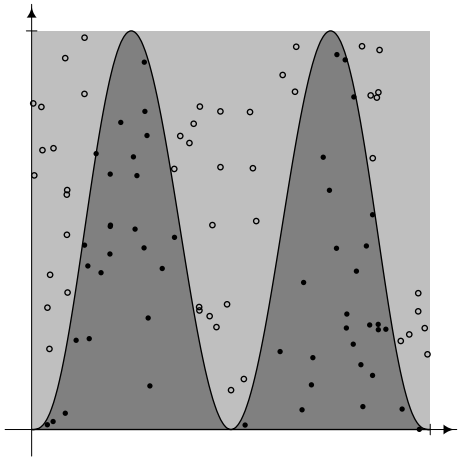


Figure 1.3. Illustration of the raindrop experiment to compute $\int_0^1 f(x)dx$

Recall that our Monte Carlo method was "only" of order $O_{\mathbb{P}}(n^{-1/2})$. However, it is easy to see that its speed of convergence is of the same order, regardless of the dimension of the support of f. This is not the case for other (deterministic) numerical integration methods. For a two-dimensional function f the error made by the Riemann approximation using n function evaluations is $\mathcal{O}(n^{-1/2})$. [2]

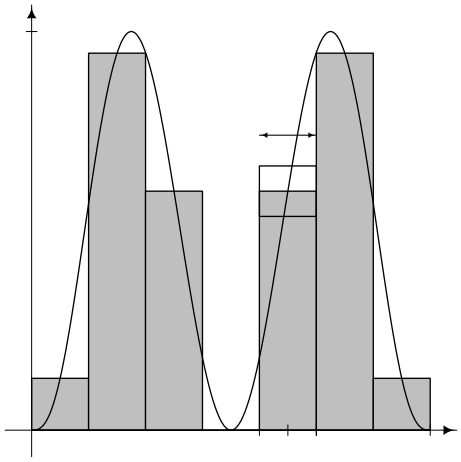


Figure 1.4. Illustration of numerical integration by Riemann sums

This makes the Monte Carlo methods especially suited for high-dimensional problems. Furthermore the Monte Carlo method offers the advantage of being relatively simple and thus easy to implement on a computer and, as we shall see later, it can be used in some settings in which numerical methods would be difficult to implement — such as the evaluation of integrals over non-compact sets.

1.2.2 Monte Carlo Testing

One of the simplest forms of simulation based inference goes under the name of *Monte Carlo Testing* or, sometimes, *randomized testing*. The idea is appealingly simple and rather widely applicable.

[2] Assume we partition both axes into m segments, i.e. we have to evaluate the function $n = m^2$ times. The error made for each "bar" is $\mathcal{O}(m^{-3})$ (each of the two sides of the base area of the "bar" is proportional to m^{-1}, so is the upper bound on $|f(x) - f(\xi_{\text{mid}})|$, yielding $\mathcal{O}(m^{-3})$). There are in total m^2 bars, so the total error is only $\mathcal{O}(m^{-1})$, or equivalently $\mathcal{O}(n^{-1/2})$.

Recall the basic idea of testing. Given a *null hypothesis* about the data, compute some *test statistic* (i.e. a real valued summary function of the observed data) whose distribution is known under the null hypothesis and would be expected to deviate systematically from this under the alternative hypothesis. If a value of the test statistic shows a deviation which would be expected no more than $\alpha\%$ of the time if the null hypothesis were true is observed, then one concludes that there is evidence which justifies *rejecting* the null hypothesis at the $\alpha\%$ level.

In principle this is reasonably straightforward, but there are often practical difficulties with following such a procedure. In particular, what if we do not know the distribution of the test statistic under the null hypothesis? The classical solution is to appeal to asymptotic theory to characterise the distribution of this statistic at least for large samples; this has two drawbacks: it is only *approximately* correct for finite samples and it can be extremely difficult to do.

One simple solution which seems to have been first suggested formally by Barnard (1963) is to use simulation. This approach has taken a little while to gain popularity, despite some far-sighted early work (Besag and Diggle, 1977, for example), perhaps in part because of limited computational resources and in part because of perceived difficulties with replication.

If T denotes the test statistic obtained from the actual data and $T_1, T_2, \ldots$ denote those obtained from repeated sampling of the null hypothesis, then, if the null hypothesis is true, $(T_1, \ldots, T_k, T)$ comprises a collection of $k+1$ iid replicates of the test statistic. The probability that T is the largest of $(T_1, \ldots, T_k, T)$ is exactly $1/(k+1)$ (by symmetry) and the probability that T is in the largest l of $(T_1, \ldots, T_k, T)$ is, similarly, $l/(k+1)$.

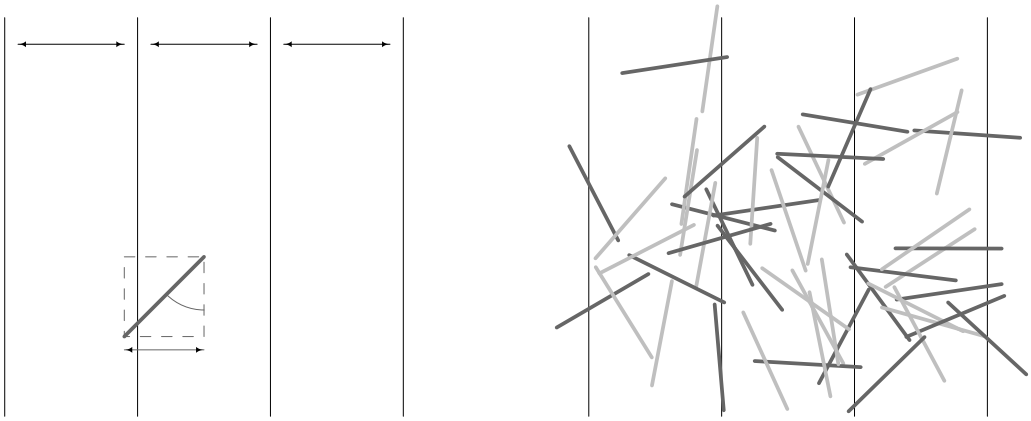
By this reasoning, we can construct a hypothesis test at the 5% significance level by drawing $k = 19$ realisations of the test statistic and rejecting the null hypothesis if and only if $T_\star$ is greater than any of those synthetic replicates. This test is clearly *exact*: the probability of rejection if the null hypothesis is true is exactly as is specified. However, there is a loss of power as a result of the randomization and, there is no guarantee that two people presented with the same data will reach the same conclusion (if they both simulate *different* artificial replicates then one may reject and the other may not). However, for "large enough" value of k these departures from the exact idealised test which this Monte Carlo procedure mimics are very small.

Although this idea might seem slightly arcane and removed from the other ideas which we've been discussing in this section it really is motivated by the same ideas. The empirical distribution of the artificial sample of test statistics converges to the true sampling distribution as the sample size becomes large and we're then just using the empirical quantiles as a proxy for the quantiles of the true distribution. With a little bit of care, as seen here, this can be done in such a way that the type I error probability is exactly that specified by the level of the test.

1.3 A Brief History of Monte Carlo Methods

Experimental Mathematics is an old discipline: the Old Testament (1 Kings vii. 23 and 2 Chronicles iv. 2) contains a rough estimate of π (using the columns of King Solomon's temple). Monte Carlo methods as we would understand them are a somewhat more recent develeopment, but still predate digital computers by a significant amount. One of the first documented Monte Carlo experiments is *Buffon's needle* experiment (see Example 1.3 below). Laplace (1812) suggested that this experiment can be used to approximate π.

Example 1.3 (Buffon's needle). In 1733, the Comte de Buffon, George Louis Leclerc, asked the following question (Buffon, 1733): Consider a floor with equally spaced lines, a distance δ apart. What is the probability that a needle of length $l < \delta$ dropped on the floor will intersect one of the lines?
Buffon answered the question himself in 1777 (Buffon, 1777).



(a) Illustration of the geometry behind *Buffon's needle*

(b) Results of the *Buffon's needle* experiment using 50 needles. Dark needles intersect the thin vertical lines, light needles do not.

Figure 1.5. Illustration of *Buffon's needle*

Assume the needle landed such that its angle is θ (see Figure 1.5). Then the question whether the needle intersects a line is equivalent to the question whether a box of width $l \sin\theta$ intersects a line. The probability of this happening is

$$\mathbb{P}(\text{intersect}|\theta) = \frac{l \sin\theta}{\delta}.$$

Assuming that the angle θ is uniform on $[0, \pi)$ we obtain

$$\mathbb{P}(\text{intersect}) = \int_0^\pi \mathbb{P}(\text{intersect}|\theta) \cdot \frac{1}{\pi}\, d\theta = \int_0^\pi \frac{l \sin\theta}{\delta} \cdot \frac{1}{\pi}\, d\theta = \frac{l}{\pi\delta} \cdot \underbrace{\int_0^\pi \sin\theta\, d\theta}_{=2} = \frac{2l}{\pi\delta}.$$

When dropping n needles the expected number of needles crossing a line is thus

$$\frac{2nl}{\pi\delta}.$$

Thus we can estimate π by

$$\pi \approx \frac{2nl}{X\delta},$$

where X is the number of needles crossing a line.
The Italian mathematician Mario Lazzarini performed Buffon's needle experiment in 1901 using a needle of length $l = 2.5cm$ and lines $d = 3cm$ apart (Lazzarini, 1901). Of 3408 needles 1808 needles crossed a line, so Lazzarini's estimate of π was

$$\pi \approx \frac{2 \cdot 3408 \cdot 2.5}{1808 \cdot 3} = \frac{17040}{5424} = \frac{355}{133},$$

which is nothing other than the best rational approximation to π with at most 4 digits each in the denominator and the numerator. This is quite remarkable, and the fact that Lazzarini's experiment was *so* precise casts some doubt over the results of his experiments (see Badger, 1994, for a more detailed discussion). ◁

Historically, the main drawback of Monte Carlo methods was that they used to be expensive to carry out. Physical random experiments were difficult to perform and so was the numerical processing of their results.

This however changed fundamentally with the advent of the digital computer. Amongst the first to realise this potential were John von Neumann and Stanisław Ulam, who were then working for the Manhattan project in Los Alamos. They proposed in 1947 to use a computer simulation for solving the problem of neutron diffusion in fissionable material (Metropolis, 1987). Enrico Fermi previously considered using Monte Carlo techniques in the calculation of neutron diffusion, however he proposed to use a mechanical device, the so-called "Fermiac", for generating the randomness. The name "Monte Carlo" goes back to Stanisław Ulam, who claimed to be stimulated by playing poker and whose uncle once borrowed money from him to go gambling in Monte Carlo (Ulam, 1983). In 1949 Metropolis and Ulam published their results in the *Journal of the American Statistical Association* (Metropolis and Ulam, 1949). Nonetheless, in the following 30 years Monte Carlo methods were used and analysed predominantly by physicists, and not by statisticians: it was only in the 1980s — following the paper by Geman and Geman (1984) proposing the Gibbs sampler — that the relevance of Monte Carlo methods in the context of (Bayesian) statistics was fully realised.

1.4 Pseudo-random numbers

For any Monte-Carlo simulation we need to be able to reproduce randomness by a computer algorithm, which, by definition, is deterministic in nature — a philosophical paradox. Indeed, von Neumann is quoted to have said: "Any one who considers artithmetical methods of reproducing random digits is, of course, in a state of sin. ... there is no such thing as a random number — there are only methods of producing random numbers, and a strict arithmetic procedure is of course not such a method."

In the following chapters we will assume that independent (pseudo-)random realisations from a uniform $\mathsf{U}[0,1]$ distribution are available. We will soon see that by using only the $\mathsf{U}(0,1)$ distribution as a source of randomness we can obtain samples from other distributions by using deterministic algorithms. are readily available. This section tries to give very brief overview of how pseudo-random numbers can be generated. For a more detailed discussion of pseudo-random number generators see Ripley (1987) or Knuth (1997).

A pseudo-random number generator (PRNG or, slightly abusively, RNG) is an algorithm for whose output the $\mathsf{U}[0,1]$ distribution is a suitable model. In other words, the number generated by the pseudo-random number generator should have the same *relevant* statistical properties as independent realisations of a $\mathsf{U}[0,1]$ random variable. Most importantly:

- The numbers generated by the algorithm should reproduce independence, i.e. the numbers $X_1, \ldots, X_n$ that we have already generated should not contain any discernible information on the next value X_{n+1}.
- The numbers generated should be spread out evenly across the interval $[0,1]$.

In the following we will briefly discuss the linear congruential generator. It is not a particularly powerful generator (so we discourage you from using it in practise), however it is simple enough to allow some insight into how pseudo-random number generators work.

Algorithm 1.1 (Congruential pseudo-random number generator). 1. Choose $a, M \in \mathbb{N}$, $c \in \mathbb{N}_0$, and the initial value ("seed") $Z_0 \in \{1, \ldots M-1\}$.

2. For $i = 1, 2, \ldots$

Set $Z_i = (aZ_{i-1} + c) \mod M$, and $X_i = Z_i/M$.

The integers Z_i generated by the algorithm are from the set $\{0, 1, \ldots, M-1\}$ and thus the X_i are in the interval $[0, 1)$.

It is easy to see that the sequence of pseudo-random numbers only depends on the seed X_0. Running the pseudo-random number generator twice with the same seed thus generates exactly the same sequence of pseudo-random numbers. This can be a very useful feature when debugging your own code.

Example 1.4. Consider the choice of $a = 81$, $c = 35$, $M = 256$, and seed $Z_0 = 4$.

$$\begin{aligned} Z_1 &= (81 \cdot 4 + 35) \mod 256 = 359 \mod 256 = 103 \\ Z_2 &= (81 \cdot 103 + 35) \mod 256 = 8378 \mod 256 = 186 \\ Z_3 &= (81 \cdot 186 + 35) \mod 256 = 15101 \mod 256 = 253 \\ &\ldots \end{aligned}$$

The corresponding X_i are $X_1 = 103/256 = 0.4023438$, $X_2 = 186/256 = 0.72656250$, $X_1 = 253/256 = 0.98828120$. ◁

The main flaw of the congruential generator its "crystalline" nature (Marsaglia, 1968). If the sequence of generated values $X_1, X_2, \ldots$ is viewed as points in an n-dimension cube (the $(k+1)$-th point has the coordinates $(X_{nk+1}, \ldots, X_{nk+n})$), they lie on a finite, and often very small number of parallel hyperplanes. Or as Marsaglia (1968) put it: "the points [generated by a congruential generator] are about as randomly spaced in the unit n-cube as the atoms in a perfect crystal at absolute zero." The number of hyperplanes depends on the choice of a, c, and M.

An example of a notoriously poor design of a congruential pseudo-random number generator is RANDU, which was (unfortunately) very popular in the 1970s and used for example in IBM's System/360 and System/370, and Digital's PDP-11. It used $a = 2^{16} + 3$, $c = 0$, and $M = 2^{31}$. The numbers generated by RANDU lie on only 15 hyperplanes in the 3-dimensional unit cube (see Figure 1.6).

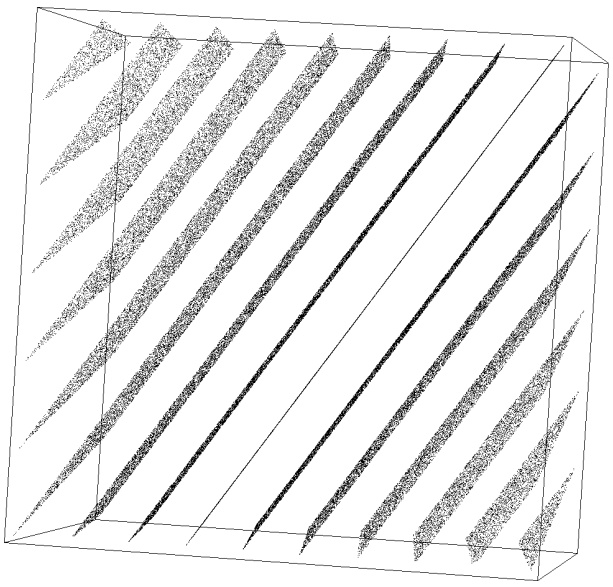
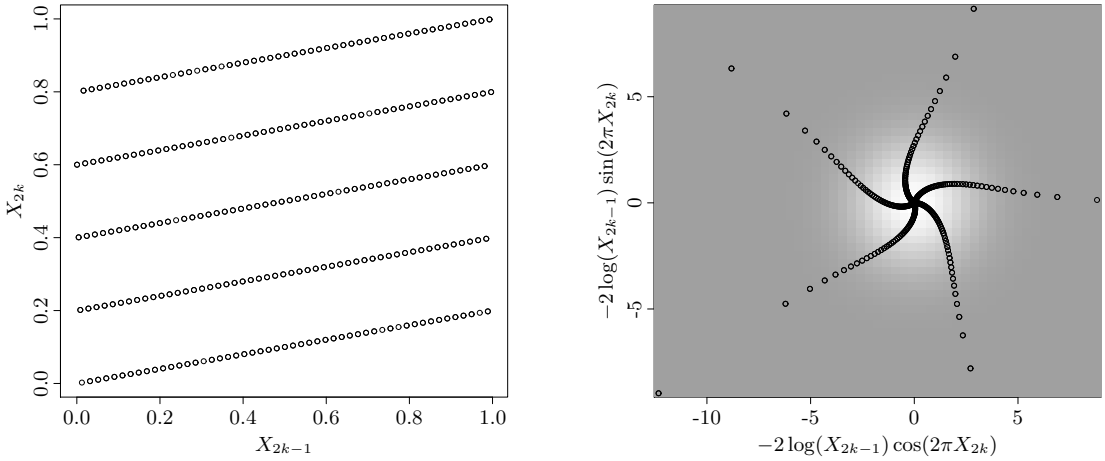


Figure 1.6. 300,000 realisations of the RANDU pseudo-random number generator plotted in 3D. A point corresponds to a triplet $(x_{3k-2}, x_{3k-1}, x_{3k})$ for $k = 1, \ldots, 100000$. The data points lie on 15 hyperplanes.



(a) 1,000 realisations of this congruential generator plotted in 2D.

(b) Supposedly bivariate Gaussian pseudo-random numbers obtained using the pseudo-random numbers shown in panel (a).

Figure 1.7. Results obtained using a congruential generator with $a = 1229$, $c = 1$, and $M = 2^{11}$

Figure 1.7 shows another cautionary example (taken from Ripley, 1987). The left-hand panel shows a plot of 1,000 realisations of a congruential generator with $a = 1229$, $c = 1$, and $M = 2^{11}$. The random numbers lie on only 5 hyperplanes in the unit square. The right hand panel shows the outcome of the Box-Muller method for transforming two uniform pseudo-random numbers into a pair of independent standard normal random variates (see example 2.2).

This is a serious deficiency of the congruential pseudo-random number generator, it should not be used in Monte Carlo experiments (although judicious choice of the constants can lead to somewhat better performance than that demonstrated here). For more powerful pseudo-random number generators see e.g. Marsaglia and Zaman (1991) or Matsumoto and Nishimura (1998). GNU R (and other environments) provide you with a large choice of powerful random number generators, see the corresponding help page (`?RNGkind`) for details. For compiled code, the GNU Scientific Library Galassi et al. (2002) provides a similar range of good random number generators.

2. Fundamental Concepts: Transformation, Rejection, and Reweighting

2.1 Transformation Methods

In Section 1.4 we saw one (not very good) method to create (pseudo-)random numbers from the uniform distribution $\mathsf{U}[0,1]$. Many better methods exist and we shall assume for the remainder of this module that such a method is available. We now turn to the issue of obtaining (pseudo-)random samples from other distributions by transforming (pseudo-)random samples from this uniform distribution.

One of the simplest methods of generating random samples from a distribution with cumulative distribution function (CDF) $F(x) = \mathbb{P}(X \leq x)$ is based on the inverse of the CDF.

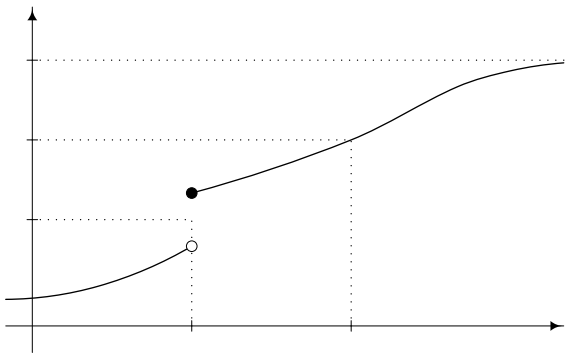


Figure 2.1. Illustration of the definition of the generalised inverse F^- of a CDF F

The CDF is an increasing function, however it is not necessarily continuous. Thus we define the *generalised inverse* $F^-(u) := \inf\{x : \ F(x) \geq u\}$. Figure 2.1 illustrates its definition. If F is continuous, then $F^-(u) = F^{-1}(u)$.

Theorem 2.1 (Inversion Method). *Let $U \sim \mathsf{U}[0,1]$ and F be a CDF. Then $F^-(U)$ has the CDF F.*

Proof. It is easy to see (e.g. in Figure 2.1) that $F^-(u) \leq x$ is equivalent to $u \leq F(x)$. Thus for $U \sim \mathsf{U}[0,1]$

$$\mathbb{P}(F^-(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x),$$

thus F is the CDF of $X = F^-(U)$. □

Example 2.1 (Exponential Distribution). The exponential distribution with rate $\lambda > 0$ has the CDF $F_\lambda(x) = 1 - \exp(-\lambda x)$ for $x \geq 0$. Thus $F_\lambda^-(u) = F_\lambda^{-1}(u) = -\log(1-u)/\lambda$. Thus we can generate random

samples from Expo(λ) by applying the transformation $-\log(1-U)/\lambda$ to a uniform U$[0,1]$ random variable U. As U and $1-U$, of course, have the same distribution we can use $-\log(U)/\lambda$ as well. ◁

The Inversion Method is a very efficient tool for generating random numbers. However, few distributions possess a CDF whose (generalised) inverse can be evaluated efficiently. Take the example of the Normal distribution, whose CDF is not even available in closed form.

Note, however, that the generalised inverse of the CDF is just one possible transformation and that there might be other transformations that yield samples from the desired distribution. An example of such a method is the Box-Muller method for generating Normal random variables.

Example 2.2 (Box-Muller Method for Generating Normals (Box and Muller, 1958)). Using the transformation of density formula one can show that $X_1, X_2 \overset{\text{iid}}{\sim} \mathsf{N}(0,1)$ iff their polar coordinates (R, θ) with

$$X_1 = R \cdot \cos(\theta), \qquad X_2 = R \cdot \sin(\theta)$$

are independent, $\theta \sim \mathsf{U}[0, 2\pi]$, and $R^2 \sim \text{Expo}(1/2)$. Using $U_1, U_2 \overset{\text{iid}}{\sim} \mathsf{U}[0,1]$ and Example 2.1 we can generate R and θ by

$$R = \sqrt{-2\log(U_1)}, \qquad \theta = 2\pi U_2$$

and thus

$$X_1 = \sqrt{-2\log(U_1)} \cdot \cos(2\pi U_2), \qquad X_2 = \sqrt{-2\log(U_1)} \cdot \sin(2\pi U_2)$$

are two independent realisations from a $\mathsf{N}(0,1)$ distribution. ◁

The idea of transformation methods like the Inversion Method was to generate random samples from a distribution other than the target distribution and to transform them such that they come from the desired target distribution. Transformation methods such as those described here are typically extremely efficient but it can be difficult to find simple transformations which produce samples from complicated distributions, especially in multivariate settings.

In these cases we have to proceed differently. One option is to sample from a distribution other than that of interest, in which case we have to find other ways of correcting for the fact that we sample from the "wrong" distribution. The next two sections present two such ideas: rejection sampling and importance sampling. Other approaches which involve more complicated simulation strategies than sampling independently from a distribution of interest also exist and we will look at some of these in Chapters 4 and 5.

2.2 Rejection Sampling

The basic idea of rejection sampling is to sample from an *instrumental distribution* (sometimes referred to as the *proposal distribution*.) and to reject samples that are "unlikely" under the target distribution in a principled way.

Assume that we want to sample from a target distribution whose density f is known to us. The simple idea underlying rejection sampling (and other Monte Carlo algorithms) is the rather trivial identity

$$f(x) = \int_0^{f(x)} 1 \, du = \int_0^{\infty} \underbrace{\mathbb{1}_{0<u<f(x)}}_{=:f(x,u)} du$$

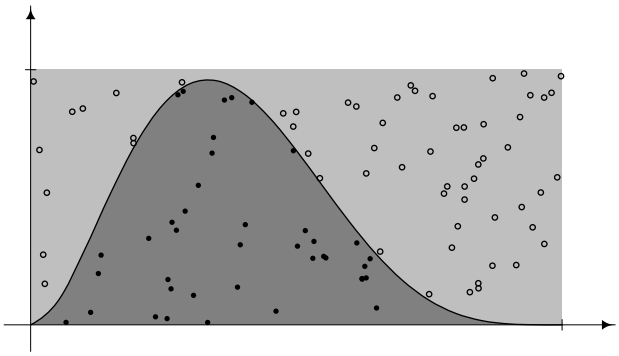


Figure 2.2. Illustration of example 2.3. Sampling from the area under the curve (dark grey) corresponds to sampling from the Beta$(3, 5)$ density. In example 2.3 we use a uniform distribution of the light grey rectangle as proposal distribution. Empty circles denote rejected values, filled circles denote accepted values.

Thus $f(x)$ can be interpreted as the marginal density of a uniform distribution on the area under the density $f(x)$, $\{(x, u) : \ 0 \leq u \leq f(x)\}$. This equivalence is very important in simulation, and has been referred to as the *fundamental theorem of simulation*. Figure 2.2 illustrates this idea.

This suggests that we can generate a sample from f by sampling from the area under the curve — but it doesn't tell us how to sample uniformly from this area, which may be quite complicated (especially if we try to extend the idea to sampling from the distribution of a multivariate random variable).

Example 2.3 (Sampling from a Beta distribution). The Beta(a, b) distribution $(a, b \geq 0)$ has the density

$$f(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}, \qquad \text{for } 0 < x < 1,$$

where $\Gamma(a) = \int_0^{+\infty} t^{a-1} \exp(-t)\, dt$ is the Gamma function. For $a, b > 1$ the Beta(a, b) density is unimodal with mode $(a-1)/(a+b-2)$. Figure 2.2 shows the density of a Beta$(3, 5)$ distribution. It attains its maximum of $1680/729 \approx 2.305$ at $x = 1/3$.

Using the above identity we can draw from Beta$(3, 5)$ by drawing from a uniform distribution on the area under the density $\{(x, u) : \ 0 < u < f(x)\}$ (the area shaded in dark gray in figure 2.2).

In order to sample from the area under the density, we will use a similar trick to that used in Examples 1.1 and 1.2. We will sample from the light grey rectangle and and keep only the samples that fall in the area under the curve. Figure 2.2 illustrates this idea.

Mathematically speaking, we sample independently $X \sim \mathrm{U}[0, 1]$ and $U \sim \mathrm{U}[0, 2.4]$. We keep the pair (X, U) if $U < f(X)$, otherwise we reject it.

The conditional probability that a pair (X, U) is kept if $X = x$ is

$$\mathbb{P}(U < f(X)|X = x) = \mathbb{P}(U < f(x)) = f(x)/2.4$$

As X and U were drawn independently we can rewrite our algorithm as: Draw X from U$[0, 1]$ and accept X with probability $f(X)/2.4$, otherwise reject X. ◁

The method proposed in Example 2.3 is based on bounding the density of the Beta distribution by a box. Whilst this is a powerful idea, it cannot be directly applied to other distributions, as the density might be unbounded or have infinite support. However we might be able to bound the density of $f(x)$ by $M \cdot g(x)$, where $g(x)$ is a density from which we can easily sample.

Algorithm 2.1 (Rejection sampling). Given two densities f, g with $f(x) < M \cdot g(x)$ for all x, we can generate a sample from f by

1. Draw $X \sim g$
2. Accept X as a sample from f with probability

$$\frac{f(X)}{M \cdot g(X)},$$

otherwise go back to step 1.

Proof. We have

$$\mathbb{P}(X \in \mathcal{X} \text{ and is accepted}) = \int_{\mathcal{X}} g(x) \underbrace{\frac{f(x)}{M \cdot g(x)}}_{=\mathbb{P}(X \text{ is accepted} | X=x)} dx = \frac{\int_{\mathcal{X}} f(x)\, dx}{M}, \tag{2.1}$$

and thus, denoting by E the set of all possible values X can take,

$$\mathbb{P}(X \text{ is accepted}) = \mathbb{P}(X \in E \text{ and is accepted}) = \frac{1}{M}, \tag{2.2}$$

yielding

$$\mathbb{P}(x \in \mathcal{X} | X \text{ is accepted}) = \frac{\mathbb{P}(X \in \mathcal{X} \text{ and is accepted})}{\mathbb{P}(X \text{ is accepted})} = \frac{\int_{\mathcal{X}} f(x)\, dx/M}{1/M} = \int_{\mathcal{X}} f(x)\, dx. \tag{2.3}$$

Thus the density of the values accepted by the algorithm is $f(\cdot)$. □

Remark 2.1. If we know f only up to a multiplicative constant, i.e. if we only know $\pi(x)$, where $f(x) = C \cdot \pi(x)$, we can carry out rejection sampling using

$$\frac{\pi(X)}{M \cdot g(X)}$$

as probability of accepting X, provided $\pi(x) < M \cdot g(x)$ for all x. Then by analogy with (2.1) - (2.3) we have

$$\mathbb{P}(X \in \mathcal{X} \text{ and is accepted}) = \int_{\mathcal{X}} g(x) \frac{\pi(x)}{M \cdot g(x)}\, dx = \frac{\int_{\mathcal{X}} \pi(x)\, dx}{M} = \frac{\int_{\mathcal{X}} f(x)\, dx}{C \cdot M},$$

$\mathbb{P}(X \text{ is accepted}) = 1/(C \cdot M)$, and thus

$$\mathbb{P}(x \in \mathcal{X} | X \text{ is accepted}) = \frac{\int_{\mathcal{X}} f(x)\, dx/(C \cdot M)}{1/(C \cdot M)} = \int_{\mathcal{X}} f(x)\, dx$$

Example 2.4 (Rejection sampling from the $\mathsf{N}(0,1)$ distribution using a Cauchy proposal). Assume we want to sample from the $\mathsf{N}(0,1)$ distribution with density

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

using a Cauchy distribution with density

$$g(x) = \frac{1}{\pi(1+x^2)}$$

as instrumental distribution. Of course, there is not much point is using this method is practice: the Box-Muller method is more efficient. The smallest M we can choose such that $f(x) \leq Mg(x)$ is $M = \sqrt{2\pi} \cdot \exp(-1/2)$. Figure 2.3 illustrates the results. As before, filled circles correspond to accepted values whereas open circles correspond to rejected values.

Note that it is impossible to do rejection sampling from a Cauchy distribution using a $\mathsf{N}(0,1)$ distribution as instrumental distribution: there is no $M \in \mathbb{R}$ such that

$$\frac{1}{\pi(1+x^2)} < M \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2}\right);$$

the Cauchy distribution has heavier tails than the Normal distribution. ◁

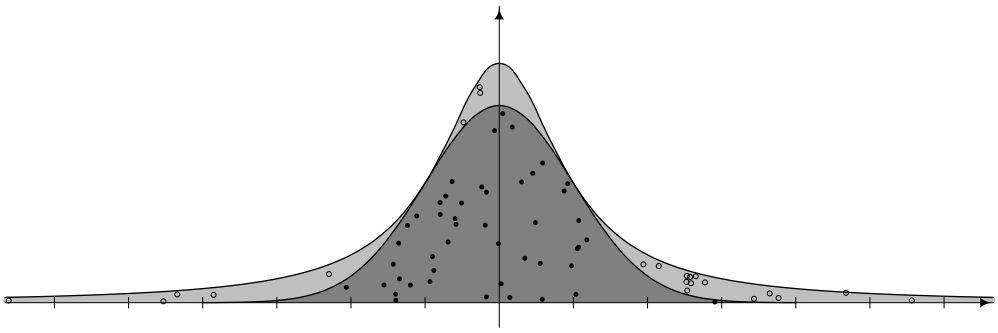


Figure 2.3. Illustration of example 2.3. Sampling from the area under the density $f(x)$ (dark grey) corresponds to sampling from the $\mathsf{N}(0,1)$ density. The proposal $g(x)$ is a $\mathsf{Cauchy}(0,1)$.

2.3 Importance Sampling

In rejection sampling we have compensated for the fact that we sampled from the instrumental distribution $g(x)$ instead of $f(x)$ by rejecting some of the proposed values. *Importance sampling* is based on the idea of instead using *weights* to correct for the fact that we sample from the instrumental distribution $g(x)$ instead of the target distribution $f(x)$.

Indeed, importance sampling is based on the identity

$$\mathbb{P}(X \in \mathcal{X}) = \int_{\mathcal{X}} f(x)\, dx = \int_{\mathcal{X}} g(x) \underbrace{\frac{f(x)}{g(x)}}_{=:w(x)}\, dx = \int_{\mathcal{X}} g(x)w(x)\, dx \tag{2.4}$$

for all $g(\cdot)$, such that $g(x) > 0$ for (almost) all x with $f(x) > 0$. We can generalise this identity by considering the expectation $\mathbb{E}_f(h(X))$ of a measurable function h:

$$\mathbb{E}_f(h(X)) = \int f(x)h(x)\, dx = \int g(x) \underbrace{\frac{f(x)}{g(x)}}_{=:w(x)} h(x)\, dx = \int g(x)w(x)h(x)\, dx = \mathbb{E}_g(w(X) \cdot h(X)), \tag{2.5}$$

if $g(x) > 0$ for (almost) all x with $f(x) \cdot h(x) \neq 0$.

Assume we have a sample $X_1, \ldots, X_n \sim g$. Then, provided $\mathbb{E}_g|w(X) \cdot h(X)|$ exists,

$$\frac{1}{n}\sum_{i=1}^{n} w(X_i)h(X_i) \overset{a.s.}{\underset{n\to\infty}{\longrightarrow}} \mathbb{E}_g(w(X) \cdot h(X))$$

and thus by (2.5)

$$\frac{1}{n}\sum_{i=1}^{n} w(X_i)h(X_i) \overset{a.s.}{\underset{n\to\infty}{\longrightarrow}} \mathbb{E}_f(h(X)).$$

In other words, we can estimate $\mu := \mathbb{E}_f(h(X))$ by using

$$\tilde{\mu} := \frac{1}{n}\sum_{i=1}^{n} w(X_i)h(X_i)$$

Note that whilst $\mathbb{E}_g(w(X)) = \int_E \frac{f(x)}{g(x)} g(x)\, dx = \int_E f(x) = 1$, the weights $w_1(X), \ldots, w_n(X)$ do not necessarily sum up to n, so one might want to consider the *self-normalised* version

$$\hat{\mu} := \frac{1}{\sum_{i=1}^{n} w(X_i)} \sum_{i=1}^{n} w(X_i)h(X_i).$$

This gives rise to the following algorithm:

Algorithm 2.2 (Importance Sampling). Choose g such that $\text{supp}(g) \supset \text{supp}(f \cdot h)$.

1. For $i = 1, \ldots, n$:
   i. Generate $X_i \sim g$.
   ii. Set $w(X_i) = \frac{f(X_i)}{g(X_i)}$.
2. Return either

$$\hat{\mu} = \frac{\sum_{i=1}^n w(X_i)h(X_i)}{\sum_{i=1}^n w(X_i)}$$

or

$$\tilde{\mu} = \frac{\sum_{i=1}^n w(W_i)h(X_i)}{n}$$

The following theorem gives the bias and the variance of importance sampling.

Theorem 2.2 (Bias and Variance of Importance Sampling). *(a)* $\mathbb{E}_g(\tilde{\mu}) = \mu$

(b) $\text{Var}_g(\tilde{\mu}) = \dfrac{\text{Var}_g(w(X) \cdot h(X))}{n}$

(c) $\mathbb{E}_g(\hat{\mu}) = \mu + \dfrac{\mu \text{Var}_g(w(X)) - \text{Cov}_g(w(X), w(X) \cdot h(X))}{n} + O(n^{-2})$

(d) $\text{Var}_g(\hat{\mu}) = \dfrac{\text{Var}_g(w(X) \cdot h(X)) - 2\mu \text{Cov}_g(w(X), w(X) \cdot h(X)) + \mu^2 \text{Var}_g(w(X))}{n} + O(n^{-2})$

Proof. (a) $\mathbb{E}_g\left(\dfrac{1}{n}\sum_{i=1}^n w(X_i)h(X_i)\right) = \dfrac{1}{n}\sum_{i=1}^n \mathbb{E}_g(w(X_i)h(X_i)) = \mathbb{E}_f(h(X))$

(b) $\text{Var}_g\left(\dfrac{1}{n}\sum_{i=1}^n w(X_i)h(X_i)\right) = \dfrac{1}{n^2}\sum_{i=1}^n \text{Var}_g(w(X_i)h(X_i)) = \dfrac{\text{Var}_g(w(X)h(X))}{n}$

(c) and (d) see (Liu, 2001, p. 35) □

□

Note that the theorem implies that contrary to $\tilde{\mu}$ the self-normalised estimator $\hat{\mu}$ is biased. The self-normalised estimator $\hat{\mu}$ however might have a lower variance. In addition, it has another advantage: we only need to know the density up to a multiplicative constant, as is often the case in Bayesian modelling, for example. Assume $f(x) = C \cdot \pi(x)$, then

$$\hat{\mu} = \frac{\sum_{i=1}^n w(X_i)h(X_i)}{\sum_{i=1}^n w(X_i)} = \frac{\sum_{i=1}^n \frac{f(X_i)}{g(X_i)}h(X_i)}{\sum_{i=1}^n \frac{f(X_i)}{g(X_i)}w(X_i)} = \frac{\sum_{i=1}^n \frac{C \cdot \pi(X_i)}{g(X_i)}h(X_i)}{\sum_{i=1}^n \frac{C \cdot \pi(X_i)}{g(X_i)}w(X_i)} = \frac{\sum_{i=1}^n \frac{\pi(X_i)}{g(X_i)}h(X_i)}{\sum_{i=1}^n \frac{\pi(X_i)}{g(X_i)}w(X_i)},$$

i.e. the self-normalised estimator $\hat{\mu}$ does not depend on the normalisation constant C. By a closely analogous argument, one can show that is also enough to know g only up to a multiplicative constant. On the other hand, as we have seen in the proof of Theorem 2.2 it is a lot harder to analyse the theoretical properties of the self-normalised estimator $\hat{\mu}$.

Although the above equations (2.4) and (2.5) hold for every g with $\text{supp}(g) \supset \text{supp}(f \cdot h)$ and the importance sampling algorithm converges for a large choice of such g, one typically only considers choices of g that lead to *finite variance estimators*. The following two conditions are each sufficient (albeit rather restrictive; see Geweke (1989) for some other possibilities) fto ensure that $\tilde{\mu}$ has finite variance:

– $f(x) < M \cdot g(x)$ and $\text{Var}_f(h(X)) < \infty$.
– E is compact, f is bounded above on E, and g is bounded below on E.

So far we have only studied whether an g is an appropriate instrumental distribution, i.e. whether the variance of the estimator $\tilde{\mu}$ (or $\hat{\mu}$) is finite. This leads to the question which instrumental distribution is *optimal*, i.e. for which choice $\text{Var}(\tilde{\mu})$ is minimal. The following theorem answers this question:

Theorem 2.3 (Optimal proposal). *The proposal distribution g that minimises the variance of $\tilde{\mu}$ is*

$$g^*(x) = \frac{|h(x)|f(x)}{\int_E |h(t)|f(t)\,dt}.$$

Proof. We have from Theorem 2.2 (b) that

$$n\cdot\mathrm{Var}_g(\tilde{\mu}) = \mathrm{Var}_g\left(w(X)\cdot h(X)\right) = \mathrm{Var}_g\left(\frac{h(X)\cdot f(X)}{g(X)}\right) = \mathbb{E}_g\left(\left(\frac{h(X)\cdot f(X)}{g(X)}\right)^2\right) - \Bigg(\underbrace{\mathbb{E}_g\left(\frac{h(X)\cdot f(X)}{g(X)}\right)}_{=\mathbb{E}_g(\tilde{\mu})=\mu}\Bigg)^2.$$

The second term is independent of the choice of proposal distribution, thus we only have to minimise $\mathbb{E}_g\left(\left(\frac{h(X)\cdot f(X)}{g(X)}\right)^2\right)$. Substituting in $g^\star$ we obtain:

$$\begin{aligned}\mathbb{E}_{g^\star}\left(\left(\frac{h(X)\cdot f(X)}{g^\star(X)}\right)^2\right) &= \int_E \frac{h(x)^2\cdot f(x)^2}{g^\star(x)}\,dx = \left(\int_E \frac{h(x)^2\cdot f(x)^2}{|h(x)|f(x)}\,dx\right)\cdot\left(\int_E |h(t)|f(t)\,dt\right)\\ &= \left(\int_E |h(x)|f(x)\,dx\right)^2\end{aligned}$$

On the other hand, we can apply Jensen's inequality to $\mathbb{E}_g\left(\left(\frac{h(X)\cdot f(X)}{g(X)}\right)^2\right)$ yielding

$$\mathbb{E}_g\left(\left(\frac{h(X)\cdot f(X)}{g(X)}\right)^2\right) \geq \left(\mathbb{E}_g\left(\frac{|h(X)|\cdot f(X)}{g(X)}\right)\right)^2 = \left(\int_E |h(x)|f(x)\,dx\right)^2$$

i.e. the estimator obtained by using an importance sampler employing instrumental distribution $g^\star$ attains the minimal possible variance amongst the class of importance sampling estimators.

□

An important corollary of Theorem 2.3 is that importance sampling can be *super-efficient*, i.e. when using the optimal $g^\star$ from Theorem 2.3 the variance of $\tilde{\mu}$ is less than the variance obtained when sampling directly from f:

$$\begin{aligned} & n\cdot\mathrm{Var}_f\left(\frac{h(X_1)+\ldots+h(X_n)}{n}\right) = \mathbb{E}_f(h(X)^2) - \mu^2\\ \geq\ & \left(\mathbb{E}_f|h(X)|\right)^2 - \mu^2 = \left(\int_E |h(x)|f(x)\,dx\right)^2 - \mu^2 = n\cdot\mathrm{Var}_{g^\star}(\tilde{\mu})\end{aligned}$$

by Jensen's inequality. Unless h is (almost surely) constant the inequality is strict. There is an intuitive explanation to the super-efficiency of importance sampling. Using $g^\star$ instead of f causes us to focus on regions of high probability where $|h|$ is large, which contribute most to the integral $\mathbb{E}_f(h(X))$.

Theorem 2.3 is, however, a rather formal optimality result. When using $\tilde{\mu}$ we need to know the normalisation constant of $g^\star$, which is exactly the integral we are attempting to approximate! Furthermore, we need to be able to draw samples from $g^\star$ efficiently. The practically important implication of Theorem 2.3 is that we should choose an instrumental distribution g whose shape is close to the one of $f\cdot|h|$.

Example 2.5 (Computing $\mathbb{E}_f|X|$ for $X\sim \mathsf{t}_3$). Assume we want to compute $\mathbb{E}_f|X|$ for X from a t-distribution with 3 degrees of freedom (t_3) using a Monte Carlo method. Three different schemes are considered

– Sampling $X_1,\ldots,X_n$ directly from t_3 and estimating $\mathbb{E}_f|X|$ by

$$\frac{1}{n}\sum_{i=1} n|X_i|.$$

– Alternatively we could use importance sampling using a t_1 (which is nothing other than a Cauchy distribution) as instrumental distribution. The idea behind this choice is that the density $g_{t_1}(x)$ of a t_1 distribution is closer to $f(x)|x|$, where $f(x)$ is the density of a t_3 distribution, as figure 2.4 shows.
– Third, we will consider importance sampling using a $N(0,1)$ distribution as instrumental distribution.

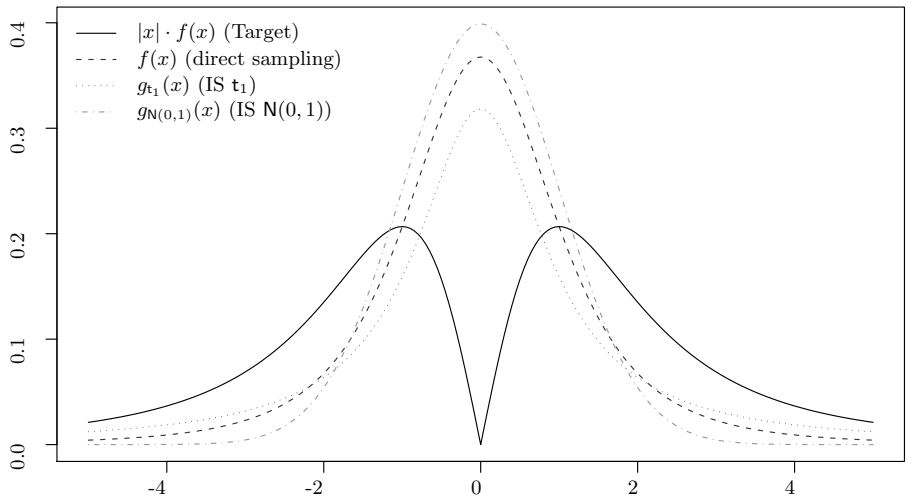


Figure 2.4. Illustration of the different instrumental distributions in example 2.5.

Note that the third choice yields weights of infinite variance, as the instrumental distribution ($N(0,1)$) has lighter tails than the distribution we want to sample from (t_3). The right-hand panel of figure 2.5 illustrates that this choice yields a very poor estimate of the integral $\int |x| f(x)\, dx$.
Sampling directly from the t_3 distribution can be seen as importance sampling with all weights $w_i \equiv 1$, this choice clearly minimises the variance of the weights. This however does not imply that this yields an estimate of the integral $\int |x| f(x)\, dx$ of minimal variance. Indeed, after 1500 iterations the empirical standard deviation (over 100 realisations) of the direct estimate is 0.0345, which is larger than the empirical standard deviation of $\tilde{\mu}$ when using a t_1 distribution as instrumental distribution, which is 0.0182. This suggests that using a t_1 distribution as instrumental distribution is super-efficient (see Figure 2.5) although we should always be careful when assuming that empirical standard deviations are a good approximation of the true standard deviation.
Figure 2.6 somewhat explains why the t_1 distribution is a far better choice than the $N(0,1)$ distributon. As the $N(0,1)$ distribution does not have heavy enough tails, the weight tends to infinity as $|x| \to +\infty$. Thus large $|x|$ can receive *very* large weights, causing the jumps of the estimate $\tilde{\mu}$ shown in figure 2.5. The t_1 distribution has heavy enough tails, to ensure that the weights are small for large values of $|x|$, explaining the small variance of the estimate $\tilde{\mu}$ when using a t_1 distribution as instrumental distribution. ◁

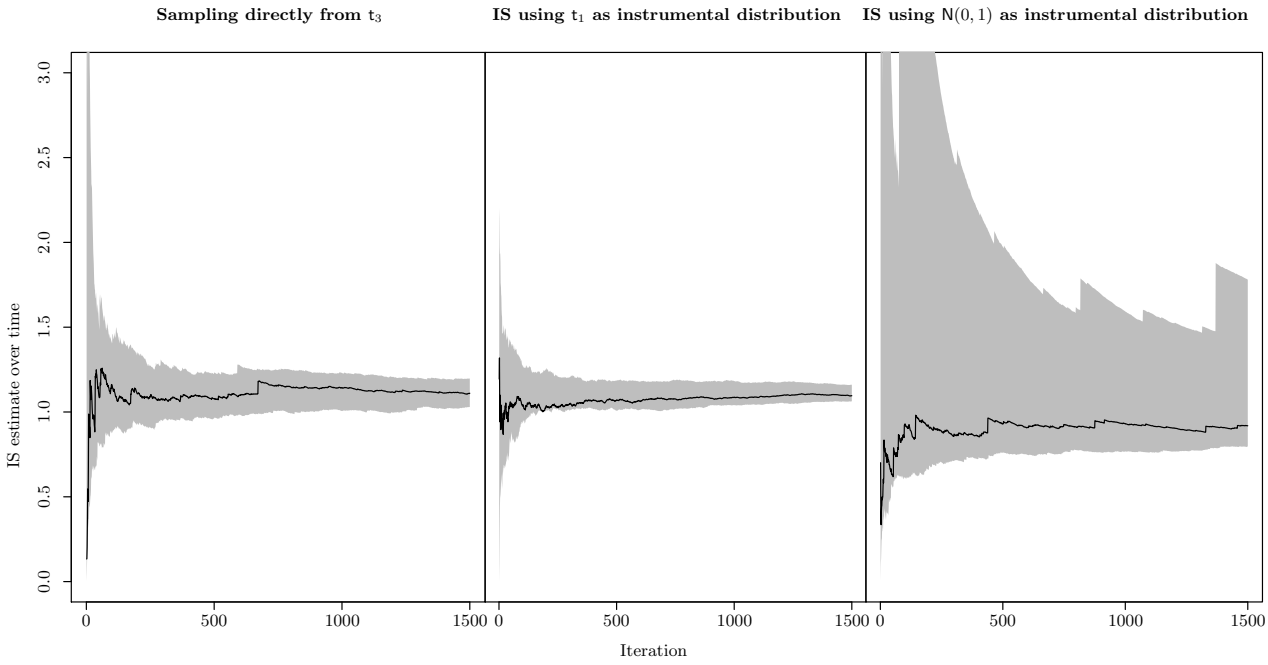


Figure 2.5. Estimates of $\mathbb{E}|X|$ for $X \sim \mathrm{t}_3$ obtained after 1 to 1500 iterations. The three panels correspond to the three different sampling schemes used. The areas shaded in grey correspond to the range of 100 replications.

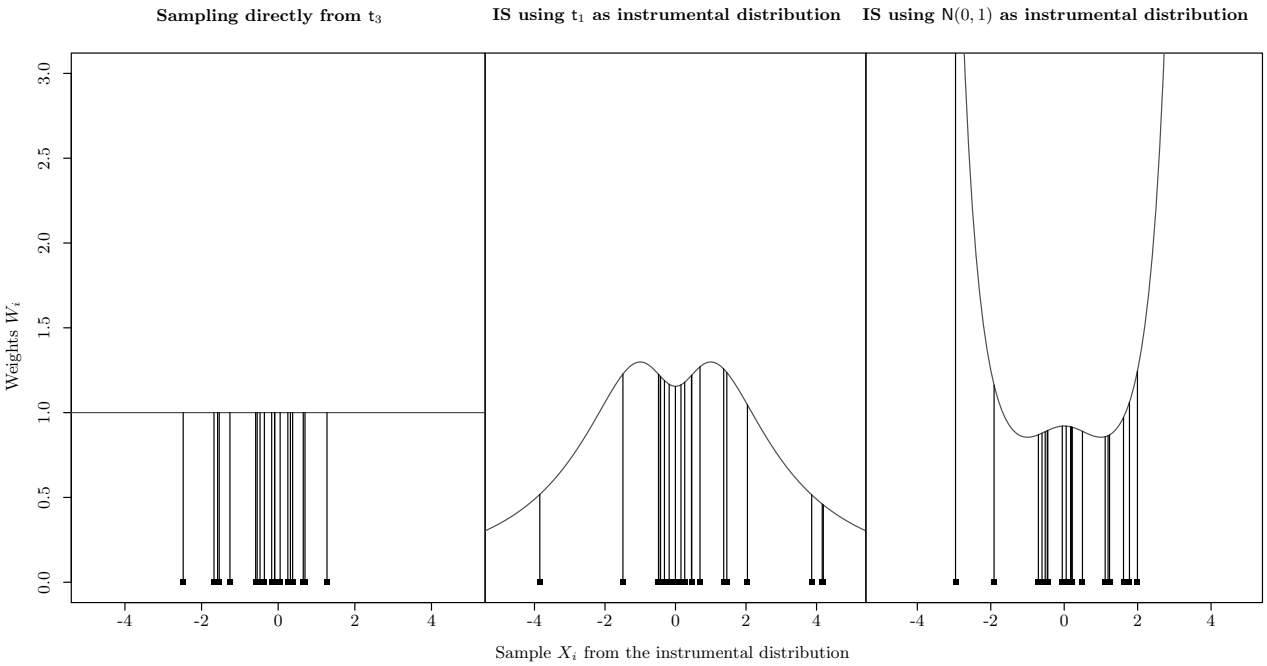


Figure 2.6. Weights W_i obtained for 20 realisations X_i from the different instrumental distributions.

3. Markov Chains

We begin this chapter by noting that the material presented here is covered in much greater depth within the *ST208 Applied Stochastic Processes* module. What is presented here is no more than a very brief introduction to certain aspects of stochastic processes, together with those details which are essential to understanding the remainder of this course. If you are interested in further details then a rigorous, lucid and inexpensive reference which starts from the basic principles is provided by (Gikhman and Skorokhod, 1996). We note, in particular, that a completely rigorous treatment of this area requires a number of measure theoretic concepts which are beyond the scope of this course. We will largely neglect these issues here in the interests of clarity of exposition, whilst attempting to retain the essence of the concepts which are presented. If you are not familiar with measure theory then please ignore the occasional references to *measurability*.

3.1 Stochastic Processes

For our purposes we can define an E-valued *process* as a function $\xi : \mathcal{I} \to E$ which maps values in some index set $\mathcal{I}$ to some other space E. The evolution of the process is described by considering the variation of $\xi(i)$ with i. An E-valued *stochastic process* (or *random process*) can be viewed as a process in which, for each $i \in \mathcal{I}$, $\xi(i)$ is a random variable taking values in E.

Although a rich literature on more general situations exists, we will consider only the case of *discrete time stochastic processes* in which the index set $\mathcal{I}$ is $\mathbb{N}$ (of course, any index set isomorphic to $\mathbb{N}$ can be used in the same framework by simple relabeling). We will use the notation ξ_i to indicate the value of the process at *time* i (note that there need be no connection between the index set and *real* time, but this terminology is both convenient and standard).

We will begin with an extremely brief description of a general stochastic process, before moving on to discuss the particular classes of process in which we will be interested. In order to characterise a stochastic process of the sort in which we are interested, it is sufficient to know all of its *finite dimensional distributions*, the joint distributions of the process at any collection of finitely many times. For any collection of times $i_1, i_2, \ldots, i_t$ and any *measurable* collection of subsets of E, $A_{i_1}, A_{i_2}, \ldots, A_{i_t}$ we are interested in the probability:

$$\mathbb{P}\left(\xi_{i_1} \in A_{i_1}, \xi_{i_2} \in A_{i_2}, \ldots, \xi_{i_t} \in A_{i_t}\right).$$

For such a collection of probabilities to define a stochastic process, we require that they meet a certain *consistency* criterion. We require the marginal distribution of the values taken by the process at any

collection of times to be the same under any finite dimensional distribution which includes the process at those time points, so, defining any second collection of times $j_1, \ldots, j_s$ with the property that $j_k \neq i_l$ for any $k \leq t, l \leq s$, we must have that:

$$\begin{aligned}&\mathbb{P}\left(\xi_{i_1} \in A_{i_1}, \xi_{i_2} \in A_{i_2}, \ldots, \xi_{i_t} \in A_{i_t}\right)\\ =&\mathbb{P}\left(\xi_{i_1} \in A_{i_1}, \xi_{i_2} \in A_{i_2}, \ldots, \xi_{i_t} \in A_{i_t}, \xi_{j_1} \in E, \ldots, \xi_{j_s} \in E\right).\end{aligned}$$

This is just an expression of the intuitive concept that any finite dimensional distribution which describes the process at the times of interest should provide the same description if we neglect any information it provides about the process at other times. Or, to put it another way, they must all be marginal distributions of *the same* distribution.

In the case of real-valued stochastic processes, in which $E = \mathbb{R}$, we may express this concept in terms of the joint distribution functions (the multivariate analogue of the distribution function). Defining the joint distribution functions according to:

$$F_{i_1,\ldots,i_t}(x_1, x_2, \ldots, x_t) = \mathbb{P}\left(\xi_{i_1} \leq x_1, \xi_{i_2} \leq x_2, \ldots, \xi_{i_t} \leq x_t\right),$$

our consistency requirement may now be expressed as:

$$F_{i_1,\ldots,i_t,j_1,\ldots,j_s}(x_1, x_2, \ldots, x_t, \infty, \ldots, \infty) = F_{i_1,\ldots,i_t}(x_1, x_2, \ldots, x_t).$$

As we can specify a stochastic process if we are able to specify its finite dimensional distributions, we might wonder how to specify these distributions. In the next two sections, we proceed to describe a class of stochastic processes which can be described constructively and whose finite dimensional distributions may be easily established. The *Markov processes* which we are about to introduce represent the most widely used class of stochastic processes, and the ones which will be of most interest in the context of Monte Carlo methods.

3.2 Discrete State Space Markov Chains

3.2.1 Basic Notions

We begin by turning our attention to the discrete state space case which is somewhat easier to deal with than the general case which will be of interest later. In the case of discrete state spaces, in which $|E|$ is either finite, or countably infinite, we can work with the actual probability of the process having a particular value at any time (you'll recall that in the case of continuous random variables more subtlety is generally required as the probability of any continuous random variable defined by a density (with respect to Lebesgue measure, in particular) taking any particular value is zero). This simplifies things considerably, and we can consider defining the distribution of the process of interest over the first t time points by employing the following decomposition:

$$\begin{aligned}&\mathbb{P}\left(\xi_1 = x_1, \xi_2 = x_2, \ldots, \xi_t = x_t\right)\\ =&\mathbb{P}\left(\xi_1 = x_1, \xi_2 = x_2, \ldots, \xi_{t-1} = x_{t-1}\right)\mathbb{P}\left(\xi_t = x_t | \xi_1 = x_1, \ldots, \xi_{t-1} = x_{t-1}\right).\end{aligned}$$

Looking at this decomposition, it's clear that we could construct all of the distributions of interest from an initial distribution from which ξ_1 is assumed to be drawn and then a sequence of conditional distributions for each t, leading us to the specification:

$$\mathbb{P}\left(\xi_1 = x_1, \xi_2 = x_2, \ldots, \xi_t = x_t\right) = \mathbb{P}\left(\xi_1 = x_1\right) \prod_{i=2}^{t} \mathbb{P}\left(\xi_i = x_i | \xi_1 = x_1, \ldots, \xi_{i-1} = x_{i-1}\right). \tag{3.1}$$

From this specification we can trivially construct all of the finite dimensional distributions using no more than the sum and product rules of probability.

So, we have a method for constructing finite distributional distributions for a discrete state space stochastic process, but it remains a little formal as the conditional distributions seem likely to become increasingly complex as the time index increases. The conditioning present in decomposition (3.1) is needed to capture any relationship between the distribution at time t and *any* previous time. In many situations of interest, we might expect interactions to exist on only a much shorter time-scale. Indeed, one could envisage a *memoryless* process in which the distribution of the state at time $t+1$ depends only upon its state at time t, ξ_t, regardless of the path by which it reached ξ_t. Formally, we could define such a process as:

$$\mathbb{P}\left(\xi_1 = x_1, \xi_2 = x_2, \ldots, \xi_t = x_t\right) = \mathbb{P}\left(\xi_1 = x_1\right) \prod_{i=2}^{t} \mathbb{P}\left(\xi_i = x_i | \xi_{i-1} = x_{i-1}\right). \tag{3.2}$$

It is clear that (3.2) is a particular case of (3.1) in which this lack of memory property is captured explicitly, as:

$$\mathbb{P}\left(\xi_t = x_t | \xi_1 = x_1, \ldots, \xi_{t-1} = x_{t-1}\right) = \mathbb{P}\left(\xi_t = x_t | \xi_{t-1} = x_{t-1}\right).$$

We will take this as the defining property of a collection of processes which we will refer to as discrete time *Markov processes* or, as they are more commonly termed in the Monte Carlo literature, *Markov chains*. There is some debate in the literature as to whether the term "Markov chain" should be reserved for those Markov processes which take place on a discrete state space, those which have a discrete index set (the only case we will consider here) or both. As is common in the field of Monte Carlo simulation, we will use the terms Markov chain and Markov process interchangeably.

When dealing with discrete state spaces, it is convenient to associate a row vector[1] with any probability distribution. We assume, without loss of generality, that the state space, E, is $\mathbb{N}$. Now, given a random variable X on E, we say that X has distribution μ, often written as $X \sim \mu$ for some vector μ with the property that:

$$\forall x \in E : \mathbb{P}\left(X = x\right) = \mu_x.$$

Homogeneous Markov Chains. The term *homogeneous Markov Chain* is used to describe a Markov process of the sort just described with the additional caveat that the conditional probabilities do not depend explicitly on the time index, so:

$$\forall m \in \mathbb{N} : \mathbb{P}\left(\xi_t = y | \xi_{t-1} = x\right) \equiv \mathbb{P}\left(\xi_{t+m} = y | \xi_{t+m-1} = x\right).$$

In this setting, it is particular convenient to define a function corresponding to the *transition probability* (as the probability distribution at time $t+1$ conditional upon the state of the process at time t) or *kernel* as it is often known, which may be written as a two argument function or, in the discrete case as a matrix, $K(i,j) = K_{ij} = \mathbb{P}\left(\xi_t = j | \xi_{t-1} = i\right)$.

Having so expressed things, we are able to describe the dynamic structure of a discrete state space, discrete time Markov chain in a particularly simple form. If we allow μ_t to describe the distribution of the chain at time t, so that $\mu_{t,i} = \mathbb{P}\left(\xi_t = i\right)$, then we have by applying the sum and product rules of probability, that:

[1] Formally, much of the time this will be an infinite dimensional vector but this need not concern us here.

$$\mu_{t+1,j} = \sum_i \mu_{t,i} K_{ij}.$$

Provided that μ_t is a *row* vector, We may recognise this as standard vector-matrix multiplication and write simply that $\mu_{t+1} = \mu_t K$ and, proceeding inductively it's straightforward to verify that $\mu_{t+m} = \mu_t K^m$ where K^m denotes the usual m^{th} matrix power of K. We will make some use of this object, as it characterises the m-step ahead condition distribution:

$$K_{ij}^m := (K^m)_{ij} = \mathbb{P}\left(\xi_{t+m} = j | \xi_t = i\right).$$

In fact, the initial distribution μ_1, together with K tells us the full distribution of the chain over any finite time horizon:

$$\mathbb{P}\left(\xi_1 = x_1, \ldots, \xi_t = x_t\right) = \mu_{1,x_1} \prod_{i=2}^{t} K_{x_{i-1}x_i}.$$

A discrete time stochastic processes is said to possess the *weak Markov property* if, for any deterministic time, t and any finite integer p, we may write that for any integrable function $\varphi : E \to \mathbb{R}$:

$$\mathbb{E}\left[\varphi(\xi_{t+p}) | \xi_1 = x_1, \ldots \xi_t = x_t\right] = \mathbb{E}\left[\varphi(\xi_{t+p}) | \xi_t = x_t\right].$$

Inhomogeneous Markov Chains. Note that it is perfectly possible to define Markov Chains whose behaviour does depend explicitly upon the time index. Although such processes are more complex to analyse than their homogeneous counterparts, they do play a rôle in Monte Carlo methodology – in both established algorithms such as simulated annealing (see Chapter 9) and in more recent developments such as adaptive Markov Chain Monte Carlo and the State Augmentation for Maximising Expectations (SAME) algorithm of Doucet et al. (2002a). In the interests of simplicity, what follows is presented for homogeneous Markov Chains.

Examples. Before moving on to introduce some theoretical properties of discrete state space Markov chains we will present a few simple examples. Whilst there are innumerable examples of homogeneous discrete state space Markov chains, we confined ourselves here to some particular simple cases which will be used to illustrate some properties below, and which will probably be familiar to you.

We begin with an example which is apparently simple, and rather well known, but which exhibits some interesting properties

Example 3.1 (the simple random walk over the integers). Given a process ξ_t whose value at time $t+1$ is $\xi_t + 1$ with probability p_+ and ξ_{t-1} with probability $p_- = 1 - p_+$, we obtain the familiar random walk. We may write this as a Markov chain by setting $E = \mathbb{Z}$ and noting that the transition kernel may be written as:

$$K_{ij} = \begin{cases} p_- & \text{if } j = i - 1 \\ p_+ & \text{if } j = i + 1 \\ 0 & \text{otherwise.} \end{cases}$$

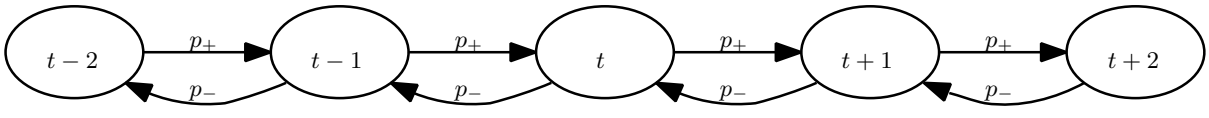


Figure 3.1. A simple random walk on $\mathbb{Z}$.

◁

Example 3.2. It will be interesting to look at a slight extension of this random walk, in which there is some probability p_0 of remaining in the present state at the next time step, so $p_+ + p_- < 1$ and $p_0 = 1 - (p_+ + p_-)$. In this case we may write the transition kernel as:

$$K_{ij} = \begin{cases} p_- & \text{if } j = i-1 \\ p_0 & \text{if } j = i \\ p_+ & \text{if } j = i+1 \\ 0 & \text{otherwise.} \end{cases}$$

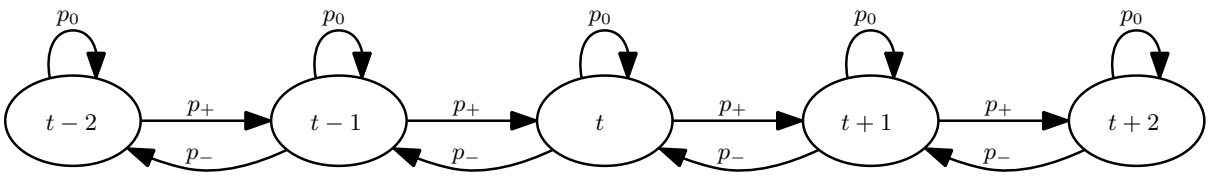


Figure 3.2. A random walk on $\mathbb{Z}$ with $K_{tt} > 0$.

◁

Example 3.3 (Random Walk on a Triangle). A third example which we will consider below could be termed a "random walk on a triangle". In this case, we set $E = \{1, 2, 3\}$ and define a transition kernel of the form:

$$K = \begin{bmatrix} 0 & p_+ & p_- \\ p_- & 0 & p_+ \\ p_+ & p_- & 0 \end{bmatrix}.$$

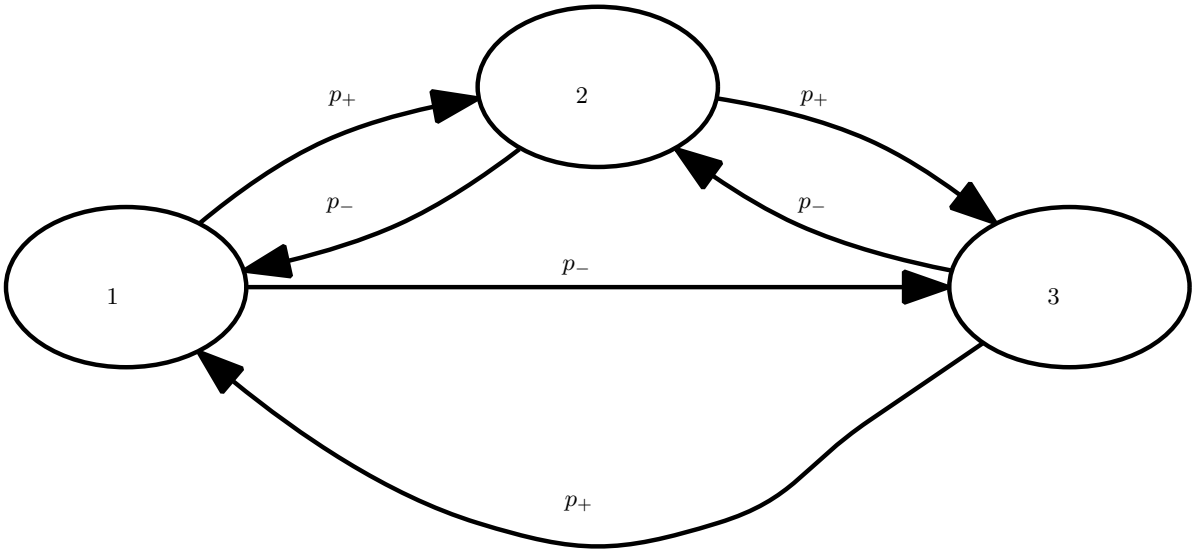


Figure 3.3. A random walk on a triangle.

◁

Example 3.4 (One-sided Random Walk). Finally, we consider the rather one-sided random walk on the positive integers, illustrated in figure 3.4, and defined by transition kernel:

$$K_{ij} = \begin{cases} p_0 & \text{if } j = i \\ p_+ = 1 - p_0 & \text{if } j = i + 1 \\ 0 & \text{otherwise.} \end{cases}$$

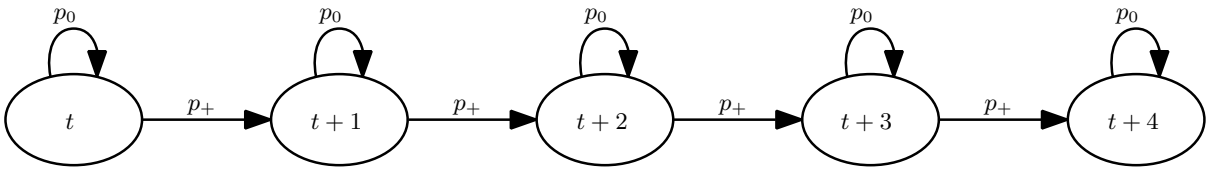


Figure 3.4. A random walk on the positive integers.

◁

3.2.2 Important Properties

In this section we introduce some important properties in the context of discrete state space Markov chains and attempt to illustrate their importance within the field of Monte Carlo simulation. As is the usual practice when dealing with this material, we will restrict our study to the homogeneous case. As you will notice, it is the transition kernel which is most important in characterising a Markov chain.

We begin by considering how the various states that a Markov chain may be reached from one another. In particular, the notion of states which *communicate* is at the heart of the study of Markov chains.

Definition 3.1 (Accessibility). *A state y is* accessible *from a state x, sometimes written as $x \to y$ if, for a discrete state space Markov chain,*

$$\inf \{t : \mathbb{P}(\xi_t = y | \xi_1 = x) > 0\} < \infty.$$

We can alternatively write this condition in terms of the transition matrix as $\inf \left\{t : K^t_{xy} > 0\right\} < \infty$.

This concept tells us which states one can reach at some finite time in the future, if one starts from a particular state and then moves, at each time, according to the transition kernel, K. That is, if $x \to y$, then there is a positive probability of reaching y at some finite time in the future, if we start from a state x and then "move" according to the Markov kernel K. It is now useful to consider cases in which one can traverse the entire space, or some subset of it, starting from any point.

Definition 3.2 (Communication). *Two states $x, y \in E$ are said to communicate (written, by some authors as $x \leftrightarrow y$) if each is accessible from the other, that is:*

$$x \leftrightarrow y \Leftrightarrow x \to y \text{ and } y \to x.$$

We're now in a position to describe the relationship, under the action of a Markov kernel, between two states. This allows us to characterise something known as the *communication structure* of the associated Markov chain to some degree, noting which points its possible to travel both to and back from. We now go on to introduce a concept which will allow us to describe the properties of the full state space, or significant parts of it, rather than individual states.

Definition 3.3 (Irreducibility). *A Markov Chain is said to be irreducible if* all states communicate*, so* $\forall x, y \in E : x \to y$*. Given a distribution* ϕ *on* E*, the term* ϕ*-irreducible is used to describe a Markov chain for which every state with positive probability under* ϕ *communicates with every other such state and is reachable from every point in the state space:*

$$\forall x \in E, y \in supp(\phi) : x \to y$$

where the support of the discrete distribution ϕ *is defined as* $supp(\phi) = \{y \in E : \phi(y) > 0\}$*.*

A Markov Chain is said to be strongly irreducible if any state can be reached from any point in the space in a single step and strongly ϕ*-irreducible if all states (except for a collection with probability 0 under* ϕ*) may be reached in a single step.*

This will prove to be important for the study of Monte Carlo methods based upon Markov chains as a chain with this property can somehow explore the entire space rather than being confined to some portion of it, perhaps one which depends upon the initial state.

It is also important to consider the type of routes which it is possible to take between a state, x, and itself as this will tell us something about the presence of long-range correlation between the states of the chain.

Definition 3.4 (Period). *A state* x *in a discrete state space Markov chain has period* $d(x)$ *defined as:*

$$d(x) = \gcd\left\{s \geq 1 : K^s_{xx} > 0\right\},$$

where gcd *denotes the* greatest common denominator*. A chain possessing such a state is said to have a* cycle of length d*.*

Proposition 3.1. *All states which communicate have the same period and hence, in an irreducible Markov chain, all states have the same period.*

Proof. Assume that $x \leftrightarrow y$. Let there exist paths of lengths r, s and t, respectively from $x \to y$, $y \to x$ and $y \to y$, respectively.

There are paths of length $r + s$ and $r + s + t$ from x to x, hence $d(x)$ must be a divisor of $r + s$ and $r + s + t$ and consequently of their difference, t. This holds for any t corresponding to a path from $y \to y$ and so $d(x)$ is a divisor of the length of any path from $y \to y$: as $d(y)$ is the greatest common divisor of all such paths, we have that $d(x) \leq d(y)$.

By symmetry, we also have that $d(y) \leq d(x)$, and this completes the proof. □

In the context of irreducible Markov chains, the term *periodic* is used to describe those chains whose states have some common period great than 1, whilst those chains whose period is 1 are termed *aperiodic*.

One further quantity needs to be characterised in order to study the Markov chains which will arise later. Some way of describing *how many times* a state is visited if a Markov chain is allowed to run for infinite time still seems required. In order to do this it is useful to define an additional random quantity, the number of times that a state is visited:

$$\eta_x := \sum_{k=1}^{\infty} \mathbb{I}_x(\xi_k).$$

We will also adopt the convention, common in the Markov chain literature that, given any function of the path of a Markov chain, φ, $\mathbb{E}_x[\varphi]$ is the expectation of that function under the law of the Markov chain initialised with $\xi_1 = x$. Similarly, if μ is some distribution over E, then $\mathbb{E}_\mu[\varphi]$ should be interpreted as the expectation of φ under the law of the process initialised with $\xi_1 \sim \mu$.

Definition 3.5 (Transience and Recurrence). *In the context of discrete state space Markov chains, we describe a state, x, as* transient *if:*

$$\mathbb{E}_x\left[\eta_x\right] < \infty$$

whilst, if we have that,

$$\mathbb{E}_x\left[\eta_x\right] = \infty,$$

then that state will be termed recurrent*.*

In the case of irreducible Markov chains, transience and recurrence are properties of the chain itself, rather than its individual states: if any state is transient (or recurrent) then all states have that property. Indeed, for an irreducible Markov chain either all states are recurrent or all are transient.

We will be particularly concerned in this course with Markov kernels which admit an invariant distribution.

Definition 3.6 (Invariant Distribution). *A distribution, μ is said to be* invariant *or* stationary *for a Markov kernel, K, if $\mu K = \mu$.*

If a Markov chain has any single time marginal distribution which corresponds to it's stationary distribution, $\xi_t \sim \mu$, then all of its future time marginals are the same as, $\xi_{t+s} \sim \mu K^s = \mu$. A Markov chain is said to be in its stationary regime once this has occurred. Note that this tells us nothing about the correlation between the states or their joint distribution. One can also think of the invariant distribution μ of a Markov kernel, K as the *left eigenvector* with unit eigenvalue.

Definition 3.7 (Reversibility). *A stationary stochastic process is said to be reversible if the statistics of the time-reversed version of the process match those of the process in the forward distribution, so that* reversing time *makes no discernible difference to the sequence of distributions which are obtained, that is the distribution of any collection of future states given any past history must match the conditional distribution of the past conditional upon the future being the reversal of that history.*

Reversibility is a condition which, if met, simplifies the analysis of Markov chains. It is normally verified by checking the detailed balance condition, (3.3). If this condition holds for a distribution, then it also tells us that this distribution is the stationary distribution of the chain, another property which we will be interested in.

Proposition 3.2. *If a Markov kernel satisfies the* detailed balance condition *for some distribution μ,*

$$\forall x, y \in E : \mu_x K_{xy} = \mu_y K_{yx} \tag{3.3}$$

then:

1. *μ is the invariant distribution of the chain.*
2. *The chain is reversible with respect to μ.*

Proof. To demonstrate that K is μ-invariant, consider summing both sides of the detailed balance equation over x:

$$\sum_{x \in E} \mu_x K_{xy} = \sum_{x \in E} \mu_y K_{yx}$$
$$(\mu K)_y = \mu_y,$$

and as this holds for all y, we have $\mu K = \mu$.

In order to verify that the chain is reversible we proceed directly:

$$\begin{aligned}\mathbb{P}\left(\xi_t = x | \xi_{t+1} = y\right) &= \frac{\mathbb{P}\left(\xi_t = x, \xi_{t+1} = y\right)}{\mathbb{P}\left(\xi_{t+1} = y\right)} \\ &= \frac{\mathbb{P}\left(\xi_t = x\right) K_{xy}}{\mathbb{P}\left(\xi_{t+1} = y\right)} \\ &= \frac{\mu_x K_{xy}}{\mu_y} = \frac{\mu_y K_{yx}}{\mu_y} \\ &= K_{yx} = \mathbb{P}\left(\xi_t = x | \xi_{t-1} = y\right),\end{aligned}$$

in the case of a Markov chain it is clear that if the transitions are time-reversible then the process must be time reversible. □

3.3 General State Space Markov Chains

3.3.1 Basic Concepts

The study of general state space Markov chains is a complex and intricate business. To do so entirely rigorously requires a degree of technical sophistication which lies somewhat outside the scope of this course. Here, we will content ourselves with explaining how the concepts introduced in the context of discrete state spaces in the previous section might be extended to continuous domains via the use of probability densities. We will not consider more complex cases – such as mixed continuous and discrete spaces, or distributions over uncountable spaces which may not be described by a density. Nor will we provide proofs of results for this case, but will provide suitable references for the interested reader.

Although the guiding principles are the same, the study of Markov chains with continuous state spaces requires considerably more subtlety as it is necessary to introduce concepts which correspond to those which we introduced in the discrete case, describe the same properties and are motivated by the same intuition but which remain meaningful when we are dealing with densities rather than probabilities. As always, the principle complication is that the probability of any random variable distributed according to a non-degenerate density on a continuous state space taking any particular value is formally zero.

We will begin by considering how to emulate the decomposition we used to define a Markov chain on a discrete state space, Equation (3.2), when E is a continuous state space. In this case, what we essentially require is that the probability of any range of possible values, given the entire history of the process depends only upon its most recent value in the sense that, for any measurable $A_t \subset E$:

$$\mathbb{P}\left(\xi_t \in A_t | \xi_1 = x_1, \ldots, \xi_{t-1} = x_{t-1}\right) = \mathbb{P}\left(\xi_t \in A_t | \xi_{t-1} = x_{t-1}\right).$$

In the case which we are considering, it is convenient to describe the distribution of a random variable over E in terms of some probability density, $\mu : E \to \mathbb{R}\$$ which has the property that, if integrated over any measurable set, it tells us the probability that the random variable in question lies within that set, i.e. if $X \sim \mu$, we have that for any measurable set A that:

$$\mathbb{P}\left(X \in A\right) = \int_A \mu(x) dx.$$

We will consider only the homogeneous case here, although the generalisation to inhomogeneous Markov chains follows in the continuous setting in precisely the same manner as the discrete one. In this context, we may describe the conditional probabilities of interest as a function $K : E \times E \to \mathbb{R}$ which has the property that for all measurable sets $A \subset E$ and all points $x \in E$:

$$\mathbb{P}\left(\xi_t \in A | \xi_{t-1} = x\right) = \int_A K(x, y)dy.$$

We note that, as in the discrete case the law of a Markov chain evaluated at any finite number of points may be completely specified by the initial distribution, call it μ, and a transition kernel, K. We have, for any suitable collection of sets $A_1, \ldots$, that the following holds:

$$\mathbb{P}\left(\xi_1 \in A_1, \ldots, \xi_t \in A_t\right) = \int_{A_1 \times \cdots \times A_t} \mu(x_1) \prod_{k=2}^{t} K(x_{k-1}, x_k) dx_1 \ldots dx_t.$$

And, again, it is useful to be able to consider the s-step ahead conditional distributions,

$$\mathbb{P}\left(\xi_{t+s} \in A | \xi_t = x_t\right) = \int_{E^{s-1} \times A} \prod_{k=t+1}^{k=t+s} K(x_{k-1}, x_k) dx_{t+1} \ldots dx_{t+s},$$

and it is useful to define an s-step ahead transition kernel in the same manner as it is in the discrete case, here matrix multiplication is replaced by a convolution operation but the intuition remains the same. Defining

$$K^s(x_t, x_{t+s}) := \int_{E^{s-1}} \prod_{k=t+1}^{k=t+s} K(x_{k-1}, x_k) dx_{t+1} \ldots dx_{t+s-1},$$

we are able to write

$$\mathbb{P}\left(\xi_{t+s} \in A | \xi_t = x_t\right) = \int_A K^s(x_t, x_{t+s}) dx_{t+s}.$$

3.3.2 Important Properties

In this section we will introduce properties which fulfill the same rôle in context of continuous state spaces as those introduced in section 3.2.2 do in the discrete setting.

Whilst it is possible to define concepts similar to communication and accessibility in a continuous state space context, this isn't especially productive. We are more interested in the property of *irreducibility*: we want some way of determining what class of states are reachable from one another and hence what part of E might be explored, with positive probability, starting from a point within such a class. We will proceed directly to a continuous state space definition of this concept.

Definition 3.8 (Irreducibility). *Given a distribution, μ, over E, a Markov chain is said to be* μ-irreducible *if for all points $x \in E$ and all measurable sets A such that $\mu(A) > 0$ there exists some t such that:*

$$\int_A K^t(x, y)dy > 0.$$

If this condition holds with $t = 1$, then the chain is said to be strongly μ-irreducible.

This definition has the same character as that employed in the discrete case, previously, but is well defined for more general state spaces. It still tells us whether a chain is likely to be satisfactory if we are interested in approximation of some property of a measure μ by using a sample of the evolution of that chain: if it is *not* μ-irreducible then there are some points in the space from which we cannot reach all of the support of μ, and this is likely to be a problem. In the sequel we will be interested more of less exclusively with Markov chains which are irreducible with respect to some measure of interest.

We need a little more subtlety in extending some of the concepts introduced in the case of discrete Markov chains to the present context. In order to do this, it will be necessary to introduce the concept of

the *small set*; these function as a replacement for the individual states of a discrete space Markov chain as we will see shortly.

A first attempt might be to consider the following sets which have the property that the distribution of taken by the Markov chain at time $t+1$ is the same if it starts at any point in this set – so the conditional distribution function of the next state is the same for every starting point within this set.

Definition 3.9 (Atoms). *A Markov chain with transition kernel K is said to have an atom, $\alpha \subset E$, if there is some probability distribution, ν, such that:*

$$\forall x \in \alpha, A \subset E : \quad \int_A K(x,y)dy = \int_A \nu(y)dy.$$

If the Markov chain in question is ν-irreducible, then α is termed an accessible atom.

Whilst the concept of *atoms* starts to allow us to introduce some sort of structure similar to that seen in discrete chains — it provides us with a set of positive probability which, if the chain ever enters it, we know the distribution of the subsequent state — most interesting Markov chains on continuous state spaces do not possess atoms. Note that this condition is much stronger than requiring that the transition density, K, exists as in general all points in the space have zero probability. The condition that the distribution of the next state is precisely the same, whatever the current state within a set of positive probabilty is indeed rather strong. Another approach would be to require only that the conditional distribution has a *component* which is common to all starting points within a set of positive probability, and that is the intuition behind a much more useful concept which underlies much of the analysis of general state space Markov chains.

Definition 3.10 (Small Sets). *A set, $C \subset E$, is termed* small *for a given Markov chain (or, when one is being precise, (ν, s, ϵ)-small) if there exists some positive integer s, some $\epsilon > 0$ and some non-trivial probability distribution, ν, such that:*

$$\forall x \in C, A \subset E : \quad \int_A K^s(x,y)dy \geq \epsilon \int_A \nu(y)dy.$$

This tells us that the distribution s-steps after the chain enters the small set has a component of size at least ϵ of the distribution ν, wherever it was within that set. In this sense, small sets are not "too big": there is potentially some commonality of all paths emerging from them. Although we have not proved that such sets exist for any particular class of Markov chains it is, in fact, the case that they do for many interesting Markov chain classes and their existence allows a number of sophisticated techniques to be applied

In order to define cycles (and hence the notion of periodicity) in the general case, we require the existence of a small set. We need some group of "sufficiently similar" points in the state space which together have a strictly positive probability of being reached. We then treat this collection of points in the same manner as an individual state in the discrete case, leading to the following definitions. In the case of real-valued random variables with continuous densities, such sets must contain uncountably many points (as any set containing finitely- or countably-many points has probability zero in such a setting) and so we need to be careful when we define sufficiently similar to ensure that such sets exist and that the similarity is of a sort which we can use.

Definition 3.11 (Cycles). *A μ-irreducible Markov chain has a* cycle of length d *if there exists a (ν, M, ϵ)-small set C, for some integer M, some $\epsilon > 0$ and some probability distribution ν for which $\int_C \nu(x)dx > 0$, such that:*

$$d = \gcd\{s \in \mathbb{N} : C \text{ is } (\nu, s, \delta_s\epsilon)\text{-small for some } \delta_s > 0\}.$$

This provides a reasonable concept of periodicity within a general state space Markov chain as it gives us a way of characterising the existence of regions of the space with the property that, wherever you start within that region you have positive probability of returning to that set after any multiple of d steps and this *does not* hold for any number of steps which is not a multiple of d. We are able to define periodicity and aperiodicity in the same manner as for discrete chains, but using this definition of a cycle. As in the discrete space, all states within the support of μ in a μ-irreducible chain must have the same period (see Proposition 3.1) although we will not prove this here.

Considering periodicity from a different viewpoint, we are able to characterise it in a manner which is rather easier to interpret but somewhat difficult to verify in practice. The following definition of period is equivalent to that given above (Nummelin, 1984): a Markov chain has a period d if there exists some partition of the state space, $E_1, \ldots, E_d$ with the properties that:

- $\forall i \neq j : E_i \cap E_j = \emptyset$
- $\bigcup_{i=1}^{d} E_i = E$
-
$$\forall i, j, t, s : \mathbb{P}\left(X_{t+s} \in E_j | X_t \in E_i\right) = \begin{cases} 1 & j = i + s \bmod d \\ 0 & \text{otherwise.} \end{cases}$$

What this actually tells us is that a Markov chain with a period of d has associated with it a disjoint partition of the state space, $E_1, \ldots, E_d$ and that we know that the chain moves with probability 1 from set E_1 to E_2, E_2 to E_3, E_{d-1} to E_d and E_d to E_1 (assuming that $d \geq 3$, of course). Hence the chain will visit a particular element of the partition with a period of d.

We also require some way of characterising how often a continuous state space Markov chain visits any particular region of the state space in order to obtain concepts analogous to those of transience and recurrence in the discrete setting. In order to do this we define a collection of random variables η_A for any subset A of E, which correspond to the number of times the set A is visited, i.e. $\eta_A := \sum_{l=1}^{\infty} \mathbb{I}_A(\xi_k)$ and, once again we use $\mathbb{E}_x$ to denote the expectation under the law of the Markov chain with initial state x. We note that if a chain is not μ-irreducible for some distribution μ, then there is no guarantee that it is either transient or recurrent, however, the following definitions do hold:

Definition 3.12 (Transience and Recurrence). *We begin by defining* uniform transience *and* recurrence *for sets $A \subset E$ for μ-irreducible general state space Markov chains. Such a set is recurrent if:*

$$\forall x \in A : \mathbb{E}_x\left[\eta_A\right] = \infty.$$

A set is uniformly transient *if there exists some $M < \infty$ such that:*

$$\forall x \in A : \quad \mathbb{E}_x\left[\eta_A\right] \leq M.$$

The weaker concept of transience of a set may then be introduced. A set, $A \subset E$, is transient *if it may be expressed as a countable union of uniformly transient sets, i.e.:*

$$\exists \{B_i \subset E\}_{i=1}^{\infty} : \quad A \subset \bigcup_{i=1}^{\infty} B_i$$
$$\forall i \in \mathbb{N} : \quad \forall x \in B_i : \quad \mathbb{E}_x\left[\eta_{B_i}\right] \leq M_i < \infty.$$

A general state space Markov chain is recurrent *if the following two conditions are satisfied:*

– *The chain is μ-irreducible for some distribution μ.*
– *For every measurable set $A \subset E$ such that $\int_A \mu(y)dy > 0$, $\mathbb{E}_x[\eta_A] = \infty$ for every $x \in A$.*

whilst it is transient *if it is μ-irreducible for some distribution μ and the entire space is transient.*

As in the discrete setting, in the case of irreducible chains, transience and recurrence are properties of the chain rather than individual states: all states within the support of the irreducibility distribution are either transient or recurrent. It is useful to note that any μ-irreducible Markov chain which has stationary distribution μ is positive recurrent (Tierney, 1994).

A slightly stronger form of recurrence is widely employed in the proof of many theoretical results which underlie many applications of Markov chains to statistical problems, this form of recurrence is known as Harris recurrence and may be defined as follows:

Definition 3.13 (Harris Recurrence). *A set $A \subset E$ is* Harris recurrent *if $\mathbb{P}_x(\eta_A = \infty) = 1$ for every $x \in A$.*

A Markov chain is Harris recurrent *if there exists some distribution μ with respect to which it is irreducible and every set A such that $\int_A \mu(x)dx > 0$ is Harris recurrent.*

The concepts of invariant distribution, reversibility and detailed balance are essentially unchanged from the discrete setting. It's necessary to consider integrals with respect to densities rather than sums over probability distributions, but no fundamental differences arise here.

3.4 Selected Theoretical Results

The probabilistic study of Markov chains dates back more than fifty years and comprises an enormous literature, much of it rather technically sophisticated. We don't intend to summarise that literature here, nor to provide proofs of the results which we present here. This section serves only to motivate the material presented in the subsequent chapters.

These two theorems fill the rôle which the law of large numbers and the central limit theorem for independent, identically distributed random variables fill in the case of simple Monte Carlo methods. They tell us, roughly speaking, that if we take the sample averages of a function at the points of a Markov chain which satisfies suitable regularity conditions and possesses the correct invariant distribution, then we have convergence of those averages to the integral of the function of interest under the invariant distribution and, furthermore, under stronger regularity conditions we can obtain a rate of convergence.

There are two levels of strength of law of large numbers which it is useful to be aware of. The first tells us that for most starting points of the chain a law of large numbers will hold. Under slightly stronger conditions (which it may be difficult to verify in practice) it is possible to show the same result holds for *all* starting points.

Theorem 3.1 (A Simple Ergodic Theorem). *If $(\xi_i)_{i\in\mathbb{N}}$ is a μ-irreducible, recurrent $\mathbb{R}^d$-valued Markov chain which admits μ as a stationary distribution, then the following strong law of large numbers holds (convergence is with probability 1) for any integrable function $\varphi : \mathbb{R}^d \to \mathbb{R}$:*

$$\lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} \varphi(\xi_i) \to \int \varphi(x)\mu(x)dx.$$

for almost every starting value x. That is, for any x except perhaps for some set $\mathcal{N}$ which has the property that $\int_{\mathcal{N}} \mu(x)dx = 0$.

An outline of the proof of this theorem is provided by (Roberts and Rosenthal, 2004, Fact 5.).

Theorem 3.2 (A Stronger Ergodic Theorem). *If $(\xi_i)_{i\in\mathbb{N}}$ is a μ-invariant, Harris recurrent Markov chain, then the following strong law of large numbers holds (convergence is with probability 1) for any integrable function $\varphi : E \to \mathbb{R}$:*

$$\lim_{t\to\infty} \frac{1}{t}\sum_{i=1}^{t} \varphi(\xi_i) \to \int \varphi(x)\mu(x)dx.$$

A proof of this result is beyond the scope of the course. This is a particular case of (Robert and Casella, 2004, p. 241, Theorem 6.63), and a proof of the general theorem is given there. The same theorem is also presented with proof in (Meyn and Tweedie, 1993, p. 433, Theorem 17.3.2).

Theorem 3.3 (A Central Limit Theorem). *Under technical regularity conditions (see (Jones, 2004) for a summary of various combinations of conditions) it is possible to obtain a central limit theorem for the ergodic averages of a Harris recurrent, μ-invariant Markov chain, and a function $\varphi : E \to \mathbb{R}$ which has at least two finite moments (depending upon the combination of regularity conditions assumed, it may be necessary to have a finite moment of order $2 + \delta$).*

$$\lim_{t\to\infty} \sqrt{t}\left[\frac{1}{t}\sum_{i=1}^{t} \varphi(\xi_i) - \int \varphi(x)\mu(x)dx\right] \xrightarrow{d} \mathsf{N}(0, \sigma^2(\varphi)),$$

$$\sigma^2(\varphi) = \mathbb{E}\left[(f(\xi_1) - \bar{\varphi})^2\right] + 2\sum_{k=2}^{\infty} \mathbb{E}\left[(\varphi(\xi_1) - \bar{\varphi})(\varphi(\xi_k) - \bar{\varphi})\right],$$

where $\bar{\varphi} = \int \varphi(x)\mu(x)dx$.

3.5 Further Reading

We conclude this chapter by noting that innumerable tutorials on the subject of Markov chains have been written, particularly with reference to their use in the field of Monte Carlo simulation. Some which might be of interest include the following:

- (Roberts, 1996) provides an elementary introduction to some Markov chain concepts required to understand their use in Monte Carlo algorithms.
- In the same volume, (Tierney, 1996) provides a more technical look at the same concepts; a more in-depth, but similar approach is taken by the earlier paper Tierney (1994).
- An alternative, elementary formulation of some of the material presented here together with some additional background material, aimed at an engineering audience, can be found in Johansen (2009).
- (Robert and Casella, 2004, chapter 6). This is a reasonably theoretical treatment intended for those interest in Markov chain Monte Carlo; it is reasonably technical in content, without dwelling on proofs. Those familiar with measure theoretic probability might find this a reasonably convenient place to start.
- Those of you interested in technical details might like to consult (Meyn and Tweedie, 1993). This is the definitive reference work on stability, convergence and theoretical analysis of Markov chains and it is now possible to download it, free of charge from the website of one of the authors.
- A less detailed, but more general and equally rigorous, look at Markov chains is provided by the seminal work of (Nummelin, 1984). This covers some material outside of the field of probability, but remains a concise work and presents only a few of the simpler results. It is perhaps a less intimidating starting point than (Meyn and Tweedie, 1993), although opinions on this vary.

– A recent survey of theoretical results relevant to Monte Carlo is provided by (Roberts and Rosenthal, 2004). Again, this is necessarily somewhat technical.

4. The Gibbs Sampler

4.1 Introduction

In Section 2.3 we saw that, by using importance sampling, we can approximate an expectation $\mathbb{E}_f(\varphi(X))$ without having to sample directly from f. However, finding an instrumental distribution which allows us to *efficiently* estimate $\mathbb{E}_f(\varphi(X))$ can be difficult, especially in large dimensions.

In this chapter and the following chapters we will use a somewhat different approach. We will discuss methods that allow us to approximate expectations with respect to f without having to sample from f directly. More mathematically speaking, we will discuss methods which generate a Markov chain whose stationary distribution is the distribution of interest f. Such methods are often referred to as Markov Chain Monte Carlo (MCMC) methods.

We begin with a motivating example before looking at a general algorithm which arises from similar considerations to those seen in the example.

Example 4.1 (Poisson change point model). Assume the following Poisson model of two regimes for n random variables $Y_1, \ldots, Y_n$.

$$\begin{aligned} Y_i &\sim \text{Poi}(\lambda_1) \quad \text{for} \quad i = 1, \ldots, M \\ Y_i &\sim \text{Poi}(\lambda_2) \quad \text{for} \quad i = M+1, \ldots, n \end{aligned}$$

(Recall that the probability distribution function of the Poi(λ) distribution is $p(y) = \frac{\exp(-\lambda)\lambda^y}{y!}$.)

A suitable (conjugate) prior distribution for λ_j is the Gamma(α_j, β_j) distribution with density

$$f(\lambda_j) = \frac{1}{\Gamma(\alpha_j)} \lambda_j^{\alpha_j - 1} \beta_j^{\alpha_j} \exp(-\beta_j \lambda_j).$$

The joint distribution of $Y_1, \ldots, Y_n$, λ_1, λ_2, and M is

$$\begin{aligned} f(y_1, \ldots, y_n, \lambda_1, \lambda_2, M) &= \left(\prod_{i=1}^{M} \frac{\exp(-\lambda_1)\lambda_1^{y_i}}{y_i!} \right) \cdot \left(\prod_{i=M+1}^{n} \frac{\exp(-\lambda_2)\lambda_2^{y_i}}{y_i!} \right) \\ &\quad \cdot \frac{1}{\Gamma(\alpha_1)} \lambda_1^{\alpha_1 - 1} \beta_1^{\alpha_1} \exp(-\beta_1 \lambda_1) \cdot \frac{1}{\Gamma(\alpha_2)} \lambda_2^{\alpha_2 - 1} \beta_2^{\alpha_2} \exp(-\beta_2 \lambda_2). \end{aligned}$$

If M is known, the posterior distribution of λ_1 has the density

$$f(\lambda_1 | Y_1, \ldots, Y_n, M) \propto \lambda_1^{\alpha_1 - 1 + \sum_{i=1}^{M} y_i} \exp(-(\beta_1 + M)\lambda_1),$$

so

$$\lambda_1|Y_1,\dots Y_n,M \sim \text{Gamma}\left(\alpha_1+\sum_{i=1}^{M} y_i, \beta_1+M\right) \tag{4.1}$$

$$\lambda_2|Y_1,\dots Y_n,M \sim \text{Gamma}\left(\alpha_2+\sum_{i=M+1}^{n} y_i, \beta_2+n-M\right). \tag{4.2}$$

Now assume that we do not know the change point M and that we assume a uniform prior on the set $\{1,\dots,n-1\}$. It is easy to compute the distribution of M given the observations $Y_1,\dots Y_n$, and λ_1 and λ_2. It is a discrete distribution with probability mass function proportional to

$$p(M|\lambda_1,\lambda_2,y_1,\dots,y_n) \propto \lambda_1^{\sum_{i=1}^{M} y_i} \cdot \lambda_2^{\sum_{i=M+1}^{n} y_i} \cdot \exp((\lambda_2-\lambda_1)\cdot M) \tag{4.3}$$

The conditional distributions in (4.1) to (4.3) are all easy to sample from. It is however rather difficult to sample from the joint posterior of (λ_1,λ_2,M). ◁

The example above suggests the strategy of alternately sampling from the (full) conditional distributions ((4.1) to (4.3) in the example). This tentative strategy however raises some questions.

– Is the joint distribution uniquely specified by the conditional distributions?
– Sampling alternately from the conditional distributions yields a Markov chain: the newly proposed values only depend on the present values, not the past values. Will this approach yield a Markov chain with the correct invariant distribution? Will the Markov chain converge to the invariant distribution?

As we will see in Sections 4.3 and 4.4, the answer to both questions is — under certain conditions — yes. First, however, the next section provides an explicit statement of the Gibbs sampling algorithm.

4.2 Algorithm

The Gibbs sampler was first proposed by Geman and Geman (1984) and further developed by Gelfand and Smith (1990). It will be convenient to use the notation $x_{-i} := (x_1,\dots,x_{i-1},x_{i+1},\dots,x_p)$, that is whenever a vector is indexed with a negative number, $-i$, we mean all components of that vector *except* i.

Algorithm 4.1 ((Systematic scan) Gibbs sampler). Starting with $(X_1^{(0)},\dots,X_p^{(0)})$ iterate for $t=1,2,\dots$

1. Draw $X_1^{(t)} \sim f_{X_1|X_{-1}}(\cdot|X_2^{(t-1)},\dots,X_p^{(t-1)})$.

...

j. Draw $X_j^{(t)} \sim f_{X_j|X_{-j}}(\cdot|X_1^{(t)},\dots,X_{j-1}^{(t)},X_{j+1}^{(t-1)},\dots,X_p^{(t-1)})$.

...

p. Draw $X_p^{(t)} \sim f_{X_p|X_{-p}}(\cdot|X_1^{(t)},\dots,X_{p-1}^{(t)})$.

Figure 4.1 illustrates the Gibbs sampler. The conditional distributions used in the Gibbs sampler are often referred to as *full conditionals* (being conditional upon everything except the variable being sampled at each step). Note that this systematic scan Gibbs sampler is *not* reversible. Liu et al. (1995) proposed the following algorithm that yields a reversible chain.

Algorithm 4.2 (Random scan Gibbs sampler). Starting with $(X_1^{(0)},\dots,X_p^{(0)})$ iterate for $t=1,2,\dots$

1. Draw an index j from a distribution on $\{1,\dots,p\}$ (e.g. uniform)
2. Draw $X_j^{(t)} \sim f_{X_j|X_{-j}}(\cdot|X_1^{(t-1)},\dots,X_{j-1}^{(t-1)},X_{j+1}^{(t-1)},\dots,X_p^{(t-1)})$, and set $X_\iota^{(t)} := X_\iota^{(t-1)}$ for all $\iota \neq j$.

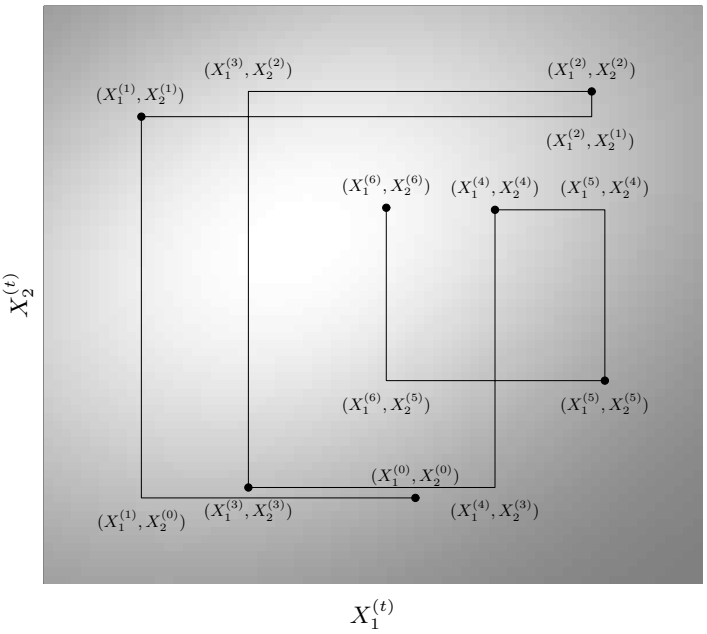


Figure 4.1. Illustration of the Gibbs sampler for a two-dimensional distribution

4.3 The Hammersley-Clifford Theorem

An interesting property of the full conditionals, upon which the Gibbs sampler is based, is that they fully specify the joint distribution, as Hammersley and Clifford proved in 1970 (Hammersley and Clifford actually never published this result, as they could not extend the theorem to the case of non-positivity.). Note that the set of marginal distributions does *not* have this property.

Definition 4.1 (Positivity condition). *A distribution with density $f(x_1, \ldots, x_p)$ and marginal densities $f_{X_i}(x_i)$ is said to satisfy the positivity condition if $f(x_1, \ldots, x_p) > 0$ for all $x_1, \ldots, x_p$ with $f_{X_i}(x_i) > 0$.*

The positivity condition thus implies that the support of the joint density f is the Cartesian product of the supports of the marginals f_{X_i}.

Theorem 4.1 (Hammersley-Clifford). *Let $(X_1, \ldots, X_p)$ satisfy the positivity condition and have joint density $f(x_1, \ldots, x_p)$. Then for all $(\xi_1, \ldots, \xi_p) \in supp(f)$*

$$f(x_1, \ldots, x_p) \propto \prod_{j=1}^{p} \frac{f_{X_j|X_{-j}}(x_j|x_1, \ldots, x_{j-1}, \xi_{j+1}, \ldots, \xi_p)}{f_{X_j|X_{-j}}(\xi_j|x_1, \ldots, x_{j-1}, \xi_{j+1}, \ldots, \xi_p)}$$

Proof. We have

$$f(x_1, \ldots, x_{p-1}, x_p) = f_{X_p|X_{-p}}(x_p|x_1, \ldots, x_{p-1}) f(x_1, \ldots, x_{p-1}) \tag{4.4}$$

and by exactly the same argument

$$f(x_1, \ldots, x_{p-1}, \xi_p) = f_{X_p|X_{-p}}(\xi_p|x_1, \ldots, x_{p-1}) f(x_1, \ldots, x_{p-1}), \tag{4.5}$$

thus

$$
\begin{aligned}
f(x_1,\ldots,x_p) &\overset{(4.4)}{=} \underbrace{f(x_1,\ldots,x_{p-1})}_{\overset{(4.5)}{=} f(x_1,\ldots,x_{p-1},\xi_p)/f_{X_p|X_{-p}}(\xi_p|x_1,\ldots,x_{p-1})} f_{X_p|X_{-p}}(x_p|x_1,\ldots,x_{p-1}) \\
&= f(x_1,\ldots,x_{p-1},\xi_p)\frac{f_{X_p|X_{-p}}(x_p|x_1,\ldots,x_{p-1})}{f_{X_p|X_{-p}}(\xi_p|x_1,\ldots,x_{p-1})} \\
&= \ldots \\
&= f(\xi_1,\ldots,\xi_p)\frac{f_{X_1|X_{-1}}(x_1|\xi_2,\ldots,\xi_p)}{f_{X_1|X_{-1}}(\xi_1|\xi_2,\ldots,\xi_p)}\cdots\frac{f_{X_p|X_{-p}}(x_p|x_1,\ldots,x_{p-1})}{f_{X_p|X_{-p}}(\xi_p|x_1,\ldots,x_{p-1})}
\end{aligned}
$$

The positivity condition guarantees that the conditional densities are non-zero. □

Note that the Hammersley-Clifford theorem does *not* guarantee the existence of a joint probability distribution for every choice of conditionals, as the following example shows. In Bayesian modeling such problems arise most often when using improper prior distributions.

Example 4.2. Consider the following "model"

$$
\begin{aligned}
X_1|X_2 &\sim \text{Expo}(\lambda X_2) \\
X_2|X_1 &\sim \text{Expo}(\lambda X_1),
\end{aligned}
$$

for which it would be easy to design a Gibbs sampler. Trying to apply the Hammersley-Clifford theorem, we obtain

$$
f(x_1,x_2) \propto \frac{f_{X_1|X_2}(x_1|\xi_2)\cdot f_{X_2|X_1}(x_2|x_1)}{f_{X_1|X_2}(\xi_1|\xi_2)\cdot f_{X_2|X_1}(\xi_2|x_1)} = \frac{\lambda\xi_2\exp(-\lambda x_1\xi_2)\cdot\lambda x_1\exp(-\lambda x_1 x_2)}{\lambda\xi_2\exp(-\lambda\xi_1\xi_2)\cdot\lambda x_1\exp(-\lambda x_1\xi_2)} \propto \exp(-\lambda x_1 x_2)
$$

The integral $\int\int\exp(-\lambda x_1 x_2)\ dx_1\ dx_2$ however is not finite, thus there is no two-dimensional probability distribution with $f(x_1,x_2)$ as its density. ◁

4.4 Convergence of the Gibbs sampler

First of all we have to establish whether the joint distribution $f(x_1,\ldots,x_p)$ is indeed the stationary distribution of the Markov chain generated by the Gibbs sampler. All the results in this section will be derived for the systematic scan Gibbs sampler (Algorithm 4.1). Very similar results hold for the random scan Gibbs sampler (Algorithm 4.2).

To proceed with such an analysis, we first have to determine the transition kernel corresponding to the Gibbs sampler.

Lemma 4.1. *The transition kernel of the Gibbs sampler is*

$$
\begin{aligned}
K(\mathbf{x}^{(t-1)},\mathbf{x}^{(t)}) &= f_{X_1|X_{-1}}(x_1^{(t)}|x_2^{(t-1)},\ldots,x_p^{(t-1)})\cdot f_{X_2|X_{-2}}(x_2^{(t)}|x_1^{(t)},x_3^{(t-1)},\ldots,x_p^{(t-1)})\cdot\ldots \\
&\quad\cdot f_{X_p|X_{-p}}(x_p^{(t)}|x_1^{(t)},\ldots,x_{p-1}^{(t)})
\end{aligned}
$$

Proof. We have, for any (measurable $\mathcal{X}$):

$$
\begin{aligned}
&\mathbb{P}(\mathbf{X}^{(t)}\in\mathcal{X}|\mathbf{X}^{(t-1)}=\mathbf{x}^{(t-1)}) = \int_{\mathcal{X}} f_{(\mathbf{X}^t|\mathbf{X}^{(t-1)})}(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})\ d\mathbf{x}^{(t)} \\
= \int_{\mathcal{X}} &\underbrace{f_{X_1|X_{-1}}(x_1^{(t)}|x_2^{(t-1)},\ldots,x_p^{(t-1)})}_{\text{corresponds to step 1. of the algorithm}}\cdot\underbrace{f_{X_2|X_{-2}}(x_2^{(t)}|x_1^{(t)},x_3^{(t-1)},\ldots,x_p^{(t-1)})}_{\text{corresponds to step 2. of the algorithm}}\cdot\ldots \\
&\cdot\underbrace{f_{X_p|X_{-p}}(x_p^{(t)}|x_1^{(t)},\ldots,x_{p-1}^{(t)})}_{\text{corresponds to step p. of the algorithm}}\ d\mathbf{x}^{(t)}\ \square
\end{aligned}
$$

□

Proposition 4.1. *The joint distribution $f(x_1, \ldots, x_p)$ is indeed the invariant distribution of the Markov chain $(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots)$ generated by the Gibbs sampler.*

Proof. Assume that $\mathbf{X}^{(t-1)} \sim f$, then

$$
\begin{aligned}
\mathbb{P}(\mathbf{X}^{(t)} \in \mathcal{X}) &= \int_{\mathcal{X}} \int f(\mathbf{x}^{(t-1)}) K(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}) \, d\mathbf{x}^{(t-1)} \, d\mathbf{x}^{(t)} \\
&= \int_{\mathcal{X}} \int \cdots \underbrace{\underbrace{\int f(x_1^{(t-1)}, \ldots, x_p^{(t-1)}) \, dx_1^{(t-1)}}_{=f(x_2^{(t-1)}, \ldots, x_p^{(t-1)})} f_{X_1|X_{-1}}(x_1^{(t)} | x_2^{(t-1)}, \ldots, x_p^{(t-1)}) \cdots f_{X_p|X_{-p}}(x_p^{(t)} | x_1^{(t)}, \ldots, x_{p-1}^{(t)}) dx_2^{(t-1)} \ldots dx_p^{(t-1)}}_{=f(x_1^{(t)}, x_2^{(t-1)}, \ldots, x_p^{(t-1)})} d\mathbf{x}^{(t)} \\
&= \int_{\mathcal{X}} \int \cdots \underbrace{\underbrace{\int f(x_1^{(t)}, x_2^{(t-1)}, \ldots, x_p^{(t-1)}) \, dx_2^{(t-1)}}_{=f(x_1^{(t)}, x_3^{(t-1)}, \ldots, x_p^{(t-1)})} f_{X_2|X_{-2}}(x_2^{(t)} | x_1^{(t)}, x_3^{(t-1)}, \ldots, x_p^{(t-1)}) \cdots f_{X_p|X_{-p}}(x_p^{(t)} | x_1^{(t)}, \ldots, x_{p-1}^{(t)}) dx_3^{(t-1)} \ldots dx_p^{(t-1)}}_{=f(x_1^{(t)}, x_2^{(t)}, x_3^{(t-1)}, \ldots, x_p^{(t-1)})} d\mathbf{x}^{(t)} \\
&= \ldots \\
&= \int_{\mathcal{X}} \underbrace{\underbrace{\int f(x_1^{(t)}, \ldots, x_{p-1}^{(t)}, x_p^{(t-1)}) \, dx_p^{(t-1)}}_{=f(x_1^{(t)}, \ldots, x_{p-1}^{(t)})} f_{X_p|X_{-p}}(x_p^{(t)} | x_1^{(t)}, \ldots, x_{p-1}^{(t)}) \, d\mathbf{x}^{(t)}}_{=f(x_1^{(t)}, \ldots, x_p^{(t)})} \\
&= \int_{\mathcal{X}} f(x_1^{(t)}, \ldots, x_p^{(t)}) \, d\mathbf{x}^{(t)}
\end{aligned}
$$

Thus f is the density of $\mathbf{X}^{(t)}$ (if $\mathbf{X}^{(t-1)} \sim f$). □

So far we have established that f is indeed the invariant distribution of the Gibbs sampler. Next, we have to establish conditions under which the Markov chain generated by the Gibbs sampler will converge to f.

First of all we have to study under which conditions the resulting Markov chain is irreducible (really, we mean f-irreducible, of course, here and in the following we understand by "irreducibilty" irreducibility with respect to the target distribution f). The following example shows that this does not need to be the case.

Example 4.3 (Reducible Gibbs sampler). Consider Gibbs sampling from the uniform distribution on $C_1 \cup C_2$ with $C_1 := \{(x_1, x_2) : \|(x_1, x_2) - (1, 1)\| \leq 1\}$ and $C_2 := \{(x_1, x_2) : \|(x_1, x_2) - (-1, -1)\| \leq 1\}$, i.e.

$$f(x_1, x_2) = \frac{1}{2\pi} \mathbb{I}_{C_1 \cup C_2}(x_1, x_2)$$

Figure 4.2 shows the density as well the first few samples obtained by starting a Gibbs sampler with $X_1^{(0)} < 0$ and $X_2^{(0)} < 0$. It is easy to that when the Gibbs sampler is started in C_1 it will stay there and never reach

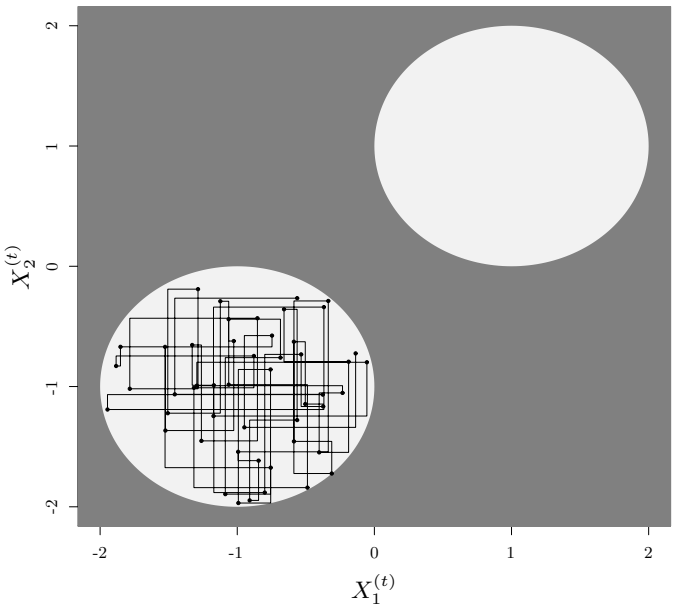


Figure 4.2. Illustration of a Gibbs sampler failing to sample from a distribution with unconnected support (uniform distribution on $\{(x_1, x_2) : \|(x_1, x_2) - (1, 1)\| \leq 1|\} \cup \{(x_1, x_2) : \|(x_1, x_2) - (-1, -1)\| \leq 1|\}$)

C_2. The reason for this is that the conditional distribution $X_2|X_1$ ($X_1|X_2$) is for $X_1 < 0$ ($X_2 < 0$) entirely concentrated on C_1. ◁

The following proposition gives a sufficient condition for irreducibility (and thus the recurrence) of the Markov chain generated by the Gibbs sampler. There are less strict conditions for the irreducibility and aperiodicity of the Markov chain generated by the Gibbs sampler (see e.g. Robert and Casella, 2004, Lemma 10.11).

Proposition 4.2. *If the joint distribution $f(x_1, \ldots, x_p)$ satisfies the positivity condition, the Gibbs sampler yields an irreducible, recurrent Markov chain.*

Proof. Let $\mathcal{X} \subset \text{supp}(f)$ be a set with $\int_{\mathcal{X}} f(x_1^{(t)}, \ldots, x_p^{(t)}) d(x_1^{(t)}, \ldots, x_p^{(t)}) > 0$.

$$\int_{\mathcal{X}} K(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}) d\mathbf{x}^{(t)} = \int_{\mathcal{X}} \underbrace{f_{X_1|X_{-1}}(x_1^{(t)}|x_2^{(t-1)}, \ldots, x_p^{(t-1)})}_{>0 \text{ (on a set of non-zero measure)}} \cdots \underbrace{f_{X_p|X_{-p}}(x_p^{(t)}|x_1^{(t)}, \ldots, x_{p-1}^{(t)})}_{>0 \text{ (on a set of non-zero measure)}} d\mathbf{x}^{(t)} > 0$$

Thus the Markov Chain $(\mathbf{X}^{(t)})_t$ is strongly f-irreducible. As f is the invariant distribution of the Markov chain, it is also recurrent (see the remark on page 39). □

If the transition kernel is absolutely continuous with respect to the dominating measure, then recurrence even implies Harris recurrence (see e.g. Robert and Casella, 2004, Lemma 10.9).

Now we have established all the necessary ingredients to state an ergodic theorem for the Gibbs sampler, which is a direct consequence of (Robert and Casella, 2004, Theorem 6.63).

Theorem 4.2. *If the Markov chain generated by the Gibbs sampler is irreducible and recurrent (which is e.g. the case when the positivity condition holds), then for any integrable function* $\varphi : E \to \mathbb{R}$

$$\lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \varphi(\mathbf{X}^{(t)}) \to \mathbb{E}_f\left(\varphi(\mathbf{X})\right)$$

for almost every starting value $\mathbf{X}^{(0)}$*. If the chain is Harris recurrent, then the above result holds for every starting value* $\mathbf{X}^{(0)}$*.*

Theorem 4.2 guarantees that we can approximate expectations $\mathbb{E}_f\left(\varphi(\mathbf{X})\right)$ by their empirical counterparts using *a single* Markov chain.

Example 4.4. Assume that we want to use a Gibbs sampler to estimate $\mathbb{P}(X_1 \geq 0, X_2 \geq 0)$ for a $\mathsf{N}_2\left(\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}, \begin{pmatrix}\sigma_1^2 & \sigma_{12}\\ \sigma_{12} & \sigma_2^2\end{pmatrix}\right)$ distribution. The marginal distributions are

$$X_1 \sim \mathsf{N}(\mu_1, \sigma_1^2) \qquad \text{and} \qquad X_2 \sim \mathsf{N}(\mu_2, \sigma_2^2)$$

In order to construct a Gibbs sampler, we need the conditional distributions $Y_1|Y_2 = y_2$ and $Y_2|Y_1 = y_1$. We have[1]

$$\begin{aligned} f(x_1, x_2) &\propto \exp\left(-\frac{1}{2}\left(\begin{pmatrix}x_1\\ x_2\end{pmatrix} - \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}\right)' \begin{pmatrix}\sigma_1^2 & \sigma_{12}\\ \sigma_{12} & \sigma_2^2\end{pmatrix}^{-1} \left(\begin{pmatrix}x_1\\ x_2\end{pmatrix} - \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}\right)\right) \\ &\propto \exp\left(-\frac{(x_1 - (\mu_1 + \sigma_{12}/\sigma_{22}^2(x_2 - \mu_2)))^2}{2(\sigma_1^2 - (\sigma_{12})^2/\sigma_2^2)}\right), \end{aligned}$$

[1] We make use of

$$\begin{aligned} &\left(\begin{pmatrix}x_1\\ x_2\end{pmatrix} - \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}\right)' \begin{pmatrix}\sigma_1^2 & \sigma_{12}\\ \sigma_{12} & \sigma_2^2\end{pmatrix}^{-1} \left(\begin{pmatrix}x_1\\ x_2\end{pmatrix} - \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}\right) \\ = \;& \frac{1}{\sigma_1^2\sigma_2^2 - (\sigma_{12})^2} \left(\begin{pmatrix}x_1\\ x_2\end{pmatrix} - \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}\right)' \begin{pmatrix}\sigma_2^2 & -\sigma_{12}\\ -\sigma_{12} & \sigma_1^2\end{pmatrix} \left(\begin{pmatrix}x_1\\ x_2\end{pmatrix} - \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}\right) \\ = \;& \frac{1}{\sigma_1^2\sigma_2^2 - (\sigma_{12})^2} \left(\sigma_2^2(x_1 - \mu_1)^2 - 2\sigma_{12}(x_1 - \mu_1)(x_2 - \mu_2)\right) + \text{ const} \\ = \;& \frac{1}{\sigma_1^2\sigma_2^2 - (\sigma_{12})^2} \left(\sigma_2^2 x_1^2 - 2\sigma_2^2 x_1\mu_1 - 2\sigma_{12}x_1(x_2 - \mu_2)\right) + \text{ const} \\ = \;& \frac{1}{\sigma_1^2 - (\sigma_{12})^2/\sigma_2^2} \left(x_1^2 - 2x_1(\mu_1 + \sigma_{12}/\sigma_2^2(x_2 - \mu_2))\right) + \text{ const} \\ = \;& \frac{1}{\sigma_1^2 - (\sigma_{12})^2/\sigma_2^2} \left(x_1 - (\mu_1 + \sigma_{12}/\sigma_2^2(x_2 - \mu_2))\right)^2 + \text{ const} \end{aligned}$$

i.e.

$$X_1|X_2 = x_2 \sim \mathsf{N}(\mu_1 + \sigma_{12}/\sigma_2^2(x_2 - \mu_2), \sigma_1^2 - (\sigma_{12})^2/\sigma_2^2)$$

Thus the Gibbs sampler for this problem consists of iterating for $t = 1, 2, \ldots$

1. Draw $X_1^{(t)} \sim \mathsf{N}(\mu_1 + \sigma_{12}/\sigma_2^2(X_2^{(t-1)} - \mu_2), \sigma_1^2 - (\sigma_{12})^2/\sigma_2^2)$
2. Draw $X_2^{(t)} \sim \mathsf{N}(\mu_2 + \sigma_{12}/\sigma_1^2(X_1^{(t)} - \mu_1), \sigma_2^2 - (\sigma_{12})^2/\sigma_1^2)$.

Now consider the special case $\mu_1 = \mu_2 = 0$, $\sigma_1^2 = \sigma_2^2 = 1$ and $\sigma_{12} = 0.3$. Figure 4.4 shows the sample paths of this Gibbs sampler.

Using Theorem 4.2 we can estimate $\mathbb{P}(X_1 \geq 0, X_2 \geq 0)$ by the proportion of samples $(X_1^{(t)}, X_2^{(t)})$ with $X_1^{(t)} \geq 0$ and $X_2^{(t)} \geq 0$. Figure 4.3 shows this estimate. ◁

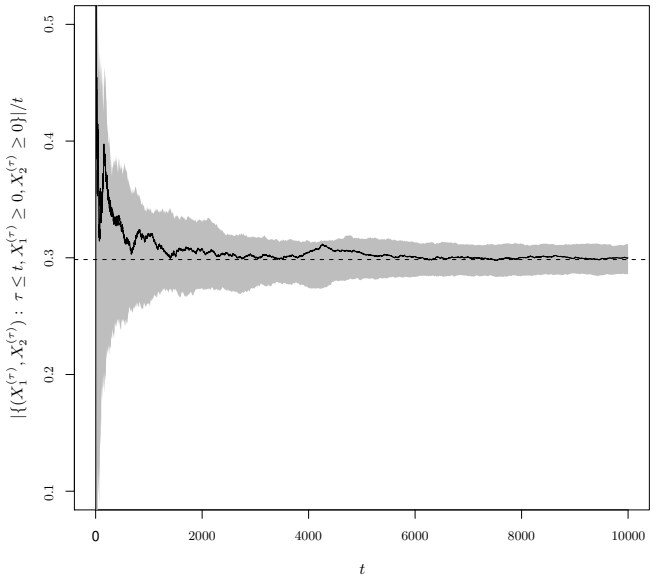
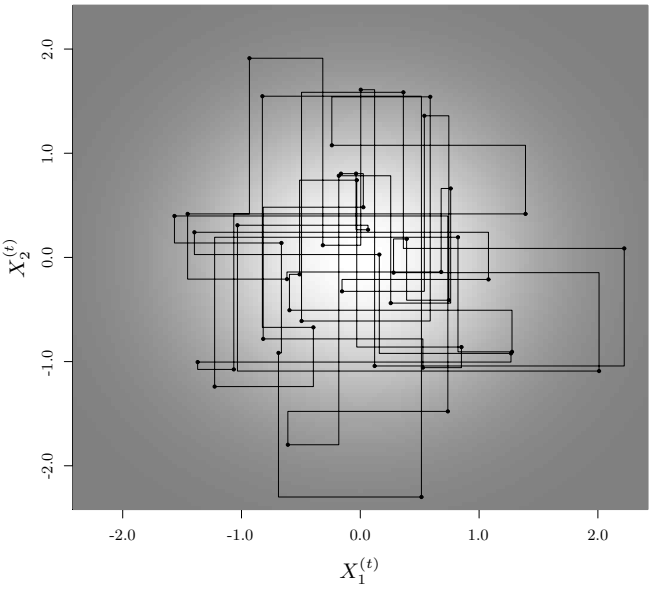


Figure 4.3. Estimate of the $\mathbb{P}(X_1 \geq 0, X_2 \geq 0)$ obtained using a Gibbs sampler. The area shaded in grey corresponds to the range of 100 replications.

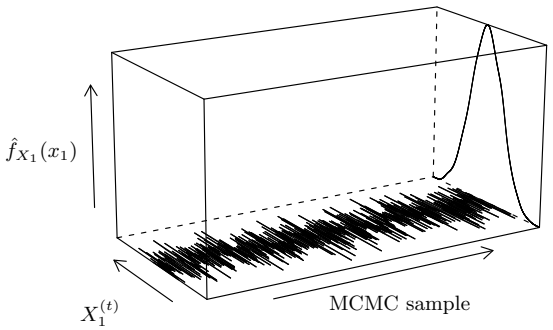
A Gibbs sampler is of course not the optimal way to sample from a $\mathsf{N}_\mathsf{p}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution. A more efficient way is: draw $Z_1, \ldots, Z_p \overset{\text{iid}}{\sim} \mathsf{N}(0, 1)$ and set $(X_1, \ldots, X_p)' = \boldsymbol{\Sigma}^{1/2}(Z_1, \ldots, Z_p)' + \boldsymbol{\mu}$ for some matrix square root, $\boldsymbol{\Sigma}^{1/2}$, such as that obtain via Cholesky decomposition, of the covariance matrix. As we shall see, in some instances the loss of efficiency arising from Gibbs sampling can be very severe.

Note that the realisations $(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots)$ form a Markov chain, and are thus *not* independent, but typically positively correlated. The correlation between the $\mathbf{X}^{(t)}$ is larger if the Markov chain moves only slowly (the chain is then said to be *slowly mixing*). For the Gibbs sampler this is typically the case if the variables X_j are strongly (positively or negatively) correlated, as the following example shows.

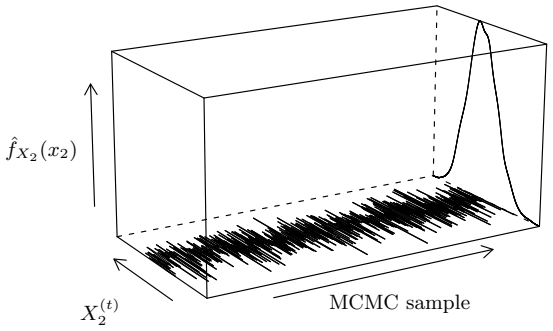
Example 4.5 (Sampling from a highly correlated bivariate Gaussian). Figure 4.5 shows the results obtained when sampling from a bivariate Normal distribution as in Example 4.4, however with $\sigma_{12} = 0.99$. This yields a correlation of $\rho(X_1, X_2) = 0.99$. This Gibbs sampler is a lot slower mixing than the one considered in Example 4.4 (and displayed in Figure 4.4): due to the strong correlation the Gibbs sampler can only perform very small movements. This makes subsequent samples $X_j^{(t-1)}$ and $X_j^{(t)}$ highly correlated and this leads to slower convergence, as the plot of the estimated densities show (panels (b) and (c) of Figures 4.4 and 4.5). ◁



(a) First 50 iterations of $(X_1^{(t)}, X_2^{(t)})$

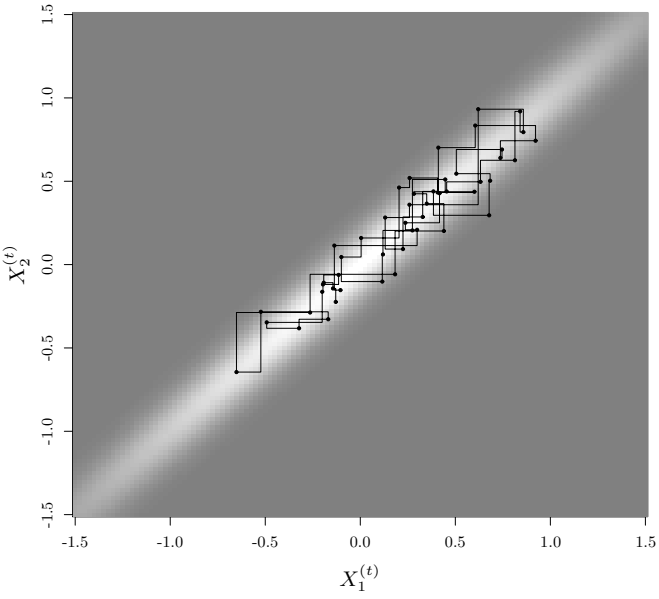


(b) Path of $X_1^{(t)}$ and estimated density of X after 1,000 iterations

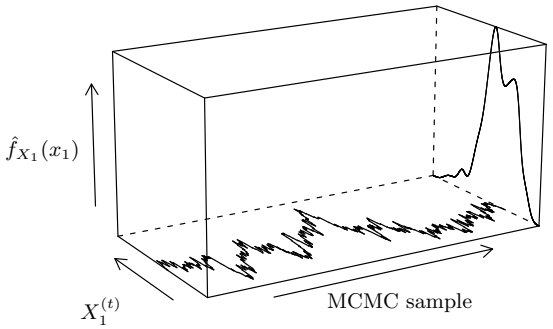


(c) Path of $X_2^{(t)}$ and estimated density of X_2 after 1,000 iterations

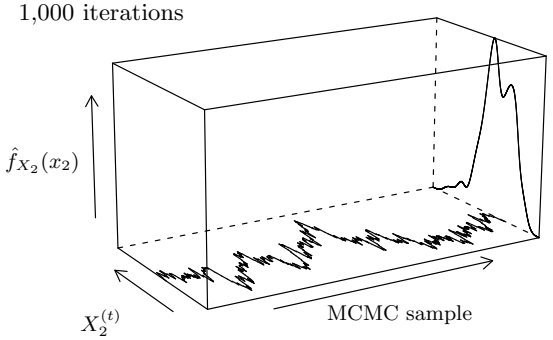
Figure 4.4. Gibbs sampler for a bivariate standard normal distribution (correlation $\rho(X_1, X_2) = 0.3$)



(a) First 50 iterations of $(X_1^{(t)}, X_2^{(t)})$



(b) Path of $X_1^{(t)}$ and estimated density of X_1 after 1,000 iterations



(c) Path of $X_2^{(t)}$ and estimated density of X_2 after 1,000 iterations

Figure 4.5. Gibbs sampler for a bivariate normal distribution with correlation $\rho(X_1, X_2) = 0.99$

5. The Metropolis-Hastings Algorithm

5.1 Algorithm

In the previous chapter we studied the Gibbs sampler, a special case of a Monte Carlo Markov Chain (MCMC) method: the target distribution is the invariant distribution of the Markov chain generated by the algorithm, to which it (hopefully) converges.

This chapter will introduce another MCMC method: the Metropolis-Hastings algorithm, which goes back to Metropolis et al. (1953) and Hastings (1970). Like rejection sampling (Algorithm 2.1), the Metropolis-Hastings algorithm is based on proposing values sampled from an instrumental distribution, which are then accepted with a certain probability that reflects how likely it is that they are from the target distribution f.

The main drawback of the rejection sampling algorithm is that it is often very difficult to come up with a suitable proposal distribution that leads to an efficient algorithm. One way around this problem is to allow for "local updates", i.e. let the proposed value depend on the last accepted value. This makes it easier to come up with a suitable (conditional) proposal, however at the price of yielding a Markov chain instead of a sequence of independent realisations.

Algorithm 5.1 (Metropolis-Hastings). Starting with $\mathbf{X}^{(0)} := (X_1^{(0)}, \ldots, X_p^{(0)})$ iterate for $t = 1, 2, \ldots$

1. Draw $\mathbf{X} \sim q(\cdot|\mathbf{X}^{(t-1)})$.
2. Compute
$$\alpha(\mathbf{X}|\mathbf{X}^{(t-1)}) = \min\left\{1, \frac{f(\mathbf{X}) \cdot q(\mathbf{X}^{(t-1)}|\mathbf{X})}{f(\mathbf{X}^{(t-1)}) \cdot q(\mathbf{X}|\mathbf{X}^{(t-1)})}\right\}. \tag{5.1}$$
3. With probability $\alpha(\mathbf{X}|\mathbf{X}^{(t-1)})$ set $\mathbf{X}^{(t)} = \mathbf{X}$, otherwise set $\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)}$.

Figure 5.1 illustrates the Metropolis-Hasting algorithm. Note that if the algorithm rejects the newly proposed value (unfilled circles joined by dotted lines in figure 5.1) it stays at its current value $\mathbf{X}^{(t-1)}$. The probability that the Metropolis-Hastings algorithm accepts the newly proposed state $\mathbf{X}$ given that it currently is in state $\mathbf{X}^{(t-1)}$ is

$$a(\mathbf{x}^{(t-1)}) = \int \alpha(\mathbf{x}|\mathbf{x}^{(t-1)}) q(\mathbf{x}|\mathbf{x}^{(t-1)}) \, d\mathbf{x}. \tag{5.2}$$

Just like the Gibbs sampler, the Metropolis-Hastings algorithm generates a Markov chain, whose properties will be discussed in the next section.

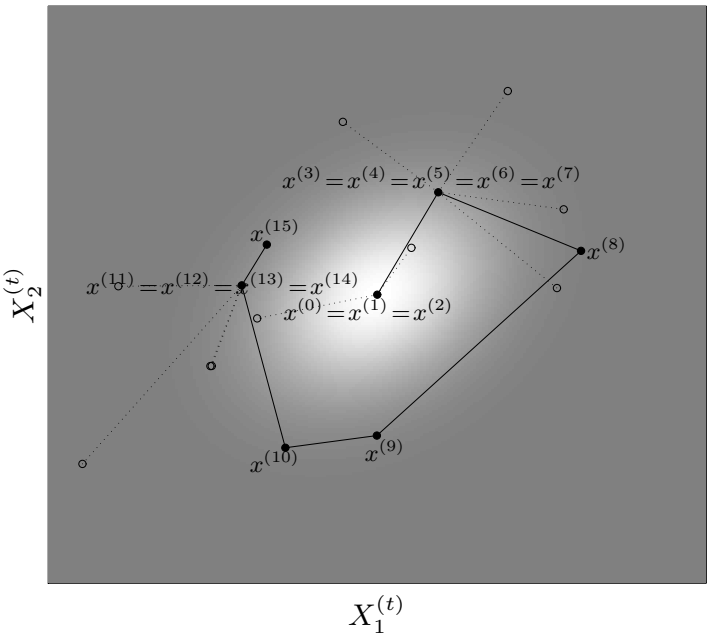


Figure 5.1. Illustration of the Metropolis-Hastings algorithm. Filled circles denote accepted states, unfilled circles rejected values.

Remark 5.1. The probability of acceptance (5.1) does not depend on the normalisation constant, i.e. if $f(\mathbf{x}) = C \cdot \pi(\mathbf{x})$, then

$$\frac{f(\mathbf{x}) \cdot q(\mathbf{x}^{(t-1)}|\mathbf{x})}{f(\mathbf{x}^{(t-1)}) \cdot q(\mathbf{x}|\mathbf{x}^{(t-1)})} = \frac{C\pi(\mathbf{x}) \cdot q(\mathbf{x}^{(t-1)}|\mathbf{x})}{C\pi(\mathbf{x}^{(t-1)}) \cdot q(\mathbf{x}|\mathbf{x}^{(t-1)})} = \frac{\pi(\mathbf{x}) \cdot q(\mathbf{x}^{(t-1)}|\mathbf{x})}{\pi(\mathbf{x}^{(t-1)}) \cdot q(\mathbf{x}|\mathbf{x}^{(t-1)})}$$

Thus f only needs to be known up to normalisation constant. Similarly, it is enough to know $q(\mathbf{x}^{(t-1)}|\mathbf{x})$ up to a multiplicative constant independent of $\mathbf{x}^{(t-1)}$ and $\mathbf{x}$.

5.2 Convergence results

Lemma 5.1. *The transition kernel of the Metropolis-Hastings algorithm is*

$$K(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}) = \alpha(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) + (1 - a(\mathbf{x}^{(t-1)}))\delta_{\mathbf{x}^{(t-1)}}(\mathbf{x}^{(t)}), \tag{5.3}$$

where $\delta_{\mathbf{x}^{(t-1)}(\cdot)}$ denotes Dirac-mass on $\{\mathbf{x}^{(t-1)}\}$.

Note that the transition kernel (5.3) is *not* absolutely continuous with respect to the Lebesgue measure (i.e. it doesn't have a simple density).

Proof. We have

$$\begin{aligned}
\mathbb{P}(\mathbf{X}^{(t)} \in \mathcal{X} | \mathbf{X}^{(t-1)} = \mathbf{x}^{(t-1)}) &= \mathbb{P}(\mathbf{X}^{(t)} \in \mathcal{X}, \text{new value accepted} | \mathbf{X}^{(t-1)} = \mathbf{x}^{(t-1)}) \\
&\quad + \mathbb{P}(\mathbf{X}^{(t)} \in \mathcal{X}, \text{new value rejected} | \mathbf{X}^{(t-1)} = \mathbf{x}^{(t-1)}) \\
&= \int_{\mathcal{X}} \alpha(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) \, d\mathbf{x}^{(t)} \\
&\quad + \underbrace{\underbrace{\mathbb{I}_{\mathcal{X}}(\mathbf{x}^{(t-1)})}_{=\int_{\mathcal{X}} \delta_{\mathbf{x}^{(t-1)}}(d\mathbf{x}^{(t)})} \underbrace{\mathbb{P}(\text{new value rejected} | \mathbf{X}^{(t-1)} = \mathbf{x}^{(t-1)})}_{=1-a(\mathbf{x}^{(t-1)})}}_{=\int_{\mathcal{X}} (1-a(\mathbf{x}^{(t-1)})) \delta_{\mathbf{x}^{(t-1)}}(d\mathbf{x}^{(t)})} \\
&= \int_{\mathcal{X}} \alpha(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) \, d\mathbf{x}^{(t)} + \int_{\mathcal{X}} (1 - a(\mathbf{x}^{(t-1)})) \delta_{\mathbf{x}^{(t-1)}}(d\mathbf{x}^{(t)})
\end{aligned}$$

□

Proposition 5.1. *The Metropolis-Hastings kernel (5.3) satisfies the detailed balance condition*

$$K(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}) f(\mathbf{x}^{(t-1)}) = K(\mathbf{x}^{(t)}, \mathbf{x}^{(t-1)}) f(\mathbf{x}^{(t)})$$

and thus $f(\mathbf{x})$ is the invariant distribution of the Markov chain $(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots)$ generated by the Metropolis-Hastings sampler. Furthermore the Markov chain is reversible.

Proof. We have that

$$\begin{aligned}
&\alpha(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) f(\mathbf{x}^{(t-1)}) = \min \left\{ 1, \frac{f(\mathbf{x}^{(t)}) q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})}{f(\mathbf{x}^{(t-1)}) q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})} \right\} q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) f(\mathbf{x}^{(t-1)}) \\
= \; & \min \left\{ f(\mathbf{x}^{(t-1)}) q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}), f(\mathbf{x}^{(t)}) q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) \right\} = \min \left\{ \frac{f(\mathbf{x}^{(t-1)}) q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})}{f(\mathbf{x}^{(t)}) q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})}, 1 \right\} q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) f(\mathbf{x}^{(t)}) \\
= \; & \alpha(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) f(\mathbf{x}^{(t)})
\end{aligned}$$

and thus

$$\begin{aligned}
K(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}) f(\mathbf{x}^{(t-1)}) &= \underbrace{\alpha(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) f(\mathbf{x}^{(t-1)})}_{=\alpha(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) f(\mathbf{x}^{(t)})} + \underbrace{(1 - a(\mathbf{x}^{(t-1)})) \underbrace{\delta_{\mathbf{x}^{(t-1)}}(\mathbf{x}^{(t)})}_{=0 \text{ if } \mathbf{x}^{(t)} \neq \mathbf{x}^{(t-1)}} f(\mathbf{x}^{(t-1)})}_{(1-a(\mathbf{x}^{(t)})) \delta_{\mathbf{x}^{(t)}}(\mathbf{x}^{(t-1)}) f(\mathbf{x}^{(t)})} \\
&= K(\mathbf{x}^{(t)}, \mathbf{x}^{(t-1)}) f(\mathbf{x}^{(t)})
\end{aligned}$$

The other conclusions follow by proposition 3.2, suitably adapted to the continuous case (i.e. replacing the sums by integrals). □

Next we need to examine whether the Metropolis-Hastings algorithm yields an irreducible chain. As with the Gibbs sampler, this is not necessarily the case, as the following example shows.

Example 5.1 (Reducible Metropolis-Hastings). Consider using a Metropolis-Hastings algorithm for sampling from a uniform distribution on $[0, 1] \cup [2, 3]$ and a $\mathsf{U}(x^{(t-1)} - \delta, x^{(t-1)} + \delta)$ distribution as proposal distribution $q(\cdot | x^{(t-1)})$. Figure 5.2 illustrates this example. It is easy to see that the resulting Markov chain is *not* irreducible if $\delta \leq 1$: in this case the chain either stays in $[0, 1]$ or $[2, 3]$. ◁

Under mild assumptions on the proposal $q(\cdot | \mathbf{x}^{(t-1)})$ one can however establish the irreducibility of the resulting Markov chain:

– If $q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})$ is positive for all $\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)} \in \text{supp}(f)$, then it is easy to see that we can reach any set of non-zero probability under f within a single step. The resulting Markov chain is thus strongly irreducible. Even though this condition seems rather restrictive, many popular choices of $q(\cdot | \mathbf{x}^{(t-1)})$ like multivariate Gaussians or t-distributions fulfil this condition.

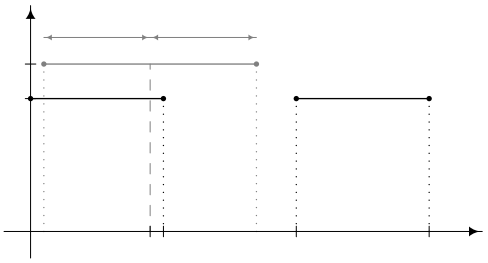


Figure 5.2. Illustration of example 5.1

– Roberts and Tweedie (1996) give a more general condition for the irreducibility of the resulting Markov chain: they only require that

$$\text{for some } \epsilon > 0 \; \exists \; \delta : \; q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) > \epsilon \text{ if } \|\mathbf{x}^{(t-1)} - \mathbf{x}^{(t)}\| < \delta$$

together with the boundedness of f on any compact subset of its support.

The Markov chain $(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \ldots)$ is further aperiodic, if there is positive probability that the chain remains in the current state, i.e. $\mathbb{P}(\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)}) > 0$, which is the case if

$$\mathbb{P}\left(f(\mathbf{X}^{(t-1)})q(\mathbf{X}|\mathbf{X}^{(t-1)}) > f(\mathbf{X})q(\mathbf{X}^{(t-1)}|\mathbf{X})\right) > 0.$$

Note that this condition is *not* met if we use a "perfect" proposal which employs the invariant distribution, f, as proposal distribution: in this case we accept every proposed value with probability 1 (see e.g. Remark 5.2 for another such case). But at least in the case of sampling from the invariant distribution we obtain iid samples from the target and need not worry about such details!

Proposition 5.2. *The Markov chain generated by the Metropolis-Hastings algorithm is Harris-recurrent if it is irreducible.*

Proof. Recurrence follows (using the result stated on page 39) from the irreducibility and the fact that f is the invariant distribution. For a proof of Harris recurrence see (Tierney, 1994). □

As we have now established (Harris-)recurrence, we are now ready to state an ergodic theorem (using theorems 3.1 and 3.2).

Theorem 5.1. *If the Markov chain generated by the Metropolis-Hastings algorithm is irreducible, then for any integrable function* $h : E \to \mathbb{R}$

$$\lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} h(\mathbf{X}^{(t)}) \to \mathbb{E}_f\left(h(\mathbf{X})\right)$$

for every starting value $\mathbf{X}^{(0)}$.

As with the Gibbs sampler the above ergodic theorem allows for inference using a single Markov chain.

5.3 The random walk Metropolis algorithm

In this section we will focus on an important special case of the Metropolis-Hastings algorithm: the random walk Metropolis-Hastings algorithm. Assume that we generate the newly proposed state $\mathbf{X}$ not using the fairly general

$$\mathbf{X} \sim q(\cdot|\mathbf{X}^{(t-1)}), \tag{5.4}$$

from algorithm 5.1, but rather

$$\mathbf{X} = \mathbf{X}^{(t-1)} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim g, \tag{5.5}$$

with g being a *symmetric* distribution. It is easy to see that (5.5) is a special case of (5.4) using $q(\mathbf{x}|\mathbf{x}^{(t-1)}) = g(\mathbf{x} - \mathbf{x}^{(t-1)})$. When using (5.5) the probability of acceptance simplifies to

$$\min\left\{1, \frac{f(\mathbf{X}) \cdot q(\mathbf{X}^{(t-1)}|\mathbf{X})}{f(\mathbf{X}^{(t-1)}) \cdot q(\mathbf{X}|\mathbf{X}^{(t-1)})}\right\} = \min\left\{1, \frac{f(\mathbf{X})}{f(\mathbf{X}^{(t-1)})}\right\},$$

as $q(\mathbf{X}|\mathbf{X}^{(t-1)}) = g(\mathbf{X} - \mathbf{X}^{(t-1)}) = g(\mathbf{X}^{(t-1)} - \mathbf{X}) = q(\mathbf{X}^{(t-1)}|\mathbf{X})$ using the symmetry of g. This yields the following algorithm which is a special case of Algorithm 5.1, which is actually the original algorithm proposed by Metropolis et al. (1953).

Algorithm 5.2 (Random walk Metropolis). Starting with $\mathbf{X}^{(0)} := (X_1^{(0)}, \ldots, X_p^{(0)})$ and using a symmetric distributon g, iterate for $t = 1, 2, \ldots$

1. Draw $\boldsymbol{\varepsilon} \sim g$ and set $\mathbf{X} = \mathbf{X}^{(t-1)} + \boldsymbol{\varepsilon}$.
2. Compute
$$\alpha(\mathbf{X}|\mathbf{X}^{(t-1)}) = \min\left\{1, \frac{f(\mathbf{X})}{f(\mathbf{X}^{(t-1)})}\right\}. \tag{5.6}$$
3. With probability $\alpha(\mathbf{X}|\mathbf{X}^{(t-1)})$ set $\mathbf{X}^{(t)} = \mathbf{X}$, otherwise set $\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)}$.

Example 5.2 (Bayesian probit model). In a medical study on infections resulting from birth by Cesarean section (taken from Fahrmeir and Tutz, 2001) three influence factors have been studied: an indicator whether the Cesarian was planned or not (z_{i1}), an indicator of whether additional risk factors were present at the time of birth (z_{i2}), and an indicator of whether antibiotics were given as a prophylaxis (z_{i3}). The response Y_i is the number of infections that were observed amongst n_i patients having the same influence factors (covariates). The data is given in table 5.1.

Number of births		planned	risk factors	antibiotics
with infection	total			
y_i	n_i	z_{i1}	z_{i2}	z_{i3}
11	98	1	1	1
1	18	0	1	1
0	2	0	0	1
23	26	1	1	0
28	58	0	1	0
0	9	1	0	0
8	40	0	0	0

Table 5.1. Data used in example 5.2

The data can be modelled by assuming that

$$Y_i \sim \mathsf{Bin}(n_i, \pi_i), \qquad \pi = \Phi(\mathbf{z}_i'\boldsymbol{\beta}),$$

where $\mathbf{z}_i = (1, z_{i1}, z_{i2}, z_{i3})$ and $\Phi(\cdot)$ being the CDF of the $\mathsf{N}(0,1)$ distribution. Note that $\Phi(t) \in [0,1]$ for all $t \in \mathbb{R}$.

A suitable prior distribution for the parameter of interest $\boldsymbol{\beta}$ is $\boldsymbol{\beta} \sim N(\mathbf{0}, \mathbb{I}/\lambda)$. The posterior density of $\boldsymbol{\beta}$ is

$$f(\boldsymbol{\beta}|y_1, \ldots, y_n) \propto \left(\prod_{i=1}^{n} \Phi(\mathbf{z}_i'\boldsymbol{\beta})^{y_i} \cdot (1 - \Phi(\mathbf{z}_i'\boldsymbol{\beta}))^{n_i - y_i}\right) \cdot \exp\left(-\frac{\lambda}{2}\sum_{j=0}^{3}\beta_j^2\right)$$

We can sample from the above posterior distribution using the following random walk Metropolis algorithm. Starting with any $\boldsymbol{\beta}^{(0)}$ iterate for $t = 1, 2, \ldots$:

1. Draw $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ and set $\boldsymbol{\beta} = \boldsymbol{\beta}^{(t-1)} + \boldsymbol{\varepsilon}$.
2. Compute
$$\alpha(\boldsymbol{\beta}|\boldsymbol{\beta}^{(t-1)}) = \min\left\{1, \frac{f(\boldsymbol{\beta}|Y_1, \ldots, Y_n)}{f(\boldsymbol{\beta}^{(t-1)}|Y_1, \ldots, Y_n)}\right\}.$$
3. With probability $\alpha(\boldsymbol{\beta}|\boldsymbol{\beta}^{(t-1)})$ set $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}$, otherwise set $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^{(t-1)}$.

To keep things simple, we choose the covariance $\boldsymbol{\Sigma}$ of the proposal to be $0.08 \cdot \mathbb{I}$.

		Posterior mean	95% credible interval	
intercept	β_0	-1.0952	-1.4646	-0.7333
planned	β_1	0.6201	0.2029	1.0413
risk factors	β_2	1.2000	0.7783	1.6296
antibiotics	β_3	-1.8993	-2.3636	-1.471

Table 5.2. Parameter estimates obtained for the Bayesian probit model from example 5.2

Figure 5.3 and table 5.2 show the results obtained using 50,000 samples (you might want to consider a longer chain in practice). Note that the convergence of the $\beta_j^{(t)}$ is to a distribution, whereas the cumulative averages $\sum_{\tau=1}^{t} \beta_j^{(\tau)}/t$ converge, as the ergodic theorem implies, to a value. For figure 5.3 and table 5.2 the first 10,000 samples have been discarded ("burn-in"). ◁

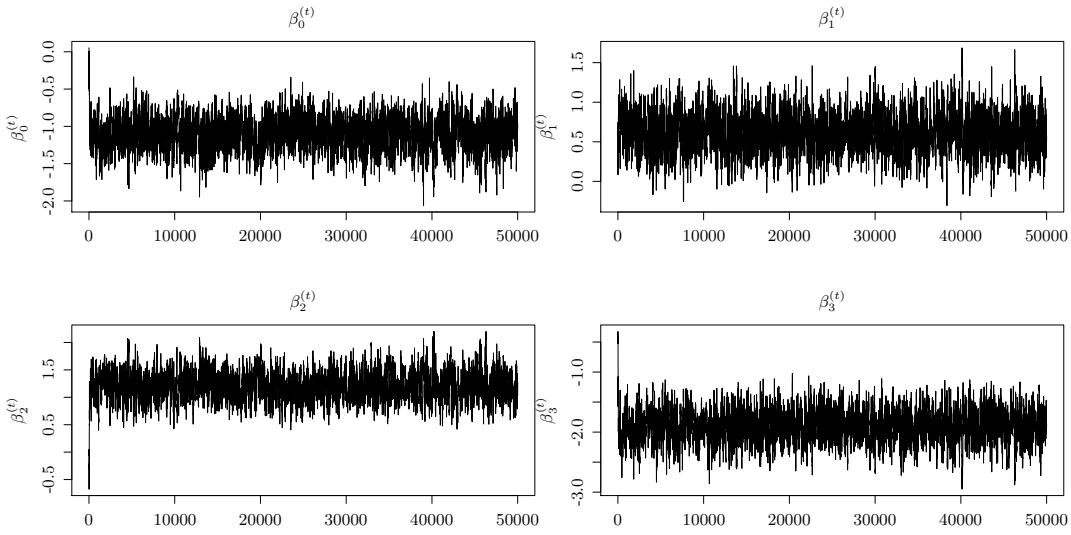
5.3.1 Choosing the proposal distribution

The efficiency of a Metropolis-Hastings sampler depends on the choice of the proposal distribution $q(\cdot|\mathbf{x}^{(t-1)})$. An ideal choice of proposal would lead to a small correlation of subsequent realisations $\mathbf{X}^{(t-1)}$ and $\mathbf{X}^{(t)}$. This correlation has two sources:

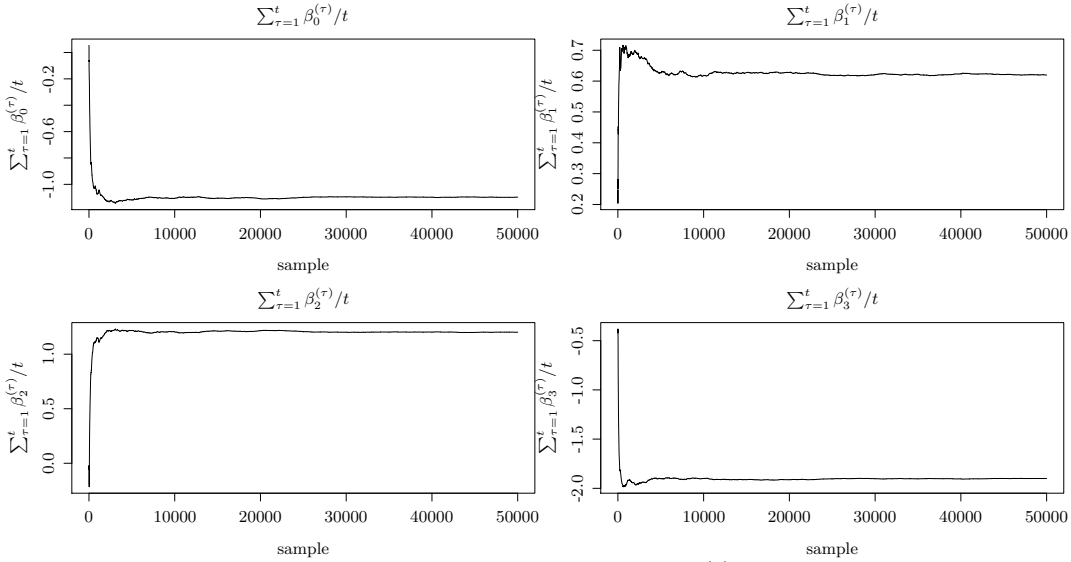
– the correlation between the current state $\mathbf{X}^{(t-1)}$ and the newly proposed value $\mathbf{X} \sim q(\cdot|\mathbf{X}^{(t-1)})$, and
– the correlation introduced by retaining a value $\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)}$ because the newly generated value $\mathbf{X}$ has been rejected.

Thus we would ideally want a proposal distribution that both allows for fast changes in the $\mathbf{X}^{(t)}$ and yields a high probability of acceptance. Unfortunately these are two competing goals. If we choose a proposal distribution with a small variance, the probability of acceptance will be high, however the resulting Markov chain will be highly correlated, as the $X^{(t)}$ change only very slowly. If, on the other hand, we choose a proposal distribution with a large variance, the $X^{(t)}$ can potentially move very fast, however the probability of acceptance will be rather low.

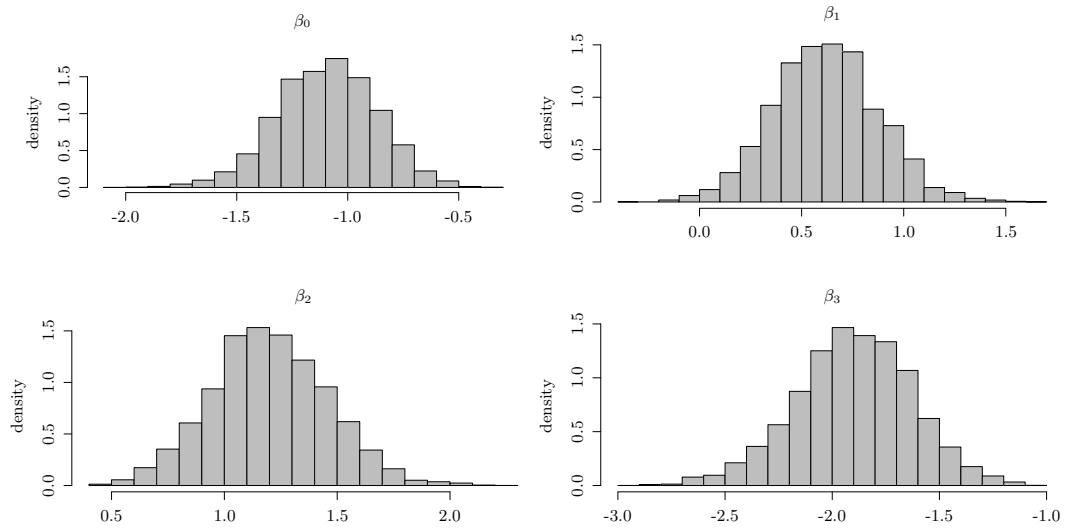
Example 5.3. Assume we want to sample from a $N(0, 1)$ distribution using a random walk Metropolis algorithm with $\varepsilon \sim N(0, \sigma^2)$. At first sight, we might think that setting $\sigma^2 = 1$ is the optimal choice, this is, however, not the case. In this example we examine the choices: $\sigma^2 = 0.1$, $\sigma^2 = 1$, $\sigma^2 = 2.38^2$, and $\sigma^2 = 10^2$. Figure 5.4 shows the sample paths of a single run of the corresponding random walk Metropolis algorithm. Rejected values are drawn as grey open circles. Table 5.3 shows the average correlation $\rho(X^{(t-1)}, X^{(t)})$ as well as the average probability of acceptance $\alpha(X|X^{(t-1)})$ averaged over 100 runs of the algorithm. Choosing σ^2



(a) Sample paths of the $\beta_j^{(t)}$



(b) Cumulative averages $\sum_{\tau=1}^{t} \beta_j^{(\tau)}/t$



(c) Posterior distributions of the β_j

Figure 5.3. Results obtained for the Bayesian probit model from example 5.2

too small yields a very high probability of acceptance, however at the price of a chain that is hardly moving. Choosing σ^2 too large allows the chain to make large jumps, however most of the proposed values are rejected, so the chain remains for a long time at each accepted value. The results suggest that $\sigma^2 = 2.38^2$ is the optimal choice. This corresponds to the theoretical results of Gelman et al. (1995). ◁

	Autocorrelation $\rho(X^{(t-1)}, X^{(t)})$		Probability of acceptance $\alpha(X, X^{(t-1)})$	
	Mean	95% CI	Mean	95% CI
$\sigma^2 = 0.1^2$	0.9901	(0.9891,0.9910)	0.9694	(0.9677,0.9710)
$\sigma^2 = 1$	0.7733	(0.7676,0.7791)	0.7038	(0.7014,0.7061)
$\sigma^2 = 2.38^2$	0.6225	(0.6162,0.6289)	0.4426	(0.4401,0.4452)
$\sigma^2 = 10^2$	0.8360	(0.8303,0.8418)	0.1255	(0.1237,0.1274)

Table 5.3. Average correlation $\rho(X^{(t-1)}, X^{(t)})$ and average probability of acceptance $\alpha(X|X^{(t-1)})$ found in example 5.3 for different choices of the proposal variance σ^2.

Finding the ideal proposal distribution $q(\cdot|\mathbf{x}^{(t-1)})$ is an art. The optimal proposal would be sampling directly from the target distribution. The very reason for using a Metropolis-Hastings algorithm is, however, that we cannot sample directly from the target! This difficulty is the price we have to pay for the generality of the Metropolis-Hastings algorithm. Popular choices for random walk proposals are multivariate Gaussians or t-distributions. The latter have heavier tails. The covariance structure of the proposal distribution should ideally reflect the covariance of the target distribution. Gelman et al. (1997) propose to adjust the proposal such that the acceptance rate is around 1/2 for one- or two dimensional target distributions, and around 1/4 for proposals in higher-dimensional spaces. This recommendation is in line with the results we obtained in the above simple example and the guidelines which motivate them. Remember, however, that these are just rough guidelines and there is little to be gained from fine-tuning acceptance rates to several decimal places.

Example 5.4 (Bayesian probit model (continued)). In the Bayesian probit model we studied in example 5.2 we drew

$$\boldsymbol{\varepsilon} \sim \mathsf{N}(\mathbf{0}, \boldsymbol{\Sigma})$$

with $\boldsymbol{\Sigma} = 0.08 \cdot \mathbf{I}$, i.e. we modelled the components of $\boldsymbol{\varepsilon}$ to be independent. The proportion of accepted values we obtained in example 5.2 was 13.9%. Table 5.4 (a) shows the corresponding autocorrelation. The resulting Markov chain can be made faster mixing by using a proposal distribution that represents the covariance structure of the posterior distribution of $\boldsymbol{\beta}$.

This can be done by resorting to the frequentist theory of generalised linear models (GLM): it suggests that the asymptotic covariance of the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$ is $(\mathbf{Z}'\mathbf{D}\mathbf{Z})^{-1}$, where $\mathbf{Z}$ is the matrix of the covariates, and $\mathbf{D}$ is a suitable diagonal matrix. When using $\boldsymbol{\Sigma} = 2 \cdot (\mathbf{Z}'\mathbf{D}\mathbf{Z})^{-1}$ in the algorithm presented in Section 5.2 we can obtain better mixing performance: the autocorrelation is reduced (see table 5.4 (b)), and the proportion of accepted values obtained increases to 20.0%. Note that the determinant of both choices of $\boldsymbol{\Sigma}$ was chosen to be the same, so the improvement of the mixing behaviour is entirely due to a difference in the structure of the the covariance. ◁

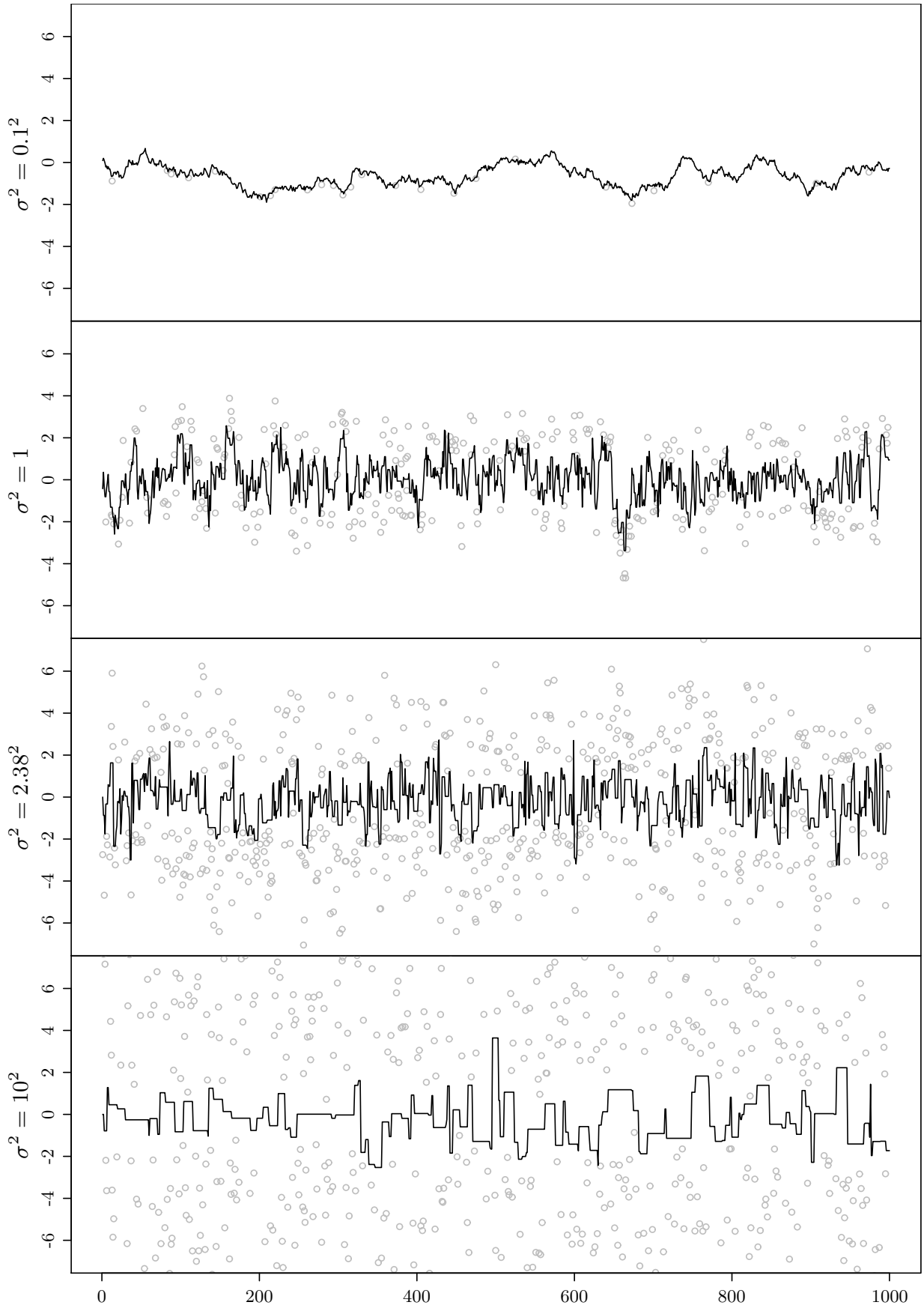


Figure 5.4. Sample paths for example 5.3 for different choices of the proposal variance σ^2. Open grey discs represent rejected values.

(a) $\boldsymbol{\Sigma} = 0.08 \cdot \mathbf{I}$

	β_0	β_1	β_2	β_3
Autocorrelation $\rho(\beta_j^{(t-1)}, \beta_j^{(t)})$	0.9496	0.9503	0.9562	0.9532

(b) $\boldsymbol{\Sigma} = 2 \cdot (\mathbf{Z}'\mathbf{D}\mathbf{Z})^{-1}$

	β_0	β_1	β_2	β_3
Autocorrelation $\rho(\beta_j^{(t-1)}, \beta_j^{(t)})$	0.8726	0.8765	0.8741	0.8792

Table 5.4. Autocorrelation $\rho(\beta_j^{(t-1)}, \beta_j^{(t)})$ between subsequent samples for the two choices of the covariance $\boldsymbol{\Sigma}$.

5.4 The (Metropolised) Indepence Sampler

Although the random walk proposals considered thus far are appealing because we can in principle employ them without detailed knowledge of the structure of the target distibution and without dedicating considerable effort to their design, if we do have information about the target distribution we may wish to use it to design *global* rather than local proposals and hence, we might hope, to reduce the autocorrelation of the chain.

This is, indeed, possible and if we can construct proposal distributions which have a similar form to the target distribution we can obtain good performance within an MCMC algorithm.

Choosing proposals of the form $q(x^{(t)}|x^{(t-1)}) = q(^{(x_t)}$ (i.e. which are independent of the current state) leads to what is known as the *Metropolised Independence Sampler* or, sometimes, just the *Independence Sampler*. This name is potentially a little misleading, as this algorithm does not yield independent samples, it simply employs proposals which are themselves independent of the current state of the chain.

Algorithm 5.3 (Metropolised Independence Sampler). Starting with $\mathbf{X}^{(0)} := (X_1^{(0)}, \ldots, X_p^{(0)})$ iterate for $t = 1, 2, \ldots$

1. Draw $\mathbf{X} \sim q(\cdot)$.
2. Compute

$$\alpha(\mathbf{X}|\mathbf{X}^{(t-1)}) = \min\left\{1, \frac{f(\mathbf{X}) \cdot q(\mathbf{X}^{(t-1)})}{f(\mathbf{X}^{(t-1)}) \cdot q(\mathbf{X})}\right\} = \min\left\{1, \frac{f(\mathbf{X})/q(\mathbf{X})}{f(\mathbf{X}^{(t-1)})/q(\mathbf{X}^{(t-1)})}\right\}. \tag{5.7}$$

3. With probability $\alpha(\mathbf{X}|\mathbf{X}^{(t-1)})$ set $\mathbf{X}^{(t)} = \mathbf{X}$, otherwise set $\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)}$.

The form of the acceptance probability given in Equation 5.7 is highly suggestive: the ratio within the minimum is exactly a ratio of importance weights. If we sampled independently from q and used those samples to approximate expectations with respect to f by importance sampling, we'd be using exactly the numerator of this ratio as the importance weight. If we used the same strategy within a rejection sampling setting, assuming this ratio to be bounded, then we'd need an acceptance probability proportional to this ratio.

In Section 5.4.1 we will see that under those conditions in which the independence sampler proposal would be a good rejection sampling proposal it will also be a good proposal within a MCMC setting. First, however, it's interesting to consider the relationship between the independence sampler and its rejection-sampling counterpart.

Proposition 5.3. *Acceptance Rates If $f(x)/q(x) \leq M < \infty$ the acceptance rate of the independence sampler is at least as high as that of the corresponding rejection sampler.*

Proof. Simply expanding the acceptance probability at any point x we establish that:

$$\begin{aligned} a(x) &= \int q(y)\alpha(x,y)dy \\ &= \int q(y)\min\left(1, \frac{f(y)/q(y)}{f(x)/q(x)}\right)dy \\ &= \int \min\left(q(y), \frac{f(y)}{f(x)/q(x)}\right)dy \\ &\geq \int \min\left(f(y)/M, f(y)/M\right)dy = 1/M. \end{aligned}$$

and as this holds for any M which bounds f/q the acceptance rate of the independence sampler is lower bounded by the best possible acceptance rate for any rejection sampler. □

However, this comes at a cost: with rejection sampling one obtains independent samples from the target, with the independence sampler this is not the case.

5.4.1 Ergodicity and the Independence Sampler

One method of assessing the convergence of Markov chains is to look at how far away from the invariant distribution it's possible for the marginal distribution of the chain to remain after a certain number of iterations. In order to make such an assessment it is necessary to define a distance on the space of probability measures — i.e. a function which quantifies *how different* to probability distributions over a common state space are. One common distance in this setting attempts to quantify this by considering how different the probability of any given subset of the state space can be under the two distributions.

Definition 5.1. *Total Variation The* total variation *distance between two probability distributions, f and g defined on a common space, E, may be defined as:*

$$||f-g||_{TV} := 2\sup_A \left|\int_A f(x) - g(x)dx\right|.$$

where the supremum is taken over all (measurable) $A \subset E$.

Actually, in the case of probability densities, this is exactly the L_1 distance between those densities and you may find this formulation easier to interpret. The following proposition formalises this relationship.

Proposition 5.4. *For any pair of probability densities defined on a common space E:*

$$||f-g||_{TV} = \int |f(x) - g(x)|dx.$$

Proof. Let $A^\star = \{x : f(x) > g(x)\}$. It's clear that for all A:

$$\left|\int_A (f(x) - g(x))dx\right| \leq \left|\int_{A^\star} f(x) - g(x)dx\right|.$$

Noting further that $\int_{A\cup A^c}(f(x) - g(x))dx = 0$ we can establish that for any (measurable) A:

$$\int_A (f(x) - g(x))dx = -\int_{A^c} (f(x) - g(x))dx$$

and so

$$2\left|\int_A f(x) - g(x)dx\right| = \left|\int_A f(x) - g(x)dx\right| + \left|\int_{A^c} f(x) - g(x)dx\right|.$$

On $A^\star$, $f(x) > g(x)$ while on $(A^\star)^c$ the reverse is true, so:

$$\left|\int_{A^\star} f(x) - g(x)dx\right| = \int_{A^\star} f(x) - g(x)dx = \int_{A^\star} |f(x) - g(x)|dx$$

and

$$\left|\int_{(A^\star)^c} f(x) - g(x)dx\right| = -\int_{(A^\star)^c} f(x) - g(x)dx = \int_{(A^\star)^c} |f(x) - g(x)|dx.$$

Combining everything, we establish that:

$$\begin{aligned}
||f - g||_{\mathrm{TV}} :=& 2\sup_A \left|\int_A f(x) - g(x)dx\right| \\
=& 2\left|\int_{A^\star} f(x) - g(x)dx\right| \\
=& \left|\int_{A^\star} f(x) - g(x)dx\right| + \left|\int_{(A^\star)^c} f(x) - g(x)dx\right| \\
=& \int_{A^\star} |f(x) - g(x)|dx + \int_{(A^\star)^c} |f(x) - g(x)|dx \\
=& \int |f(x) - g(x)|dx.
\end{aligned}$$

□

One reason that we might be interested in this type of distance in a Monte Carlo setting is that if we can bound the total variation distance between two probability distributions then we can bound the difference in the expectation of measurable functions as made precise in proposition 5.5. Therefore, if we can bound the total variation difference between the marginal distribution of a Markov chain after finitely-many iterations and its invariant distribution then we can have confidence that the bias introduced by initialising the chain out of equilibrium (i.e. not taking $X_1 \sim f$ and starting the chain from its stationary distribution as we would like to) leaving only the Monte Carlo error to worry about.

Proposition 5.5 (TV Error Bound). *If* $||f - g||_{TV} \leq \epsilon$*, then* $|\mathbb{E}_f[\varphi] - \mathbb{E}_g[\varphi]| \leq \epsilon \sup_x |\varphi(x)|$.

Proof. Let $A^\star = \{x : f(x) > g(x)\}$. Then:

$$\begin{aligned}
&\sup_{\{\psi:\sup_x(|\psi(x)|)\leq 1\}} \left|\int \psi(x)(f(x) - g(x))dx\right| \\
=& \int_{A^\star} f(x) - g(x)dx - \int_{(A^\star)^C} (f(x) - g(x))dx \\
=& \int |f(x) - g(x)| = ||f - g||_{\mathrm{TV}}.
\end{aligned}$$

Writing $\psi = \varphi/(\sup_x \varphi(x))$, the result is immediate. □

Having defined total variation, we can define three forms of ergodicity.

Definition 5.2. *Forms of Ergodicity An f-invariant Markov kernel, K, is said to be* ergodic *if*

$$\lim_{n\to\infty} ||K^n(x, \cdot) - f(\cdot)||_{TV} = 0$$

where $||K^n(x, \cdot) - f(\cdot)||_{TV} = \int |K^n(x, y) - f(y)|dy$.

If, this statement can be strengthened to:

$$||K^n(x, \cdot) - f(\cdot)||_{TV} \leq M(x)\rho^n$$

for some $M(x) < \infty$ and $\rho < 1$ then the kernel is said to be geometrically ergodic *and if it can be further strengthened to:*

$$||K^n(x,\cdot) - f(\cdot)||_{TV} \leq M\rho^n$$

for some $M < \infty$ which does not depend upon x then it is said to be uniformly ergodic *(or, more pedantically, uniformly geometrically ergodic: this condition tells us that the chain is geometrically ergodic and that the associated constants are uniform over the entire space).*

These are useful because they tell us something about the qualitative *rate of convergence* of the Markov chain to stationarity. However, we should bear in mind that if we don't know the constants M and ρ then even a uniformly ergodic chain can in practice converge rather slowly.

We're now in a position to state and prove one celebrated result about independence samplers.

Proposition 5.6. *If an independence sampler uses proposal q and target f and $f(y)/q(y) \leq M < \infty$ then the associated Markov kernel is* uniformly ergodic.

Proof. We follow the argument of (Robert and Casella, 2004, Exercise 7.11). First we show that $f(y)/q(y) \leq M \Rightarrow K(x,y) \geq f(y)/M$:

$$\begin{aligned} K(x,y) =& q(y)\alpha(x,y) + (1 - a(x))\delta_x(y) \geq q(y)\alpha(x,y) \\ \geq& q(y)\min\left(\frac{f(y)/q(y)}{f(x)/q(x)}, 1\right) \\ =& \min\left(\frac{f(y)}{f(x)/q(x)}, q(y)\right) \end{aligned}$$

Under the assumptions of the proposition we have that $f(x)/q(x) \leq M$ and $q(y) \geq f(y)/M$ and so:

$$K(x,y) \geq \min\left(\frac{f(y)}{M}, f(y)/M\right) = f(y)/M. \tag{5.8}$$

Now we establish a preliminary result, defining $A^\star(x) = \{y : f(y) > K(x,y)\}$:

$$\begin{aligned} \sup_A \left| \int_A K(x,y) - f(y)dy \right| &= \left| \int_{A^\star(x)} K(x,y) - f(y)dy \right| \\ &= \int_{A^\star(x)} f(y) - K(x,y)dy \\ &\leq \int_{A^\star(x)} f(y) - (1/M)f(y)dy \leq (1 - 1/M) \end{aligned}$$

using Equation 5.8 to bound the negative term from below. We use this as a base case for induction, we have (by substituting this bound into the definition of the total variation norm) for $n = 1$: $||K^n(x,\cdot) - f(\cdot)||_{\mathrm{TV}} \leq 2(1 - 1/M)^n$.

We now turn to the induction step, and assume that the hypothesis $||K^n(x,\cdot) - f(\cdot)||_{\mathrm{TV}} \leq 2(1-1/M)^n$ holds for some n and write, for any (measurable) A:

$$\int_A (K^{n+1}(x,y) - f(y))dy = \int \int_A (K^n(u,y) - f(y))dy(K(x,u) - f(u))du$$

(you can check this by remembering that K if f-invariant and expanding the right hand side explicitly).

The induction hypothesis tells us that the integral over y is bounded by $(1 - 1/M)^n$ (it's easy to establish that the integral over any set of the difference between any pair of probability densities is at most half of the total variation distance between those densities) and a similar argument to that used to prove the base case establishes that the integral over u can then be bounded by $(1 - 1/M)$:

$$\int_A (K^{n+1}(x,y) - f(y))dy = \int \int_A (K^n(u,y) - f(y))dy(K(x,u) - f(u))du$$
$$\leq \int (1 - 1/M)^n (K(x,u) - f(u))du \leq (1 - 1/M)^{n+1}$$

Writing the total variation as twice the supremum over A of quantities of this form completes the argument. □

5.5 Composing kernels: Mixtures and Cycles

It can be advantageous, especially in the case of more complex distributions, to combine different Metropolis-Hastings updates into a single algorithm. Each of the different Metropolis-Hastings updates corresponds to a transition kernel $K^{(j)}$. As with the substeps of Gibbs sampler there are two ways of combining the transition kernels $K^{(1)}, \dots, K^{(r)}$:

– As in the systematic scan Gibbs sampler, we can cycle through the kernels in a deterministic order, i.e. first carry out the Metropolis-Hastings update corresponding the the kernel $K^{(1)}$, then carry out the one corresponding to $K^{(2)}$, etc. until we start again with $K^{(1)}$. The transition kernel of this composite chain is

$$K^\circ(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}) = \int \cdots \int K^{(1)}(\mathbf{x}^{(t-1)}, \boldsymbol{\xi}^{(1)}) K^{(2)}(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}) \cdots K^{(r)}(\boldsymbol{\xi}^{(r-1)}, \mathbf{x}^{(t)})\, d\boldsymbol{\xi}^{(r-1)} \cdots d\boldsymbol{\xi}^{(1)}$$

If each of the transition kernels $K^{(j)}$ has the invariant distribution f (i.e. $\int f(\mathbf{x}^{(t-1)}) K(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)})\, d\mathbf{x}^{(t-1)} = f(\mathbf{x}^{(t)})$), then K° has f as invariant distribution, too, as

$$\begin{aligned}
&\int f(\mathbf{x}^{(t-1)}) K^\circ(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)})\, d\mathbf{x}^{(t-1)} \\
= &\underbrace{\underbrace{\int \cdots \int \underbrace{\int K^{(1)}(\mathbf{x}^{(t-1)}, \boldsymbol{\xi}^{(1)}) f(\mathbf{x}^{(t-1)}) d\mathbf{x}^{(t-1)}}_{=f(\boldsymbol{\xi}^{(1)})} K^{(2)}(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}) d\boldsymbol{\xi}^{(1)}}_{f(\boldsymbol{\xi}^{(2)})} \cdots d\boldsymbol{\xi}^{(r-2)}}_{=f(\boldsymbol{\xi}^{(r-1)})} K^{(r)}(\boldsymbol{\xi}^{(r-1)}, \mathbf{x}^{(t)}) d\boldsymbol{\xi}^{(r-1)} \\
= &f(\mathbf{x}^{(t)})
\end{aligned}$$

– Alternatively, we can, as in the random scan Gibbs sampler, choose each time at random which of the kernels should be used, i.e. use the kernel $K^{(j)}$ with probability $w_j > 0$ ($\sum_{\iota=1}^r w_\iota = 1$). The corresponding kernel of the composite chain is the mixture

$$K^+(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}) = \sum_{\iota=1}^r w_\iota K^{(\iota)}(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)})$$

Once again, if each of the transition kernels $K^{(j)}$ has the invariant distribution f, then K^+ has f as invariant distribution:

$$\int f(\mathbf{x}^{(t-1)}) K^+(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)})\, d\mathbf{x}^{(t-1)} = \sum_{\iota=1}^r w_\iota \underbrace{\int f(\mathbf{x}^{(t-1)}) K^{(\iota)}(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)})\, d\mathbf{x}^{(t-1)}}_{=f(\mathbf{x}^{(t)})} = f(\mathbf{x}^{(t)}).$$

Example 5.5 (One-at-a-time Metropolis-Hastings). One example of a method using composite kernels is the so-called *one-at-a-time* Metropolis-Hastings algorithm. Consider the case of a p-dimensional random variable $\mathbf{X} = (X_1, \dots, X_p)$. The Metropolis-Hastings algorithms 5.1 and 5.2 update all components at a time.

It can, however, be difficult to come up with a suitable proposal distribution $q(\cdot|\mathbf{x}^{(t-1)})$ (or g) for all variables. Alternatively, we could, as in the Gibbs sampler, update each component separately. For this we need p proposal distributions $q_1, \ldots, q_p$ for updating each of the X_j. The j-th proposal q_j (and thus the j-th kernel $K^{(j)}$) corresponds to updating the X_j.

As mentioned above we can cycle deterministically through the kernels (corresponding to the kernel K°), yielding the following algorithm. Starting with $\mathbf{X}^{(0)} = (X_1^{(0)}, \ldots, X_p^{(0)})$ iterate

1. i. Draw $X_1 \sim q_1(\cdot|X_2^{(t-1)}, \ldots, X_p^{(t-1)})$.
   ii. Compute $\alpha_1 = \min\left\{1, \frac{f(X_1, X_2^{(t-1)}, \ldots, X_p^{(t-1)}) \cdot q_1(X_1^{(t-1)}|X_1, X_2^{(t-1)}, \ldots, X_p^{(t-1)})}{f(X_1^{(t-1)}, X_2^{(t-1)}, \ldots, X_p^{(t-1)}) \cdot q_1(X_1|X_1^{(t-1)}, X_2^{(t-1)}, \ldots, X_p^{(t-1)})}\right\}$.
   iii. With probability α_1 set $X_1^{(t)} = X_1$, otherwise set $X_1^{(t)} = X_1^{(t-1)}$.

...

j. i. Draw $X_j \sim q_j(\cdot|X_1^{(t)}, \ldots, X_{j-1}^{(t)}, X_j^{(t-1)}, \ldots, X_p^{(t-1)})$.
   ii. Compute $\alpha_j = \min\left\{1, \frac{f(X_1^{(t)}, \ldots, X_{j-1}^{(t)}, X_j, X_{j+1}^{(t-1)}, \ldots, X_p^{(t-1)}) \cdot q_j(X_j^{(t-1)}|X_1^{(t)}, \ldots, X_{j-1}^{(t)}, X_j, X_{j+1}^{(t-1)}, \ldots, X_p^{(t-1)})}{f(X_1^{(t)}, \ldots, X_{j-1}^{(t)}, X_j^{(t-1)}, X_{j+1}^{(t-1)}, \ldots, X_p^{(t-1)}) \cdot q_j(X_j|X_1^{(t)}, \ldots, X_{j-1}^{(t)}, X_j^{(t-1)}, X_{j+1}^{(t-1)}, \ldots, X_p^{(t-1)})}\right\}$.
   iii. With probability α_j set $X_j^{(t)} = X_j$, otherwise set $X_j^{(t)} = X_j^{(t-1)}$.

...

p. i. Draw $X_p \sim q_p(\cdot|X_1^{(t)}, \ldots, X_{p-1}^{(t)}, X_p^{(t-1)})$.
   ii. Compute $\alpha_p = \min\left\{1, \frac{f(X_1^{(t)}, \ldots, X_{p-1}^{(t)}, X_p) \cdot q_p(X_p^{(t-1)}|X_1^{(t)}, \ldots, X_{p-1}^{(t)}, X_p)}{f(X_1^{(t)}, \ldots, X_{p-1}^{(t)}, X_p^{(t-1)}) \cdot q_p(X_p|X_1^{(t)}, \ldots, X_{p-1}^{(t)}, X_p^{(t-1)})}\right\}$.
   iii. With probability α_p set $X_p^{(t)} = X_p$, otherwise set $X_p^{(t)} = X_p^{(t-1)}$.

The corresponding random sweep algorithm (corresponding to K^+) is: Starting with $\mathbf{X}^{(0)} = (X_1^{(0)}, \ldots, X_p^{(0)})$ iterate

1. Draw an index j from a distribution on $\{1, \ldots, p\}$ (e.g. uniform)
2. Draw $X_j \sim q_j(\cdot|X_1^{(t-1)}, \ldots, X_p^{(t-1)})$.
3. Compute $\alpha_j = \min\left\{1, \frac{f(X_1^{(t-1)}, \ldots, X_{j-1}^{(t-1)}, X_j, X_{j+1}^{(t-1)}, \ldots, X_p^{(t-1)}) \cdot q_j(X_j^{(t-1)}|X_1^{(t-1)}, \ldots, X_{j-1}^{(t-1)}, X_j, X_{j+1}^{(t-1)}, \ldots, X_p^{(t-1)})}{f(X_1^{(t-1)}, \ldots, X_{j-1}^{(t-1)}, X_j^{(t-1)}, X_{j+1}^{(t-1)}, \ldots, X_p^{(t-1)}) \cdot q_j(X_j|X_1^{(t-1)}, \ldots, X_{j-1}^{(t-1)}, X_j^{(t-1)}, X_{j+1}^{(t-1)}, \ldots, X_p^{(t-1)})}\right\}$.
4. With probability α_j set $X_j^{(t)} = X_j$, otherwise set $X_j^{(t)} = X_j^{(t-1)}$.
5. Set $X_\iota^{(t)} := X_\iota^{(t-1)}$ for all $\iota \neq j$.

Note the similarity to the Gibbs sampler. Indeed, the Gibbs sampler is a special case of a one-at-a-time Metropolis-Hastings algorithm as the following remark shows. ◁

Remark 5.2. The Gibbs sampler for a p-dimensional distribution is a special case of a one-at-a-time Metropolis-Hasting algorithm: the (systematic scan) Gibbs sampler (Algorithm 4.1) is a cycle of p kernels, whereas the random scan Gibbs sampler (Algorithm 4.2) is a mixture of these kernels. The proposal q_j corresponding to the j-th kernel consists of drawing $X_j^{(t)} \sim f_{X_j|X_{-j}}$. The corresponding probability of acceptance is uniformly equal to 1.

Proof. The update of the j-th component of the Gibbs sampler consists of sampling from $X_j|X_{-j}$, i.e. it has the proposal

$$q_j(x_j|\mathbf{x}^{(t-1)}) = f_{X_j|X_{-j}}(x_j|x_1^{(t)}, \ldots, x_{j-1}^{(t)}, x_{j+1}^{(t-1)}, \ldots, x_p^{(t-1)}).$$

We obtain for the j-th kernel that

$$
\begin{aligned}
& \frac{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_j,x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})q_j(x_j^{(t-1)}|x_1^{(t)},\dots,x_{j-1}^{(t)},x_j,x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})}{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_j^{(t-1)},x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})q_j(x_j|x_1^{(t)},\dots,x_{j-1}^{(t)},x_j^{(t-1)},x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})} \\
= & \frac{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_j,x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})f_{X_j|X_{-j}}(x_j^{(t-1)}|x_1^{(t)},\dots,x_{j-1}^{(t)},x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})}{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_j^{(t-1)},x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})f_{X_j|X_{-j}}(x_j|x_1^{(t)},\dots,x_{j-1}^{(t)},x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})} \\
= & \frac{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_j,x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})\frac{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_j^{(t-1)},x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})}{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})}}{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_j^{(t-1)},x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})\frac{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_j,x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})}{f(x_1^{(t)},\dots,x_{j-1}^{(t)},x_{j+1}^{(t-1)},\dots,x_p^{(t-1)})}} \\
= & 1,
\end{aligned}
$$

thus $\alpha_j \equiv 1$. □

As explained above, the composite kernels K^+ and K° have the invariant distribution f, if all kernels $K^{(j)}$ have f as invariant distribution. Similarly, it is sufficient for the irreducibility of the kernels K^+ and K° that all kernels $K^{(j)}$ are irreducible. This is however not a very useful condition, nor is it a necessary condition. Often, some of the kernels $K^{(j)}$ focus on certain subspaces, and thus cannot be irreducible for the entire space. The kernels $K^{(j)}$ corresponding to the Gibbs sampler are *not* irreducible themselves: the j-th Gibbs kernel $K^{(j)}$ only updates X_j, not the other X_ι ($\iota \neq j$).

6. Diagnosing convergence

6.1 Practical considerations

The theory of Markov chains you saw in the first part of the course guarantees that a Markov chain that is irreducible and has invariant distribution f converges to the invariant distribution. The ergodic theorems 4.2 and 5.1 allow for approximating expectations $\mathbb{E}_f(\varphi(\mathbf{X}))$ by their the corresponding empirical means

$$\frac{1}{T}\sum_{t=1}^{T}\varphi(\mathbf{X}^{(t)}) \longrightarrow \mathbb{E}_f(\varphi(\mathbf{X}))$$

using the *entire* chain. In practice, however, often only a subset of the chain $(\mathbf{X}^{(t)})_t$ is used:

Burn-in Depending on how $\mathbf{X}^{(0)}$ is chosen, the distribution of $(\mathbf{X}^{(t)})_t$ for small t might still be far from the stationary distribution f. Thus it might be beneficial to discard the first iterations $\mathbf{X}^{(t)}$, $t = 1, \ldots, T_0$. This early stage of the sampling process is often referred to as *burn-in* period. How large a value of T_0 should be chosen depends upon how rapidly mixing the Markov chain $(\mathbf{X}^{(t)})_t$ is. Figure 6.1 illustrates the idea of a burn-in period.

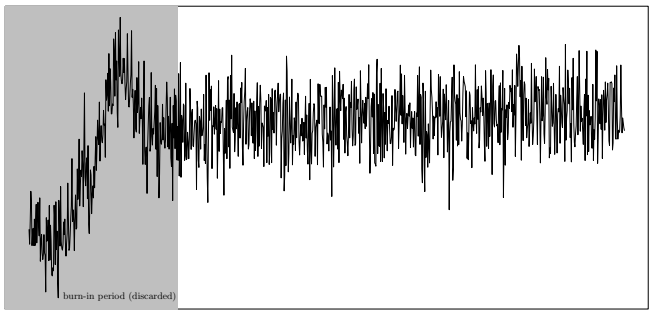


Figure 6.1. Illustration of the idea of a burn-in period.

Thinning Markov chain Monte Carlo methods typically yield a Markov chain with positive autocorrelation, i.e. $\rho(X_k^{(t)}, X_k^{(t+\tau)})$ is positive for small τ. This suggests building a subchain by only keeping every m-th value ($m > 1$), i.e. we consider a Markov chain $(\mathbf{Y}^{(t)})_t$ with $\mathbf{Y}^{(t)} = \mathbf{X}^{(m\cdot t)}$ instead of $(\mathbf{X}^{(t)})_t$. If the correlation $\rho(\mathbf{X}^{(t)}, \mathbf{X}^{(t+\tau)})$ decreases monotonically in τ, then

$$\rho(Y_k^{(t)}, Y_k^{(t+\tau)}) = \rho(X_k^{(t)}, X_k^{(t+m\cdot\tau)}) < \rho(X_k^{(t)}, X_k^{(t+\tau)}),$$

i.e. the thinned chain $(\mathbf{Y}^{(t)})_t$ exhibits less autocorrelation than the original chain $(\mathbf{X}^{(t)})_t$. Thus thinning can be seen as a technique for reducing the autocorrelation, however at the price of yielding a chain $(\mathbf{Y}^{(t)})_{t=1,\dots\lfloor T/m\rfloor}$, whose length is reduced to $(1/m)$-th of the length of the original chain $(\mathbf{X}^{(t)})_{t=1,\dots,T}$. Even though thinning is very popular, it cannot be justified in terms of minimizing the variance of resulting estimates when the objective is estimating $\mathbb{E}_f(\varphi(\mathbf{X}))$, as the following lemma shows.

Lemma 6.1. *Let $(\mathbf{X}^{(t)})_{t=1,\dots,T}$ be a sequence of random variables obtained from an f-invariant Markov chain at stationarity (i.e. with $\mathbf{X}^{(t)} \sim f$) and let $(\mathbf{Y}^{(t)})_{t=1,\dots,\lfloor T/m\rfloor}$ be a second sequence defined by $\mathbf{Y}^{(t)} := \mathbf{X}^{(m\cdot t)}$. If $\mathrm{Var}_f(\varphi(\mathbf{X}^{(t)})) < +\infty$, then*

$$\mathrm{Var}\left(\frac{1}{T}\sum_{t=1}^{T}\varphi(\mathbf{X}^{(t)})\right) \leq \mathrm{Var}\left(\frac{1}{\lfloor T/m\rfloor}\sum_{t=1}^{\lfloor T/m\rfloor}\varphi(\mathbf{Y}^{(t)})\right).$$

Proof. To simplify the proof we assume that T is divisible by m, i.e. $T/m \in \mathbb{N}$. Using

$$\sum_{t=1}^{T}\varphi(\mathbf{X}^{(t)}) = \sum_{\tau=1}^{m}\sum_{t=0}^{T/m-1}\varphi(\mathbf{X}^{(t\cdot m+\tau)})$$

and

$$\mathrm{Var}\left(\sum_{t=0}^{T/m-1}\varphi(\mathbf{X}^{(t\cdot m+\tau_1)})\right) = \mathrm{Var}\left(\sum_{t=0}^{T/m-1}\varphi(\mathbf{X}^{(t\cdot m+\tau_2)})\right)$$

for $\tau_1, \tau_2 \in \{1,\dots,m\}$, we obtain that

$$\begin{aligned}
\mathrm{Var}\left(\sum_{t=1}^{T}\varphi(\mathbf{X}^{(t)})\right) &= \mathrm{Var}\left(\sum_{\tau=1}^{m}\sum_{t=0}^{T/m-1}\varphi(\mathbf{X}^{(t\cdot m+\tau)})\right) \\
&= m\cdot\mathrm{Var}\left(\sum_{t=0}^{T/m-1}\varphi(\mathbf{X}^{(t\cdot m)})\right) + \sum_{\eta\neq\tau=1}^{m}\underbrace{\mathrm{Cov}\left(\sum_{t=0}^{T/m-1}\varphi(\mathbf{X}^{(t\cdot m+\eta)}),\ \sum_{t=0}^{T/m-1}\varphi(\mathbf{X}^{(t\cdot m+\tau)})\right)}_{\leq \mathrm{Var}\left(\sum_{t=1}^{T/m}\varphi(\mathbf{X}^{(t\cdot m)})\right)} \\
&\leq m^2\cdot\mathrm{Var}\left(\sum_{t=0}^{T/m-1}\varphi(\mathbf{X}^{(t\cdot m)})\right) = m^2\cdot\mathrm{Var}\left(\sum_{t=1}^{T/m}\varphi(\mathbf{Y}^{(t)})\right).
\end{aligned}$$

Thus

$$\mathrm{Var}\left(\frac{1}{T}\sum_{t=1}^{T}\varphi(\mathbf{X}^{(t)})\right) = \frac{1}{T^2}\mathrm{Var}\left(\sum_{t=1}^{T}\varphi(\mathbf{X}^{(t)})\right) \leq \frac{m^2}{T^2}\mathrm{Var}\left(\sum_{t=1}^{T/m}\varphi(\mathbf{Y}^{(t)})\right) = \mathrm{Var}\left(\frac{1}{T/m}\sum_{t=1}^{T/m}\varphi(\mathbf{Y}^{(t)})\right).$$

□

Thinning can, however, be useful for other reasons. If storage is limited it may not be possible to store all of an arbitrarily long chain; in this context, it can be much better to store the thinned skeletion of a long chain than to consider the entire sample path of a shorter chain. Furthermore, it can be easier to assess the convergence of the thinned chain $(\mathbf{Y}^{(t)})_t$ as opposed to entire chain $(\mathbf{X}^{(t)})_t$.

6.2 Tools for monitoring convergence

Although the theory presented in the preceding chapters guarantees the convergence of the Markov chains to the required distributions, this does not imply that a *finite* sample from such a chain yields a good

approximation to the target distribution. As with all approximating methods this must be confirmed in practice.

This section tries to give a brief overview over various approaches to diagnosing convergence. A more detailed review with many practical examples can be diagnofound in (Guihennec-Jouyaux et al., 1998) or (Robert and Casella, 2004, Chapter 12). There is an R package (`CODA`) that provides a vast selection of tools for diagnosing convergence.

Diagnosing convergence is an art. The techniques presented in the following are no more than exploratory tools that help you judge whether the chain has reached its stationary regime. This section contains several cautionary examples where the different tools for diagnosing convergence fail.

Broadly speaking, convergence assessment can be split into the following three categories, each of which considers the assessment of a different aspect of convergence:

Convergence to the target distribution. The first, and most important, question is whether $(\mathbf{X}^{(t)})_t$ provides a "dependent sample" from the target distribution? In order to answer this question we need to assess ...
- whether $(\mathbf{X}^{(t)})_t$ has reached a stationary regime, so that $\mathbf{X}^{(t)}$ is marginally distributed according to (or at least close to) the target distribution for values of t within the finite range available, and
- whether $(\mathbf{X}^{(t)})_t$ covers the entire support of the target distribution.

Convergence of the averages. Does $\sum_{t=1}^{T} \varphi(\mathbf{X}^{(t)})/T$ provide a good approximation to the expectation $\mathbb{E}_f(\varphi(\mathbf{X}))$ under the target distribution?

Comparison to i.i.d. sampling. How much information is contained in the sample from the Markov chain compared to i.i.d. sampling?

6.2.1 Basic plots

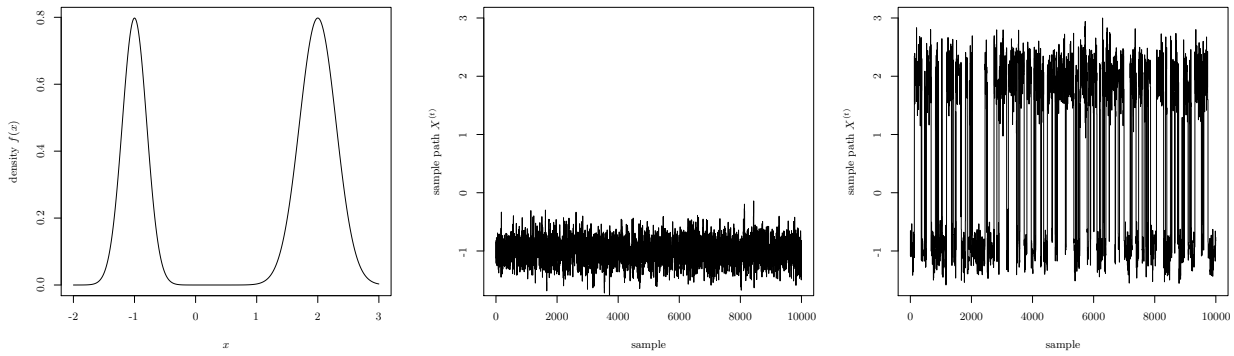
The most basic approach to diagnosing the output of a Markov Chain Monte Carlo algorithm is to plot the sample path $(\mathbf{X}^{(t)})_t$ as in Figures 4.4 (b) (c) and 4.5 (b) (c). Note that the convergence of $(\mathbf{X}^{(t)})_t$ is in distribution, i.e. the sample path is *not* supposed to converge to a single value. Ideally, the plot should be oscillating very fast and show very little structure or trend (like for example Figure 4.4). The smoother such a plot seems, the slower mixing the resulting chain is (see Figure 4.5 for an illustration of a slowly mixing trajectory).

Note however that this plot suffers from the "you've only seen where you've been" problem. It is impossible to see from a plot of the sample path whether the chain has explored the entire support of the distribution (without additional information).

Example 6.1 (A simple mixture of two Gaussians). Consider sampling from a mixture of two well-separated Gaussians

$$f(x) = 0.4 \cdot \phi_{(-1,0.2^2)}(x) + 0.6 \cdot \phi_{(2,0.3^2)}(x)$$

(see Figure 6.2 (a) for a plot of the density) using a random walk Metropolis algorithm with an $\mathsf{N}(0, \mathrm{Var}(\varepsilon))$ increment distribution. If we choose the proposal variance $\mathrm{Var}(\varepsilon)$ too small, we only sample from one component of the mixture, not from the mixture itself. Figure 6.2 shows the sample paths of for two choices of $\mathrm{Var}(\varepsilon)$: $\mathrm{Var}(\varepsilon) = 0.4^2$ and $\mathrm{Var}(\varepsilon) = 1.2^2$. The first choice of $\mathrm{Var}(\varepsilon)$ is too small: the chain is very likely to remain in one of the two modes of the distribution. Note that it is impossible to tell from Figure 6.2 (b) alone that the chain has not explored the entire support of the target. ◁



(a) Density $f(x)$

(b) Sample path of a random walk Metropolis algorithm with proposal variance $\mathrm{Var}(\varepsilon) = 0.4^2$

(c) Sample path of a random walk Metropolis algorithm with proposal variance $\mathrm{Var}(\varepsilon) = 1.2^2$

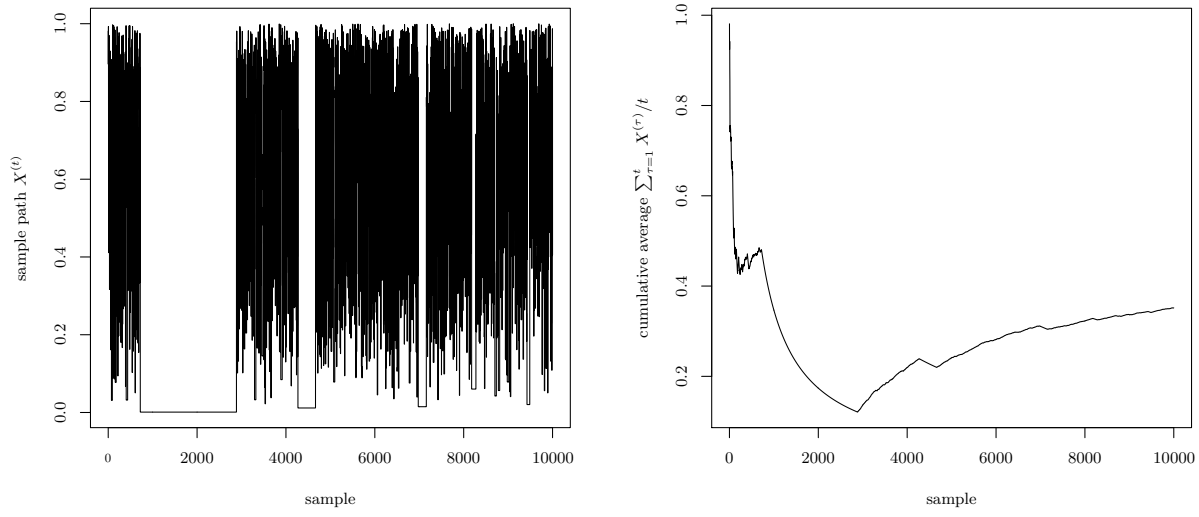
Figure 6.2. Density of the mixture distribution with two random walk Metropolis samples using two different variances $\mathrm{Var}(\varepsilon)$ of the proposal.

In order to diagnose the convergence of sample averages, one can look at a plot of the cumulative averages $(\sum_{\tau=1}^{t} \varphi(X^{(\tau)})/t)_t$. Note that the convergence of the cumulative averages is — as the ergodic theorems suggest — to a value $(\mathbb{E}_f(\varphi(\mathbf{X}))$. Figures 4.3, and 5.3 (b) shows plots of the cumulative averages. An alternative to plotting the cumulative means is using the so-called CUSUMs $(\bar{\varphi(X)} - \sum_{\tau=1}^{t} \varphi(X_j^{(\tau)})/t)_t$ with $\bar{X}_j = \sum_{\tau=1}^{T} \varphi(X_j^{(\tau)})/T$, which is nothing other than the difference between the cumulative averages and the estimate of the limit $\mathbb{E}_f(\varphi(\mathbf{X}))$.

Example 6.2 (A pathological generator for the Beta distribution). The following MCMC algorithm (for details, see Robert and Casella, 2004, Problem 7.5) yields a sample from the Beta$(\alpha, 1)$ distribution. Starting with any $X^{(0)}$ iterate for $t = 1, 2, \ldots$

1. With probability $1 - X^{(t-1)}$, set $X^{(t)} = X^{(t-1)}$.
2. Otherwise draw $X^{(t)} \sim \text{Beta}(\alpha + 1, 1)$.

This algorithm yields a very slowly converging Markov chain, to which no central limit theorem applies. This slow convergence can be seen in a plot of the cumulative means (Figure 6.3 (b)). ◁



(a) Sample path $X^{(t)}$

(b) Cumulative means $\sum_{\tau=1}^{t} X^{(\tau)}/t$

Figure 6.3. Sample paths and cumulative means obtained for the pathological Beta generator.

Note that it is impossible to tell from a plot of the cumulative means whether the Markov chain has explored the entire support of the target distribution.

6.2.2 Non-parametric tests of stationarity

A variety of nonparametric tests can be employed to establish whether the samples from a Markov chain behave in particular ways. This section presents an illustration of the (informal, approximate) use of the Kolmogorov-Smirnov test to assess whether there is evidence that a Markov chain has not yet reached stationarity.

In its simplest version, it is based on splitting the chain into three parts: $(\mathbf{X}^{(t)})_{t=1,\ldots,\lfloor T/3\rfloor}$, $(\mathbf{X}^{(t)})_{t=\lfloor T/3\rfloor+1,\ldots,2\lfloor T/3\rfloor}$, and $(\mathbf{X}^{(t)})_{t=2\lfloor T/3\rfloor+1,\ldots,T}$. The first block is considered to be the burn-in period. If the Markov chain has reached its stationary regime after $\lfloor T/3\rfloor$ iterations, the second and third block should be from the same distribution. Thus we should be able to tell whether the chain has converged by comparing the distribution of $(\mathbf{X}^{(t)})_{t=\lfloor T/3\rfloor+1,\ldots,2\lfloor T/3\rfloor}$ to the one of $(\mathbf{X}^{(t)})_{t=2\lfloor T/3\rfloor+1,\ldots,T}$ using suitable nonparametric two-sample tests. One such test is the Kolmogorov-Smirnov test.

Definition 6.1. *Kolmogorov-Smirnov Statistic The two-sample Kolmogorov-Smirnov test for comparing two i.i.d. samples $Z_{1,1},\ldots,Z_{1,n}$ and $Z_{2,1},\ldots,Z_{2,n}$ is based on comparing their empirical CDFs*

$$\hat{F}_k(z) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{(-\infty,z]}(Z_{k,i}).$$

The Kolmogorov-Smirnov test statistic is the maximum difference between the two empirical CDFs:

$$K = \sup_{z\in\mathbb{R}} |\hat{F}_1(z) - \hat{F}_2(z)|.$$

For $n\to\infty$ the CDF of $\sqrt{n}\cdot K$ converges to the CDF

$$R(k) = 1 - \sum_{i=1}^{+\infty} (-1)^{i-1}\exp(-2i^2k^2).$$

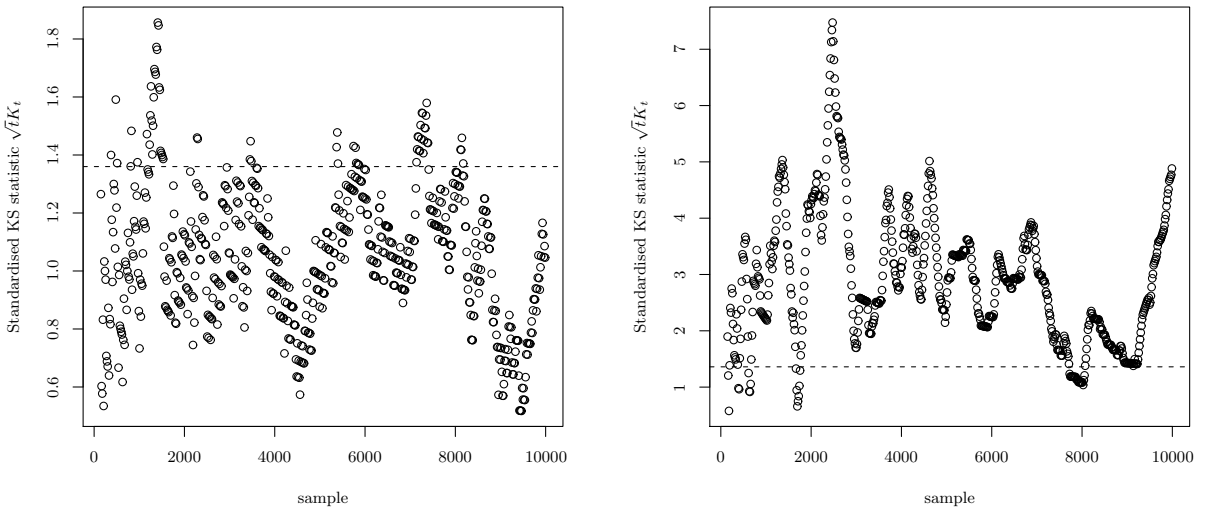
As the Kolmogorov-Smirnov test is designed for i.i.d. samples, we do not apply it to the $(\mathbf{X}^{(t)})_t$ directly, but to a thinned chain $(\mathbf{Y}^{(t)})_t$ with $\mathbf{Y}^{(t)} = \mathbf{X}^{(m\cdot t)}$: the thinned chain is less correlated and thus closer to being an i.i.d. sample. This, of course, still formally violates the conditions under which the Kolmogorov-Smirnov test is exact but one can still hope to obtain useful information when the conditions are *close* to being satisfied.

We can now compare the distribution of $(\mathbf{Y}^{(t)})_{t=\lfloor T/(3m)\rfloor+1,\ldots,2\lfloor T/(3m)\rfloor}$ to the one of $(\mathbf{Y}^{(t)})_{t=2\lfloor T/(3m)\rfloor+1,\ldots,\lfloor T/m\rfloor}$ using the Kolmogorov-Smirnov statistic .

$$K = \sup_{x\in\mathbb{R}} \left| \hat{F}_{(\mathbf{Y}^{(t)})_{t=\lfloor T/(3m)\rfloor+1,\ldots,2\lfloor T/(3m)\rfloor}}(x) - \hat{F}_{(\mathbf{Y}^{(t)})_{t=2\lfloor T/(3m)\rfloor+1,\ldots,\lfloor T/m\rfloor}}(x) \right|.$$

As the thinned chain is not an i.i.d. sample, we cannot use the Kolmogorov-Smirnov test as a formal statistical test (besides, we would run into problems of multiple testing). However, we can use it as an informal tool by monitoring the standardised statistic $\sqrt{t}K_t$ as a function of t., where K_t denotes the Kolmogorov-Smirnov statistic obtained from the sample consisting of the first t observations only. If a significant proportion of the values of this standardised statistic are above the corresponding quantile of the asymptotic distribution, it is safe to assume that the chain has not yet reached its stationary regime.

Example 6.3 (Gibbs sampling from a bivariate Gaussian (continued)). In this example we consider sampling from a bivariate Gaussian distribution, once with $\rho(X_1, X_2) = 0.3$ (as in example 4.4) and once with $\rho(X_1, X_2) = 0.99$ (as in example 4.5). The former leads a fast mixing chain, the latter a very slowly mixing chain. Figure 6.4 shows the plots of the standardised Kolmogorov-Smirnov statistic. It suggests that the sample size of 10,000 is large enough for the low-correlation setting, but not large enough for the high-correlation setting. ◁



(a) $\rho(X_1, X_2) = 0.3$ (b) $\rho(X_1, X_2) = 0.99$

Figure 6.4. Standardised Kolmogorov-Smirnov statistic for $X_1^{(5 \cdot t)}$ from the Gibbs sampler from the bivariate Gaussian for two different correlations.

Note that this use of the Kolmogorov-Smirnov test suffers from the "you've only seen where you've been" problem, as it is based on comparing $(\mathbf{Y}^{(t)})_{t=\lfloor T/(3m) \rfloor + 1, \ldots, 2\lfloor T/(3m) \rfloor}$ and $(\mathbf{Y}^{(t)})_{t=2\lfloor T/(3m) \rfloor + 1, \ldots, \lfloor T/m \rfloor}$. A plot of the Kolmogorov-Smirnov statistic for the chain with $\mathrm{Var}(\varepsilon) = 0.4$ from Example 6.1 would not reveal anything unusual.

6.2.3 Riemann sums and control variates

A simple tool for diagnosing convergence of a one-dimensional Markov chain can be based on the fact that

$$\int_E f(x)\, dx = 1.$$

We can estimate this integral using the Riemann sum

$$\sum_{t=2}^{T} (X^{[t]} - X^{[t-1]}) f(X^{[t]}), \tag{6.1}$$

where $X^{[1]} \leq \ldots \leq X^{[T]}$ is the ordered sample from the Markov chain. If the Markov chain has explored all the support of f, then (6.1) should be around 1 (as it is an estimate of the integral of a probability density). Note that this method, often referred to as Riemann sums (Philippe and Robert, 2001), requires that the density f is known inclusive of normalisation constants (and thus to apply this technique to a univariate marginal of a multivariate problem would require that at least one univariate marginal of the target distributions is known exactly).

Example 6.4 (A simple mixture of two Gaussians (continued)). In Example 6.1 we considered two random-walk Metropolis algorithms: one ($\mathrm{Var}(\varepsilon) = 0.4^2$) failed to explore the entire support of the target distribution, whereas the other one ($\mathrm{Var}(\varepsilon) = 1.2^2$) managed to. The corresponding Riemann sums are 0.598 and 1.001, clearly indicating that the first algorithm does not explore the entire support. ◁

Riemann sums can be seen as a special case of a technique called *control variates*. The idea of control variates is essentially to compare several ways of estimating the same quantity using the same collection of samples. If the different estimates disagree, the chain has not yet converged. Note that the technique of control variates is only useful if the different estimators converge about as fast as the quantity of interest — otherwise we would obtain an overly optimistic, or an overly conservative estimate of whether the chain has converged. In the special case of the Riemann sum we compare two quantities: the constant 1 and the Riemann sum (6.1).

6.2.4 Comparing multiple chains

A family of convergence diagnostics (see e.g. Gelman and Rubin, 1992; Brooks and Gelman, 1998) is based on running $L > 1$ chains — which we will denote by $(\mathbf{X}^{(1,t)})_t, \ldots, (\mathbf{X}^{(L,t)})_t$ — with overdispersed (in the sense that the variance of the starting values should be larger than the variance of the target distribution) starting values $\mathbf{X}^{(1,0)}, \ldots, \mathbf{X}^{(L,0)}$. These starting values should in principle be chosen to give reasonable coverage of the support of the target distribution.

All L chains should converge to the same distribution, so comparing the plots described in Section 6.2.1 for the L different chains should not reveal any difference. A more formal approach to diagnosing whether the L chains are all from the same distribution can be based on comparing the inter-quantile distances.

We can estimate the inter-quantile distances in two ways. The first consists of estimating the inter-quantile distance for each of the L chain and averaging over these results, i.e. our estimate is $\sum_{l=1}^{L} \delta_\gamma^{(L,\cdot)}/L$, where $\delta_\gamma^{(L,\cdot)}$ is the distance between the γ and $(1-\gamma)$ quantile of the l-th chain$(X_k^{(l,t)})_t$. Alternatively, we can pool the data first, and then compute the distance between the γ and $(1-\gamma)$ quantile of the pooled data. If all chains are a sample from the same distribution, both estimates should be roughly the same, so their ratio

$$\hat{S}_\gamma^{\text{interval}} = \frac{\sum_{l=1}^{L} \delta_\gamma^{(l)}/L}{\delta_\gamma^{(\cdot)}}$$

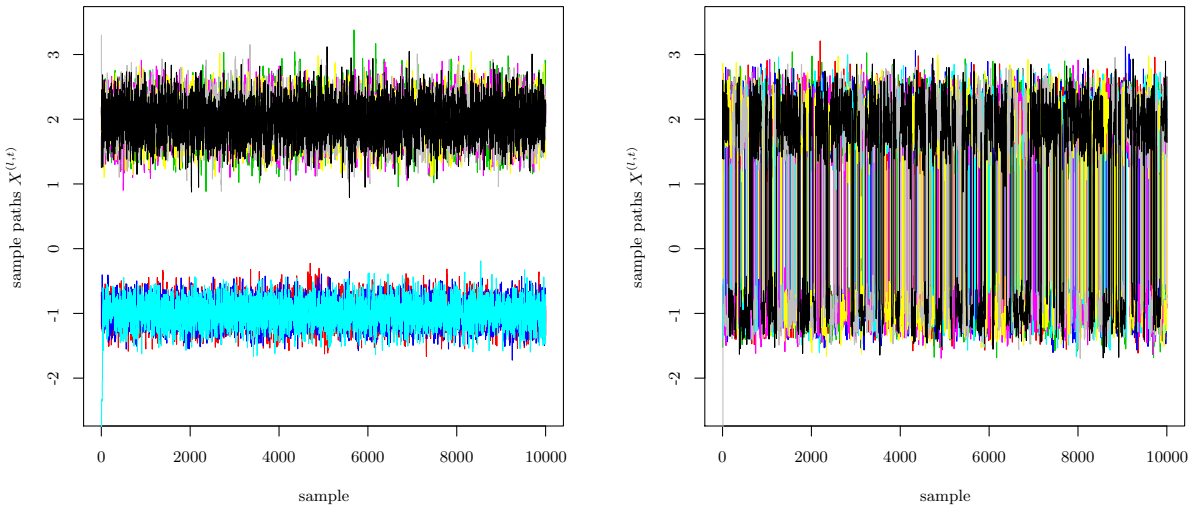
can be used as a tool to diagnose whether all chains sampled from the same distribution, in which case the ratio should be around 1.

Alternatively, one could compare the variances within the L chains to the pooled estimate of the variance (see Brooks and Gelman, 1998, for more details).

Example 6.5 (A simple mixture of two Gaussians (continued)). In the example of the mixture of two Gaussians we will consider $L = 8$ chains initialised with iid samples from a $\mathsf{N}(0, 10^2)$ distribution. Figure 6.5 shows the sample paths of the 8 chains for both choices of $\mathrm{Var}(\varepsilon)$. The corresponding values of $\hat{S}_{0.05}^{\text{interval}}$ are:

$$\mathrm{Var}(\varepsilon) = 0.4^2 \quad : \quad \hat{S}_{0.05}^{\text{interval}} = \frac{0.9789992}{3.630008} = 0.2696962$$
$$\mathrm{Var}(\varepsilon) = 1.2^2 \quad : \quad \hat{S}_{0.05}^{\text{interval}} = \frac{3.634382}{3.646463} = 0.996687.$$

◁



(a) $\mathrm{Var}(\varepsilon) = 0.4^2$ (b) $\mathrm{Var}(\varepsilon) = 1.2^2$

Figure 6.5. Comparison of the sample paths for $L = 8$ chains for the mixture of two Gaussians.

Note that this method depends crucially on the choice of initial values $\mathbf{X}^{(1,0)}, \ldots, \mathbf{X}^{(L,0)}$, and thus can easily fail, as the following example shows.

Example 6.6 (Witch's hat distribution). Consider a distribution with the following density:

$$f(x_1, x_2) \propto \begin{cases} (1-\delta)\phi_{(\boldsymbol{\mu}, \sigma^2 \cdot \mathbb{I})}(x_1, x_2) + \delta & \text{if } x_1, x_2 \in (0,1) \\ 0 & \text{else,} \end{cases}$$

which is a mixture of a Gaussian and a uniform distribution, both truncated to $[0,1] \times [0,1]$. Figure 6.6 illustrates the density. For very small σ^2, the Gaussian component is concentrated in a very small area around $\boldsymbol{\mu}$.

The conditional distribution of $X_1 | X_2$ is

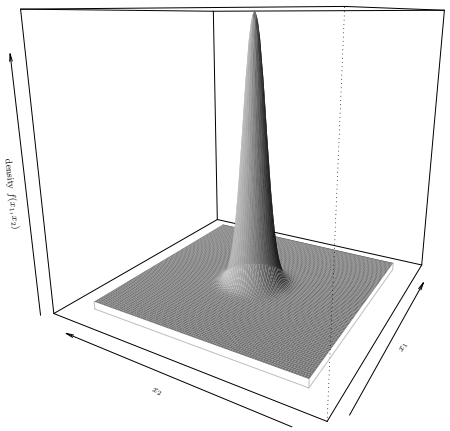
$$f(x_1 | x_2) = \begin{cases} (1-\delta_{x_2})\phi_{(\boldsymbol{\mu}, \sigma^2 \cdot \mathbb{I})}(x_1, x_2) + \delta_{x_2} & \text{for } x_1 \in (0,1) \\ 0 & \text{otherwise} \end{cases}$$

with $\delta_{x_2} = \dfrac{\delta}{\delta + (1-\delta)\phi_{(\mu_2, \sigma^2)}(x_2)}$.

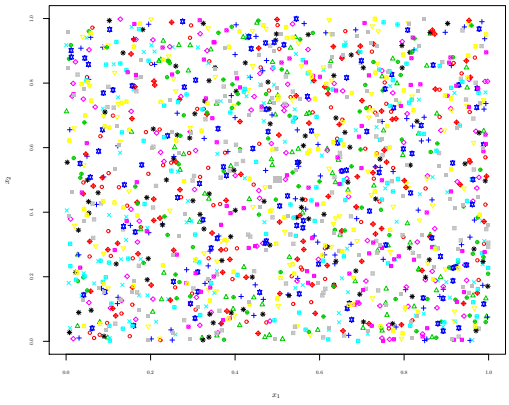
Assume we want to estimate $\mathbb{P}(0.49 < X_1, X_2 \leq 0.51)$ for $\delta = 10^{-3}$, $\boldsymbol{\mu} = (0.5, 0.5)'$, and $\sigma = 10^{-5}$ using a Gibbs sampler. Note that 99.9% of the mass of the distribution is concentrated in a very small area around $(0.5, 0.5)$, i.e. $\mathbb{P}(0.49 < X_1, X_2 \leq 0.51) \approx 0.999$.

Nonetheless, it is very unlikely that the Gibbs sampler visits this part of the distribution. This is due to the fact that unless x_2 (or x_1) is very close to μ_2 (or μ_1), δ_{x_2} (or δ_{x_1}) is almost 1, i.e. the Gibbs sampler only samples from the uniform component of the distribution. Figure 6.6 shows the samples obtained from 15 runs of the Gibbs sampler (first 100 iterations only) all using different initialisations. On average only 0.04% of the sampled values lie in $(0.49, 0.51) \times (0.49, 0.51)$ yielding an estimate of $\hat{\mathbb{P}}(0.49 < X_1, X_2 \leq 0.51) = 0.0004$ (as opposed to $\mathbb{P}(0.49 < X_1, X_2 \leq 0.51) = 0.999$).

It is, however, close to impossible to detect this problem with any technique based on multiple initialisations. The Gibbs sampler shows this behaviour for practically all starting values. In Figure 6.6 all 15 starting values yield a Gibbs sampler that is stuck in the "brim" of the witch's hat and thus misses 99.9% of the probability mass of the target distribution. ◁



(a) Density for $\delta = 0.2$, $\boldsymbol{\mu} = (0.5, 0.5)'$, and $\sigma = 0.05$



(b) First 100 values from 15 samples using different starting values.
($\delta = 10^{-3}$, $\boldsymbol{\mu} = (0.5, 0.5)'$, and $\sigma = 10^{-5}$)

Figure 6.6. Density and sample from the witch's hat distribution.

6.2.5 Comparison to i.i.d. sampling and the effective sample size

MCMC algorithms typically yield a positively correlated sample $(\mathbf{X}^{(t)})_{t=1,\ldots,T}$, which contains less information than an i.i.d. sample of size T. If the $(\mathbf{X}^{(t)})_{t=1,\ldots,T}$ are positively correlated, then the variance of the average

$$\operatorname{Var}\left(\frac{1}{T}\sum_{t=1}^{T}\varphi(\mathbf{X}^{(t)})\right) \tag{6.2}$$

is larger than the variance we would obtain from an i.i.d. sample, which is $\operatorname{Var}(\varphi(\mathbf{X}^{(t)}))/T$.

The effective sample size (ESS) attempts to quantify the loss of information caused by this positive correlation. The effective sample size is the size an i.i.d. would have to have in order to obtain the same variance (6.2) as the estimate from the Markov chain $(\mathbf{X}^{(t)})_{t=1,\ldots,T}$.

As the exact computation of this quantity is generally impossible, a number of simplifying approximations are usually made in order to obtain a computationally tractable proxy for this quantity. Slightly confusingly, the *approximate* equivalent independent sample size arrived at following this chain of approximations is also referred to as the ESS.

For illustrative purposes, we describe here one particularly simple way to obtain an approximation of the effective sample size; the much more robust procedure of Gong and Flegal (2016) is implemented in the mcmcse R package and would be preferred in real applications.

In order to compute the variance (6.2) we make the simplifying assumption that $(\varphi(\mathbf{X}^{(t)}))_{t=1,\ldots,T}$ is from a second-order stationary time series, i.e. $\operatorname{Var}(\varphi(\mathbf{X}^{(t)})) = \sigma^2$, and $\rho(\varphi(\mathbf{X}^{(t)}), \varphi(\mathbf{X}^{(t+\tau)})) = \rho(\tau)$. Then

$$\begin{aligned}
\operatorname{Var}\left(\frac{1}{T}\sum_{t=1}^{T}\varphi(\mathbf{X}^{(t)})\right) &= \frac{1}{T^2}\left(\sum_{t=1}^{T}\underbrace{\operatorname{Var}(\varphi(\mathbf{X}^{(t)}))}_{=\sigma^2} + 2\sum_{1\le s<t\le T}\underbrace{\operatorname{Cov}(\varphi(\mathbf{X}^{(s)}),\varphi(\mathbf{X}^{(t)}))}_{=\sigma^2\cdot\rho(t-s)}\right) \\
&= \frac{\sigma^2}{T^2}\left(T + 2\sum_{\tau=1}^{T-1}(T-\tau)\rho(\tau)\right) = \frac{\sigma^2}{T}\left(1 + 2\sum_{\tau=1}^{T-1}\left(1-\frac{\tau}{T}\right)\rho(\tau)\right).
\end{aligned}$$

If $\sum_{\tau=1}^{+\infty}|\rho(\tau)| < +\infty$, then we can obtain from the dominated convergence theorem (see e.g. Brockwell and Davis (1991, Theorem 7.1.1) for details) that

$$T \cdot \mathrm{Var}\left(\frac{1}{T}\sum_{t=1}^{T}\varphi(\mathbf{X}^{(t)})\right) \longrightarrow \sigma^2\left(1+2\sum_{\tau=1}^{+\infty}\rho(\tau)\right)$$

as $T \to \infty$. Note that the variance of the simple Monte Carlo esimate of $\mathbb{E}_f[\varphi(X)]$ would be $\sigma^2/T_{\mathrm{ESS}}$ if we were to use an i.i.d. sample of size T_{ESS}. We can now obtain the effective sample size T_{ESS} by equating these two variances and solving for T_{ESS}, yielding

$$T_{\mathrm{ESS}} = \frac{1}{1+2\sum_{\tau=1}^{+\infty}\rho(\tau)} \cdot T.$$

If we assume that $(\varphi(\mathbf{X}^{(t)}))_{t=1,\ldots,T}$ is a first-order autoregressive time series (AR(1)), i.e. $\rho(\tau) = \rho(\varphi(\mathbf{X}^{(t)}), \varphi(\mathbf{X}^{(t+\tau)})) = \rho^{|\tau|}$, then we obtain using $1+2\sum_{\tau=1}^{+\infty}\rho^{\tau} = (1+\rho)/(1-\rho)$ that

$$T_{\mathrm{ESS}} = \frac{1-\rho}{1+\rho} \cdot T.$$

Example 6.7 (Gibbs sampling from a bivariate Gaussian (continued)). In examples 4.4) and 4.5 we obtained for the low-correlation setting that $\rho(X_1^{(t-1)}, X_1^{(t)}) = 0.078$, thus the effective sample size is

$$T_{\mathrm{ESS}} = \frac{1-0.078}{1+0.078} \cdot 10000 = 8547.$$

For the high-correlation setting we obtained $\rho(X_1^{(t-1)}, X_1^{(t)}) = 0.979$, thus the effective sample size is considerably smaller:

$$T_{\mathrm{ESS}} = \frac{1-0.979}{1+0.979} \cdot 10000 = 105.$$

◁

7. Data Augmentation and Related Techniques

Thus far, we have considered either sampling from a pre-specified distribution of interest in order to approximate properties of that distribution or sampling from a related distribution in such a way that we can approximate expectations with respect to the distribution of interest by making use of the obtained sample. In the context of MCMC we have done this by constructing ergodic Markov chains for which the distribution of interest, f, is invariant.

In this chapter we consider a very general technique which allows us to readily deal with some problems which would otherwise be very difficult. The idea of *extending the space*: defining a distribution over a larger space that admits the distribution of interest as a particular marginal.

7.1 Data Augmentation and The Gibbs Sampler

Gibbs sampling is only feasible when we can sample easily from the full conditionals. However, this does not need to be the case. A technique that can help achieving full conditionals that are easy to sample from is *demarginalisation*: we introduce a set of auxiliary random variables $Z_1, \dots, Z_r$ such that f is the marginal density of $(X_1, \dots, X_p, Z_1, \dots, Z_r)$, i.e.

$$f(x_1, \dots, x_p) = \int f(x_1, \dots, x_n, z_1, \dots, z_r)\, d(z_1, \dots, z_r).$$

In many cases there is a "natural choice" of the *completion* $(Z_1, \dots, Z_r)$, as the following example shows.

Example 7.1 (Mixture of Gaussians). Consider data $Y_1, \dots, Y_n$, each of which might stem for one of K populations. The distribution of Y_i in the k-th population is $\mathsf{N}(\mu_k, 1/\tau)$. The probability that an observation is from the k-th population is π_k. If we cannot observe from which population the i-th observation was drawn, then it is from a *mixture distribution*:

$$f(y_i) = \sum_{k=1}^{K} \pi_k \phi_{(\mu_k, 1/\tau)}(y_i). \tag{7.1}$$

In a Bayesian framework a suitable prior distribution for the mean parameters μ_k is the $\mathsf{N}(\mu_0, 1/\tau_0)$ distribution. A suitable prior distribution for $(\pi_1, \dots, \pi_K)$ is the Dirichlet distribution (a multivariate generalisation of the Beta distribution) with parameters $\alpha_1, \dots, \alpha_K > 0$ with density

$$f_{(\alpha_1, \dots, \alpha_K)}(\pi_1, \dots, \pi_K) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

for $\pi \geq 0$ and $\sum_{k=1}^K \pi_k = 1$. For the sake of simplicity we assume that the precision τ is known (otherwise, a Gamma distribution would be a common choice of prior distribution for τ, for which we could compute the full conditional distributions of the resulting extended posterior if we employed the same data augmentation strategy as that used here), as well as the number of populations, K. The case of models in which the number of components is not known a priori can be addressed using the techniques described in Chapter 8.

It is however difficult to sample from the posterior distribution of $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$ given data $Y_1, \dots Y_n$ using a Gibbs sampler. This is due to the mixture nature of (7.1). This suggests introducing auxiliary variables $Z_1, \dots Z_n$ which indicate which population the i-th individual is from, i.e.

$$\mathbb{P}(Z_i = k) = \pi_k \qquad \text{and} \qquad Y_i | Z_i = k \sim N(\mu_k, 1/\tau).$$

It is easy to see that the marginal distribution of Y_i is given by (7.1), i.e. the Z_i are indeed a completion. Now we have that

$$\begin{aligned} & f(y_1, \dots, y_n, z_1, \dots, z_n, \mu_1, \dots, \mu_K, \pi_1, \dots, \pi_K) \\ \propto & \left(\prod_{i=1}^n \pi_{z_i} \exp\left(-\tau (y_i - \mu_{z_i})^2/2\right) \right) \cdot \left(\prod_{k=1}^K \exp\left(-\tau_0 (\mu_k - \mu_0)^2/2\right) \right) \cdot \left(\prod_{k=1}^K \pi_k^{\alpha_k - 1} \right). \end{aligned} \tag{7.2}$$

Thus the full conditional distributions given $Y_1, \dots, Y_n$ are

$$\begin{aligned} \mathbb{P}(Z_i = k | Y_1, \dots, Y_n, \mu_1, \dots, \mu_K, \pi_1, \dots, \pi_K) &= \frac{\pi_k \phi_{(\mu_k, 1/\tau)}(y_i)}{\sum_{\iota=1}^K \pi_\iota \phi_{(\mu_\iota, 1/\tau)}(y_i)} \\ \mu_k | Y_1, \dots, Y_n, Z_1, \dots, Z_n, \pi_1, \dots, \pi_K &\sim \mathsf{N}\left(\frac{\tau \left(\sum_{i:\, Z_i = k} Y_i\right) + \tau_o \mu_0}{|\{:\ Z_i = k\}|\tau + \tau_0}, \frac{1}{|\{i:\ Z_i = k\}|\tau + \tau_0} \right) \\ \pi_1, \dots, \pi_K | Y_1, \dots, Y_n, Z_1, \dots, Z_n, \mu_1, \dots, \mu_K &\sim \text{Dirichlet}\left(\alpha_1 + |\{i:\ Z_i = 1\}|, \dots, \alpha_K + |\{I:\ Z_i = K\}|\right). \end{aligned} \tag{7.3}$$

To derive the full conditional of μ_k we have used that the joint density (7.2) is proportional to

$$\prod_{k=1}^K \exp\left(-\frac{|\{i:\ Z_i = k\}|\tau + \tau_0}{2} \left(\mu_k - \frac{\tau \left(\sum_{i:\, Z_i = k} Y_i\right) + \tau_0 \mu_0}{|\{i:\ Z_i = k\}|\tau + \tau_0} \right)^2 \right),$$

as

$$\begin{aligned} \tau \sum_{Z_i = k} (Y_i - \mu_k)^2 + \tau_0 (\mu_k - \mu_0)^2 &= (|\{i:\ Z_i = k\}|\tau + \tau_0)\mu_K^2 + 2\mu_K \left(\tau \left(\sum_{i:\, Z_i = k} Y_i \right) + \tau_0 \mu_0 \right) + \text{ const} \\ &= (|\{i:\ Z_i = k\}|\tau + \tau_0) \left(\mu_K - \frac{\tau \left(\sum_{i:\, Z_i = k} Y_i\right) + \tau_o \mu_0}{|\{i:\ Z_i = K\}|\tau + \tau_0} \right)^2 + \text{ const.} \end{aligned}$$

Thus we can obtain a sample from the posterior distribution of $\mu_1, \dots, \mu_K$ and $\pi_1, \dots, \pi_K$ given observations $Y_1, \dots, Y_n$ using the following auxiliary variable Gibbs sampler: Starting with initial values $\mu_1^{(0)}, \dots, \mu_K^{(0)}, \pi_1^{(0)}, \dots, \pi_K^{(0)}$ iterate the following steps for $t = 1, 2, \dots$

1. For $i = 1, \dots, n$:
   Draw $Z_i^{(t)}$ from the discrete distribution on $\{1, \dots, K\}$ specified by (7.3).
2. For $k = 1, \dots, K$:
   Draw $\mu_k^{(t)} \sim \mathsf{N}\left(\frac{\tau \left(\sum_{i:\, Z_i^{(t)} = k} Y_i\right) + \tau_o \mu_0}{|\{i:\ Z_i^{(t)} = k\}|\tau + \tau_0}, \frac{1}{|\{i:\ Z_i^{(t)} = k\}|\tau + \tau_0} \right)$.
3. Draw $(\pi_1^{(t)}, \dots, \pi_K^{(t)}) \sim \text{Dirichlet}\left(\alpha_1 + |\{i:\ Z_i^{(t)} = 1\}|, \dots, \alpha_K + |\{i:\ Z_i^{(t)} = K\}|\right)$. ◁

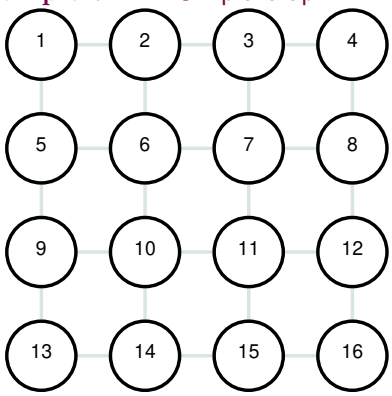
7.2 Data Augmentation and the Ising Model

The *Ising model* is tremendously important for modelling spin-glasses in statistical physics, but also arises naturally in many other contexts including modelling spatial relationships in many statistical contexts: for example, in image denoising. We will see that there is a particularly effective data augmentation strategy which allows us to obtain fast mixing Markov chains which target Ising models in settings in which the simple Gibbs sampler does not behave well.

We need to define a number of quantities before moving on to the Ising model.

Definition 7.1. *Graph A* graph *comprises a collection of vertices $\mathcal{V}$ and a collection of edges which conntect pairs of vertices $\mathcal{E}$.*

Example 7.2. A Simple Graph

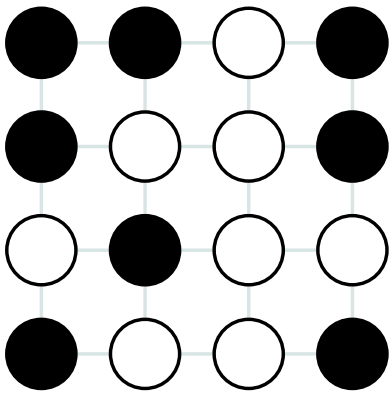


For this graph:

$$\begin{aligned}\mathcal{V} &= \{1, 2, \ldots, 16\} \\ \mathcal{E} &= \{(i,j) \in \mathcal{V}^2 : |i-j| \in \{1, 4\}\} \\ &= \{(1,2), (1,5), \ldots, (15,16)\}\end{aligned}$$

Note: The convention we're following here is that $(i,j) \in \mathcal{E}$ only if $i < j$. Some references adopt the different convention that $(i,j) \in \mathcal{E} \Leftrightarrow (j,i) \in \mathcal{E}$. ◁

Definition 7.2. *Ising Model*



The Ising model on $(\mathcal{V}, \mathcal{E})$ associates a binary variable to each vertex (corresponding to a spin *state in certain physical systems). To each $v_i \in \mathcal{V}$ we attach a binary-valued random variable, X_i, which takes values in $\{-1, +1\}$. A particular configuration of such a model is often indicated by* colouring *the vertices (one colour for $+1$, the other for -1) as in the graph shown here.*
The Ising model specifies a probability distribution over the possible values these binary variables can take.

If the number of vertices, $|\mathcal{V}| =: m$, then:

$$f(x_1, \ldots, x_m) = \frac{1}{Z} \exp\left(J \sum_{(i,j) \in \mathcal{E}} x_i, x_j \right)$$

where J is a parameter which determines interaction type and strength and Z is a normalising constant.

If $J > 0$ then the model favours configurations in which adjacent vertices take the same value; if $J < 0$ then configurations in which adjacent vertices tend to be dissimilar are preferred (the $J = 0$ case is quite uninteresting, corresponding to independent vertices). We will consider only the $J > 0$ case, which is sometimes known as the ferromagnetic Ising model, *in these notes.*

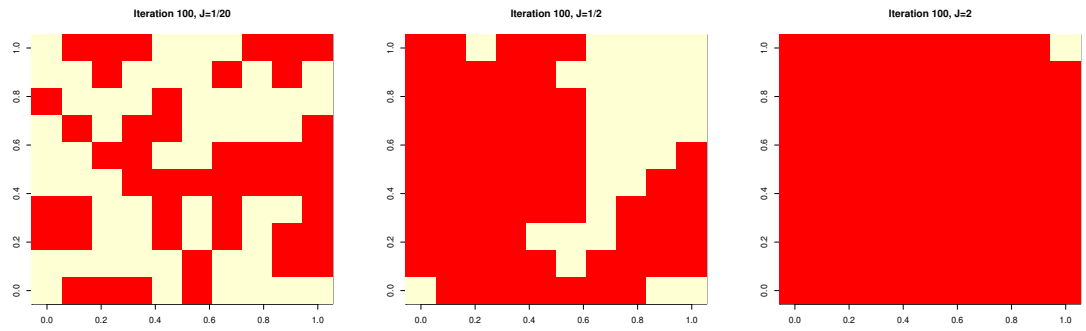
The distribution over $\mathbf{x} = (x_1, \ldots, x_m)$ is clearly a complex one and is of a quite different sort to those we've seen in most of the other examples considered in this module. However, it's still straightforward to

implement a Gibbs sampler for this type of model. The full conditional distributions can be obtained by noting that $f_{X_j|X_{-j}}(x_j|x_{-j}) = f(\mathbf{x})/(f(\mathbf{x}) + f(x_1, \ldots, x_{j-1}, 1 - x_j, x_{j+1}, \ldots, x_m)$:

$$f_{X_j|X_{-j}}(x_j|x_{-j}) = \exp\left(J\sum_{i\sim j} x_i x_j\right) \Big/ \left[\exp\left(-J\sum_{i\sim j} x_i\right) + \exp\left(J\sum_{i\sim j} x_i\right)\right].$$

where $i \sim j$ is a compact notation meaning $\{i : \{(i,j),(j,i)\} \cap \mathcal{E} \neq \emptyset\}$ (i.e. the summation is over all i which are neighbours of j within our graph in that i and j share a common edge).

We can the readily implement a Gibbs sampler (Algorithm 4.1 or 4.2) which targets this Ising model and when the graph is small enough or the coupling strength J is close to zero these algorithms can work well. However, when J is large (and the meaning of larger depends upon the size of the graph) the correlation between the adjacent states within the Ising model can cause the Gibbs sampler to mix slowly, just as in the case of the multivariate normal distribution. Figure 7.1 illustrates this phenomenon: notice that the chain with $J = 0.05$ shown in the leftmost panel has reached a state with little in common with its starting value, while that in the rightmost panel has hardly moved from the starting value in 100 iterations.



(a) J=0.05 (b) J=0.50 (c) J=2.00

Figure 7.1. The 100th sample from a Gibbs sampler chain initialised with all states equal to 1 (shown as red squares in these figures) targetting an Ising model with the natural neighbourhood structure on a 10×10 lattice for three different values of J.

Data Augmentation for the Ising Model One way in which we could try to improve the mixing of our Gibbs sampler would be to perform some sort of blocking. It's difficult, however to design a good block structure for a model of this sort. One strategy which can be tremendously successful is to try to introduce auxiliary variables which introduce a natural random block structure that depends upon the current configuration and which can then be exploited to propose large moves — and that is what we will do in this section.

First, note that we can easily rewrite the distribution as:

$$\begin{aligned} f(x_1, \ldots, x_m) =& \frac{1}{Z} \exp\left(J \sum_{(i,j)\in\mathcal{E}} x_i, x_j\right) \\ =& \frac{1}{Z} \exp(-J|\mathcal{E}|) \exp\left(2J \sum_{(i,j)\in\mathcal{E}} \mathbb{I}(x_i = x_j)\right) \\ =& \frac{1}{Z'} \exp\left(2J \sum_{(i,j)\in\mathcal{E}} \mathbb{I}(x_i = x_j)\right) \end{aligned}$$

where $\mathbb{I}(x_i = x_j)$ takes the value one when $x_i = x_j$ and zero otherwise and so identifies pairs of like vertices and Z' is a different normalising constant which incorporates the $\exp(-J|\mathcal{E}|)$ factor. This rewriting exploits the fact that as all pairs of vertices are either like (both having the same value associated with them) or unlike (having different values to one another) we need count only the number of like neighbours of any given vertex to completely characterise the configuration.

Our approach to augmentation is going to be to add "bonds" between some like vertices. To be more precise, we add an auxiliary variable for every edge in the graph. For any $(i, j) \in \mathcal{E}\}$ we add an auxiliary variable $u_{i,j}$. We denote by $\mathbf{u}$ the full collection of these auxiliary distributions and specify the distribution over the $\mathbf{u}$ variables via their full conditional distributions:

$$u_{i,j}|\mathbf{u} \setminus u_{i,j}, x \sim \mathsf{Bernoulli}(\rho\mathbb{I}(x_i = x_j))$$

where $\rho = 1 - \exp(-2J)$. That is, these variables are certainly zero if they connect two vertices which take different values (grey bonds in Figure 7.2, but are one with probability ρ if they connect vertices which take like values (red bonds), otherwise taking the value 0 (blue bonds). This can be thought of as a bonding interaction between like vertices

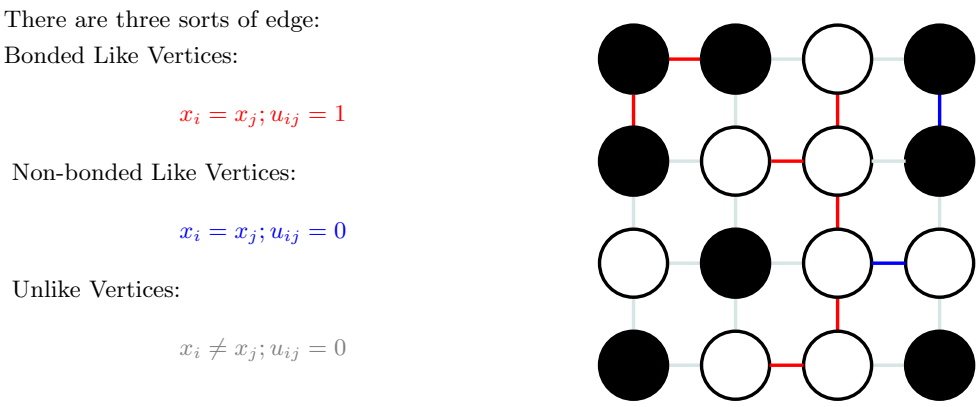


Figure 7.2. An illustration of "bonding" within the augmented Ising model: unlike vertices are always connected by edges which take value 0; like vertices with probability ρ are connected by edges which take the value 1 (bonds) but are otherwise also connected by edges which take value zero. Note this is just one *possible* bonding configuration associated with these vertex values.

Noting that we know the marginal distribution of $\mathbf{x}$ and we know the full conditional distribution of each $u_{i,j}$, and that the $u_{i,j}$ are independent of one another conditional upon $\mathbf{x}$, we can write the joint distribution as the product of the marginal distribution of $\mathbf{x}$ and the conditional distribution $\mathbf{u}$ given $\mathbf{x}$:

$$\begin{aligned} f(\mathbf{x}, \mathbf{u}) &\propto \exp\left(2J \sum_{(i,j)\in\mathcal{E}} \mathbb{I}(x_i = x_j)\right) \prod_{(i,j)\in\mathcal{E}:x_i=x_j} (\mathbb{I}(u_{i,j} = 1)\rho + \mathbb{I}(u_{i,j} = 0)(1-\rho)) \prod_{(i,j)\in\mathcal{E}:x_i\neq x_j} u_{i,j} = 0 \\ &\propto \exp\left(2J \sum_{(i,j)\in\mathcal{E}} \mathbb{I}(x_i = x_j)\right) \prod_{(i,j)\in\mathcal{E}:x_i=x_j} (\mathbb{I}(u_{i,j} = 1)\rho + \mathbb{I}(u_{i,j} = 0)(1-\rho)) \prod_{(i,j)\in\mathcal{E}:u_{ij}=1} \mathbb{I}(x_i \neq x_j) \end{aligned}$$

where the final term in each line is present just to ensure that there are no bonds between unlike vertices (i.e. whenever $u_{i,j} = 1$ we must have $x_i = x_j$). This is the *contrapositive* of the statement that whenever $x_i \neq x_j$ we must have that $u_{i,j} = 0$ and the final term of the second line enforces exactly the same condition as that in the first, but this form will prove easier to manipulate later.

Although this expression may not immediately appear very useful it will allow us to establish that this distribution has a quite remarkable property: it provides a complete decoupling of $\mathbf{x}|\mathbf{u}$ and also of $\mathbf{u}|\mathbf{x}$ in

the sense that given $\mathbf{u}$ the elements of $\mathbf{x}$ are independent of one another *and* given $\mathbf{x}$ the elements of $\mathbf{u}$ are independent of one another.

In order to establish this fact, it is convenient to compose the contributions the (log) probability density arising from the types of edge shown in Figure 7.2.

$$E_\Delta := \{(i,j) : x_i \neq x_j\}, \qquad E_0 := \{(i,j) : x_i = x_j, u_{ij} = 0\}, \qquad E_1 := \{(i,j) : x_i = x_j, u_{ij} = 1\}.$$

where E_Δ are the edges between unlike vertices, E_0 are edges between like vertices with an associated $u_{i,j} = 0$ (i.e. unbonded edges) and E_1 are the edges between like vertices for which $u_{i,j} = 1$ (i.e. bonded edges).

We can rewrite the distribution in terms of these sets of edges in the following way:

$$\begin{aligned} f(\mathbf{x},\mathbf{u}) &\propto \underbrace{\exp\left(2J \sum_{(i,j)\in\mathcal{E}} \mathbb{I}(x_i = x_j)\right)}_{(1-\rho)^{-(|E_0|+|E_1|)}} \underbrace{\prod_{(i,j)\in\mathcal{E}:x_i=x_j} (\mathbb{I}(u_{i,j}=1)\rho + \mathbb{I}(u_{i,j}=0)(1-\rho))}_{\rho^{|E_1|}(1-\rho)^{|E_0|}} \prod_{(i,j)\in\mathcal{E}:u_{ij}=1} \mathbb{I}(x_i = x_j) \\ &\propto (1-\rho)^{-(|E_0|+|E_1|)}\rho^{|E_1|}(1-\rho)^{|E_0|} \prod_{(i,j)\in\mathcal{E}:u_{ij}=1} \mathbb{I}(x_i = x_j) \\ &\propto (1-\rho)^{|E_\Delta|}\rho^{|E_1|}(1-\rho)^{|E_0|} \prod_{(i,j)\in\mathcal{E}:u_{ij}=1} \mathbb{I}(x_i = x_j) \\ &\propto (1-\rho)^{|E_\Delta|+|E_0|}\rho^{|E_1|} \prod_{(i,j)\in\mathcal{E}:u_{ij}=1} \mathbb{I}(x_i = x_j) \end{aligned}$$

the first equality here follows from the definition of ρ and the various subsets of vertices and the second from noting that $(1-\rho)^{|\mathcal{E}|}$ is a constant and

$$(1-\rho)^{|\mathcal{E}|}(1-\rho)^{-(|E_0|+|E_1|)} = (1-\rho)^{|E_\Delta|}.$$

This tells us that the resulting distribution vanishes anywhere that the vertices at the end of a bonding edge (one for which $u_{i,j} = 1$ are different, and otherwise has a density which depends only upon the number of edges for which $u_{i,j} = 1$.

Before considering the mathematical consequences of this representation, it's instructive to consider the following algorithm which exploits the representation in order to make large-scale moves.

Algorithm 7.1 (The Swendsen-Wang Algorithm). Consider the extended target distribution:

$$f(\mathbf{x},\mathbf{u}) \propto (1-\rho)^{|E_\Delta|+|E_0|}\rho^{E_1} \prod_{(i,j)\in\mathcal{E}:u_{ij}=1} \mathbb{I}(x_i = x_j).$$

The Swendsen-Wang Algorithm is simply a block Gibbs sampler for this extended target distribution which proceeds iteratively with two steps:

1. Sample $\mathbf{U}^t|\mathbf{X}^{t-1}$
2. Sample $\mathbf{X}^t|\mathbf{U}^t$

The implementation is remarkably simple:

Cluster: For $(i,j) \in \mathcal{E}$: Sample

$$U_{i,j}^t \sim \mathsf{Bernoulli}(\rho\mathbb{I}(x_i^{t-1} = x_j^{t-1})).$$

Flip: Two steps:

1. Identify connected components.

2. Set every $X_\cdot^t$ within each U^t-connected component to the same value sampled uniformly from $\{-1, +1\}$.

Figure 7.3 illustrates this algorithm. It's intuitively simple: given a particular set of vertex values, each bond is sampled independently from a distribution which places probability 1 on the bond variable taking the value 0 if those vertices take different values, but otherwise places probablity ρ on the bond variable taking the value one. This produces a complete set of bond variables and hence *clusters* of connected like vertices. Given a particular clustering configuration, the algorithm then samples the values of the (like) vertices within each cluster uniformly from the set of possible values, $\{-1, +1\}$ — and that's it.

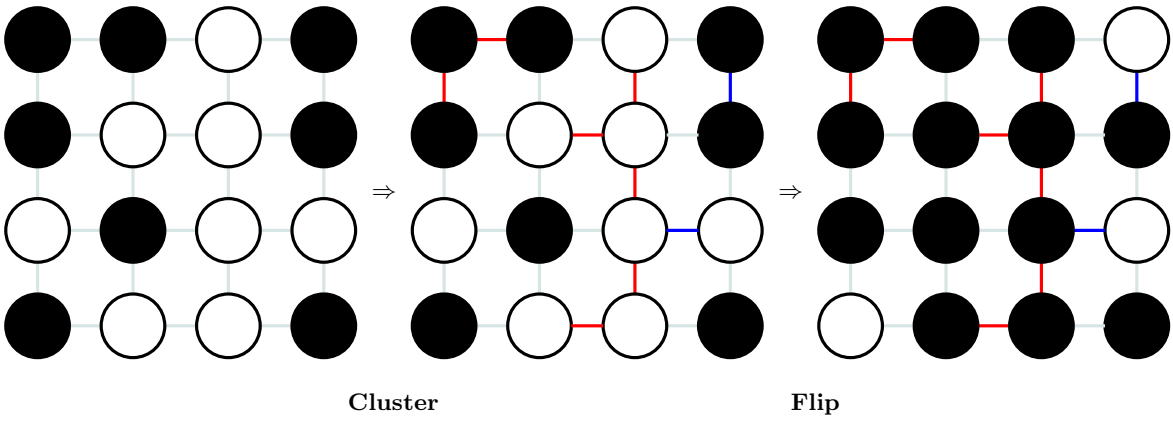


Figure 7.3. The Swendsen-Wang "Cluster-Flip" Dynamics

Having seen the Algorithm, we can return to establishing that it produces a Markov chain with the correct invariant distribution.

By construction:

$$\mathbf{U}|\mathbf{X} \sim \prod_{(i,j)\in\mathcal{E}} \mathsf{Bernoulli}(u_{i,j}; \rho\mathbb{I}(x_i = x_i))$$

Whilst:

$$f(\mathbf{x}|\mathbf{u}) \propto f(\mathbf{x}, \mathbf{u}) = (1-\rho)^{|E_\Delta|+|E_0|}\rho^{E_1} \prod_{(i,j)\in\mathcal{E}:u_{ij}=1} \mathbb{I}(x_i = x_j)$$

where $|E_\Delta| + |E_0| = |\mathcal{E}| - \sum_{(i,j)\in\mathcal{E}} u_{i,j}$ and $|E_1| = \sum_{(i,j)\in\mathcal{E}} u_{i,j}$ so:

$$f(\mathbf{x}|\mathbf{u}) \propto \prod_{(i,j)\in\mathcal{E}:u_{ij}=1} \mathbb{I}(x_i = x_j)$$

is uniform over u-compatible configurations.

Consequently, Algorithm 7.2 really is just a block Gibbs sampler for an extended distribution which admits the Ising model as the marginal distribution over $\mathbf{x}$.

7.3 Universal Augmentation and Rejection Sampling

In Section 7.1 we saw an approach to data augmentation for Gibbs sampling within mixture models, which is a common and important problem. In Section 7.2 we saw an elegant but specialised approach to data augmentation for the particular case of Ising models. One obvious question is whether there are

any *general* recipes for constructing augmented distributions for *any* target distribution and, with some caveats, there *is*.

Given any probability distribution $f(\mathbf{x})$, we can define the extended distribution

$$\bar{f}(\mathbf{x}, u) := f(\mathbf{x}) \cdot \frac{1}{f(\mathbf{x})} \mathbb{I}_{[0, f(\mathbf{x})](u)}.$$

If we do this we find that $\bar{f}(\mathbf{x}, u) = \mathbb{I}_{[0, f(\mathbf{x})]}(u)$.

This is just another way of stating the result used to justify rejection sampling: that simulation of a point uniformly on the set of points beneath the density $f(\mathbf{x})$ and discarding the vertical coordinate is equivalent to sampling from the distribution associated with that density.

Armed with this extended-space representation, we can first revisit rejection sampling and treat it as an importance sampling algorithm:

Proposition 7.1 (Rejection Sampling is (self-normalised) Importance Sampling on an Extended Space). *Given a distribution of interest $f(x)$, define $\bar{f}(\mathbf{x}, u) = \mathbb{I}_{[0, f(\mathbf{x})]}(u)$. Given a proposal distribution $g(\mathbf{x})$ and a constant $M \geq \sup_{\mathbf{x}} f(\mathbf{x})/g(\mathbf{x})$, define $\bar{g}(\mathbf{x}, u) = \frac{1}{M} \mathbb{I}_{[0, M \cdot g(\mathbf{x})]}$.*

Given an (integrable) function $\varphi(\mathbf{x}) : E \to \mathbb{R}$ the associated self-normalised importance sampling estimator of $\mathbb{E}_{\bar{f}}[\varphi(\mathbf{X})] \equiv \mathbb{E}_f[\varphi(\mathbf{X})]$ is the standard rejection sampling estimator obtained using g as a proposal distribution and M as a bounding constant.

Proof. Sampling from $\bar{g}$ and importance weighting to target $\bar{f}$ gives importance weights of the form:

$$w(\mathbf{x}, u) = \frac{\bar{f}(\mathbf{x}, u)}{\bar{g}(\mathbf{x}, u)} = \frac{\mathbb{I}_{[0, f(\mathbf{x})]}(u)}{\frac{1}{M} \mathbb{I}_{[0, M \cdot g(\mathbf{x})]}(u)}$$

by construction $\mathbb{I}_{[0, f(x)]}(u) \neq 0 \Rightarrow \mathbb{I}_{[0, M \cdot g(x)]}(u) = 1$ and so:

$$w(\mathbf{x}, u) = M \cdot \mathbb{I}_{[0, f(\mathbf{x})]}(u).$$

The self-normalised importance sampling estimator makes use of the same weights, $w(\mathbf{x}, u)$, and divides by the sum of these weights, leading to an estimator:

$$\frac{\sum_{i=1}^{n} w(\mathbf{X}^{(i)}, U^{(i)}) \varphi(\mathbf{X}^{(i)})}{\sum_{j=1}^{n} w(\mathbf{X}^{(j)}, U^{(j)})} = \frac{\sum_{i=1}^{n} \mathbb{I}_{[0, f(\mathbf{X}^{(i)})]}(U^{(i)}) \varphi(\mathbf{X}^{(i)})}{\sum_{j=1}^{n} \mathbb{I}_{[0, f(\mathbf{X}^{(i)})]}(U^{(i)})} = \frac{\sum_{\{i: U^{(i)} < f(\mathbf{X}^{(i)})\}} \varphi(\mathbf{X}^{(i)})}{|\{i : U^{(i)} < f(\mathbf{X}^{(i)})|}$$

where $U^{(i)} | \mathbf{X}^{(i)} \sim \mathsf{U}[0, M \cdot g(\mathbf{X}^{(i)})$ and so this is exactly the average of those samples which would be *accepted* in the rejection sampling setting (noting that a $\mathsf{U}[0, M \cdot g(\mathbf{X}^{(i)})]$ random variable can be obtained by simulating a $\mathsf{U}[0, 1]$ random variable and multiplying by $M \cdot g(\mathbf{X}^{(i)})$). □

Looking again at Figure 2.3, we can interpret rejection sampling as simulating uniformly from the area beneath the bounding curve and attaching importance weights of M to points in the dark gray area and of 0 to those in the light gray area (before discarding the vertical coordiante to obtain samples from the normal distribution of interest, f, in the notation of Proposition 7.1.

Again, we don't need to know the normalising constant of f (or g) in order to make use of this type of algorithm, provided we can bound the ratio of unnormalised versions of f and g. For example, using $f(x) = C \cdot f(x)$ with unknown normalising constant in place of f doesn't change anything of any consequence with this data augmentation strategy, if we replace $\bar{f}(\mathbf{x}, u)$ with $\bar{\pi}(\mathbf{x}, u)$ and choose M such that $M \geq \sup_x \pi(x)/g(x)$ then exactly the same self-normalised importance sampling estimator results (and, indeed, any algorithm which samples uniformly on the area beneath the graph of $f(x)$ will, if the vertical coordinate is discarded, produce exactly the same output as a sample obtained uniformly under the graph of $f(x)$ by multiplying the vertical coordinates by M).

7.4 Universal Augmentation and the Slice Sampler

A natural question, having obtained this reinterpretation of rejection sampling is whether we can apply other sampling methods to the same extended distribution.

The *slice sampler* is an algorithm which allows us to approximate expectations with respect to $f(\mathbf{x})$ by *Gibbs sampling* from $\bar{f}(\mathbf{x}, u) \propto \mathbb{I}_{[0,f(x)]}u$.

Algorithm 7.2 (Slice Sampler). Starting with $(\mathbf{X}^{(0)}, U^{(0)})$ iterate for $t = 1, 2, \ldots$

1. Draw $\mathbf{X}^{(t)} \sim \bar{f}_{\mathbf{X}|U}(\cdot|U^{(t-1)})$.
2. Draw $U^{(t)} \sim \bar{f}_{U|\mathbf{X}}(\cdot|\mathbf{X}^{(t)})$.

In order to implement this algorithm we need to identify the two conditional distributions from which it samples, the first of these is the more difficult to deal with:

$$\begin{aligned} \bar{f}_{\mathbf{X}|U}(\mathbf{x}|u) &\propto \bar{f}(\mathbf{x}, u) \propto \mathbb{I}_{[0,f(\mathbf{x})]}u \\ \Rightarrow \bar{f}_{\mathbf{X}|U}(\mathbf{x}|u) &= \frac{\mathbb{I}_{[0,f(\mathbf{x})]}(u)}{\int \mathbb{I}_{[0,f(\mathbf{x}')]}(u)d\mathbf{x}'} \end{aligned}$$

i.e. $\bar{f}_{\mathbf{X}|U}(\mathbf{x}|u)$ is uniform over *level sets* of $f(\mathbf{x})$. That is, letting $L(u) := \{x : f(x) \geq u\}$, $\bar{f}_{\mathbf{X}|U}(\mathbf{x}|u) \propto \mathbb{I}_{L(u)}(x)$. Whils the second full conditional distribution is straightforward to simulate from by construction:

$$\bar{f}_{U|\mathbf{X}}(u|\mathbf{x}) = \frac{1}{f(\mathbf{x})}\mathbb{I}_{[0,f(\mathbf{x})]}(u).$$

The conditional distributions are illustrated in Figure 7.4.

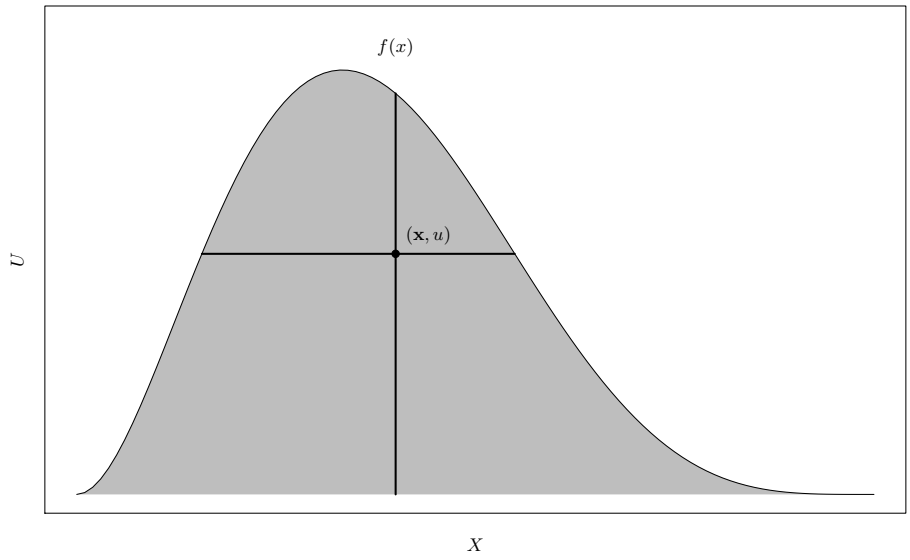


Figure 7.4. Slice sampling: starting from the point marked $(\mathbf{x}, u)$, the horizontal line illustrates the range of possible values a slice-sampler move in the x-direction could reach, the horizontal line those values that could be reached by a move in the u-direction.

An example of the Algorithm 7.2 applied to a beta distribution is shown in Figure 7.5.

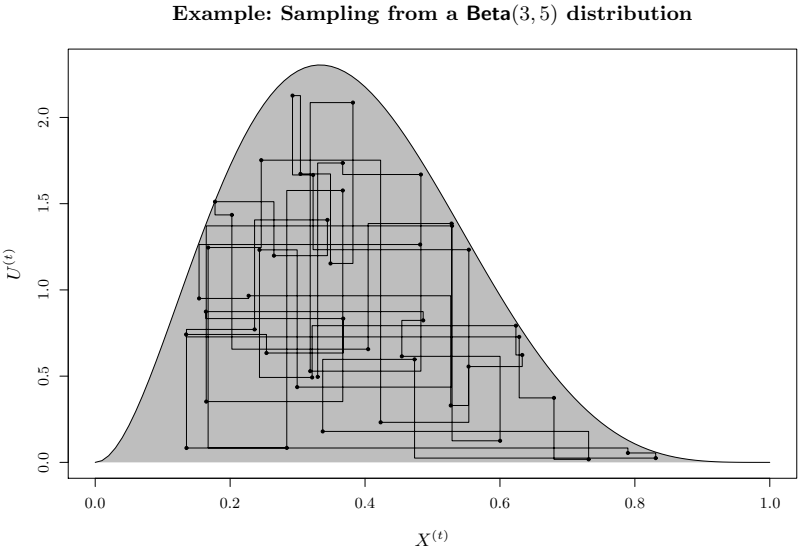


Figure 7.5. The trajectory of a slice sampler approximating a $\mathsf{Beta}(3,5)$ distribution.

As slice sampling only involves sampling from uniform distributions it seems as though its implementation should generally be straightforward, and for simple problems it often is. However, for complicated problems, especially multi-modal ones the apparent simplicity of the sampling steps masks a significant complication: the identification of $L(u)$. Sampling uniformly from the (often disconnected) region in which the density is above a particular threshold value is very hard: even working of which parts of the space make up this region can be extremely difficult.

There are two related algorithms which progressively relax this requirement, at the expense of producing less rapidly mixing Markov chains. The first replaces this Gibbs sampling step with a sequence of more modest Gibbs sampling steps, reducing the problem of identifying and sampling from the level sets of a multivariate function to that of identifying and sampling from the level sets of several univariate functions. This gives rise to Algorithm 7.3.

Algorithm 7.3 (Coordinate-wise Slice Sampler). Starting with $(X_1^{(0)}, \ldots, X_p^{(0)}, U^{(0)})$ iterate for $t = 1, 2, \ldots$

1. Draw $X_1^{(t)} \sim \bar{f}_{X_1|X_{-1},U}(\cdot|X_{-1}^{(t-1)}, U^{(t-1)})$. $\vdots$
p. Draw $X_p^{(t)} \sim \bar{f}_{X_p|X_{-p},U}(\cdot|X_{-p}^{(t)}, U^{(t-1)})$.
p+1. Draw $U^{(t)} \sim \bar{f}_{U|\mathbf{X}}(\cdot|\mathbf{X}^{(t)})$.

This is a simple Gibbs sampler for $\bar{f}$ whereas the slice sampler presented as Algorithm 7.2 can be interpreted as a blocked version of the same, with all components of $\mathbf{X}$ sampled simultaneously. As such, the full slice sampler will always mix more quickly than this coordinate-at-a-time algorithm, but it is more difficult to implement. Note that:

$$\bar{f}_{X_j|X_{-j},U}(x_j|x_{-j},u) \propto \bar{f}(\mathbf{x},u) \qquad \Rightarrow \bar{f}_{X_j|X_{-j},U}(x_j|x_{-j},u) \propto \mathbb{I}_{[u\leq f(\mathbf{x})]}$$

which means it must be the uniform distribution on the set $L_j(u|x_{-j}) := \{x_j : f(x_1, \ldots, x_j, \ldots, x_p) \geq u\}$.

Although identifying these univariate level sets may be easier, it is still too difficult to be practical for many interesting problems. As always, when a Gibbs sampler is too difficult to implement, a natural strategy is to replace the Gibbs sampling step with an appropriate Metropolis-Hastings step and this gives rise to Algorithm 7.4.

Algorithm 7.4 (Metropolised Slice Sampler). Starting with $(\mathbf{X}^{(0)}, U^{(0)})$ iterate for $t = 1, 2, \ldots$

1. Draw $\mathbf{X} \sim \bar{q}_(\cdot | \mathbf{X}^{(t-1)}, U^{(t-1)})$.
2. With probability

$$\alpha(\mathbf{X}|\mathbf{X}^{(t-1)}; U^{(t-1)}) = \min\left(1, \frac{\bar{f}(\mathbf{X}, U^{(t-1)}) q(\mathbf{X}^{(t-1)}|\mathbf{X}, U^{(t-1)})}{\bar{f}(\mathbf{X}^{(t-1)}, U^{(t-1)}) q(\mathbf{X}|\mathbf{X}^{(t-1)}, U^{(t-1)})}\right)$$

accept and set $\mathbf{X}^{(t)} = \mathbf{X}$.

otherwise, set $\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)}$.

2. Draw $U^{(t)} \sim \bar{f}_{U|\mathbf{X}}(\cdot|\mathbf{X}^{(t)})$.

Note that this is just a standard Metropolis-Hastings step in which the U-component of the state vector is unchanged, and this acceptance probability follows directly from the standard expression. This Metropolised slice sampler is much more generally applicable than the purer form described above, but it's mixing properties depend critically on the proposal distribution (and are typically much poorer than those of the slice sampler itself).

8. The Reversible Jump Algorithm

There are many problems in which the *dimension* of the parameter vector is unknown and performing inference over this dimension as well as the value of the parameter is of interest. In this chapter, we explore one common setting in which such problems arise, Bayesian model comparison, before moving on to explore a reinterpretation of the Metropolis-Hastings algorithm which provides one approach to addressing problems of this kind.

8.1 Bayesian multi-model inference

Examples 4.1, 7.1 and many others earlier this module illustrated how MCMC techniques can be used in Bayesian modeling. In both examples we have only considered a single model. In many real word situations however there is (a priori) more than one plausible model.

Assume that we consider a finite or countable set of models $\{\mathcal{M}_1, \mathcal{M}_2, \ldots\}$. Each model is characterised by a density f_k and the associated parameter space Θ_k, i.e. $\mathcal{M}_k := \{f_k(\cdot|\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta_k\}$, where f_k is the density and Θ_k the parameter space of the k-th model.

Using a hierarchical Bayesian setup (i.e. that is one in which there are several layers of unknown quantities, in this case with the highest level unknown quantity being which model describes the process by which data is generated) we first place a prior distribution of the set of models, i.e.

$$\mathbb{P}(\mathcal{M}_k) = p_k$$

with $\sum_k p_k = 1$. The prior distribution on the model space can for example be used to express our prior belief in simple models. Further we need to place a prior on each parameter space Θ_k, i.e.

$$\boldsymbol{\theta}|\mathcal{M}_k \sim f_k^{\text{prior}}(\boldsymbol{\theta}).$$

Assume now that we have observed data $\mathbf{y}_1, \ldots, \mathbf{y}_n$. When considering model $\mathcal{M}_k$ the likelihood, conditional upon this being the model, is

$$l_k(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta}) := \prod_{i=1}^{n} f_k(\mathbf{y}_i|\boldsymbol{\theta}),$$

and the posterior density of $\boldsymbol{\theta}$ is

$$f_k^{\text{post}}(\boldsymbol{\theta}) = \frac{f_k^{\text{prior}}(\boldsymbol{\theta}) l_k(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta})}{\int_{\Theta_k} f_k^{\text{prior}}(\boldsymbol{\vartheta}) l_k(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\vartheta}) d\boldsymbol{\vartheta}}.$$

Now we can use Bayes formula to compute the posterior probability that the data was generated by model $\mathcal{M}_k$ by writing the posterior distribution over the model *and* parameters jointly and then marginalising out those parameters:

$$\begin{aligned}\mathbb{P}(\mathcal{M}_k|\mathbf{y}_1,\ldots,\mathbf{y}_n) &= \int_{\Theta_k} \frac{p_k f_k^{\text{prior}}(\boldsymbol{\theta}) l_k(\mathbf{y}_1,\ldots,\mathbf{y}_n|\boldsymbol{\theta})}{\sum_\kappa p_\kappa \int_{\Theta_\kappa} f_\kappa^{\text{prior}}(\boldsymbol{\vartheta}) l_\kappa(\mathbf{y}_1,\ldots,\mathbf{y}_n|\boldsymbol{\vartheta}) d\boldsymbol{\vartheta}} d\boldsymbol{\theta} \\ &= \frac{p_k \int_{\Theta_k} f_k^{\text{prior}}(\boldsymbol{\theta}) l_k(\mathbf{y}_1,\ldots,\mathbf{y}_n|\boldsymbol{\theta})\, d\boldsymbol{\theta}}{\sum_\kappa p_\kappa \int_{\Theta_\kappa} f_\kappa^{\text{prior}}(\boldsymbol{\vartheta}) l_\kappa(\mathbf{y}_1,\ldots,\mathbf{y}_n|\boldsymbol{\vartheta})\, d\boldsymbol{\vartheta}}\end{aligned}$$

The comparison between two models $\mathcal{M}_{k_1}$ and $\mathcal{M}_{k_2}$ can be summarised by the posterior odds

$$\frac{\mathbb{P}(\mathcal{M}_{k_1}|\mathbf{y}_1,\ldots,\mathbf{y}_n)}{\mathbb{P}(\mathcal{M}_{k_2}|\mathbf{y}_1,\ldots,\mathbf{y}_n)} = \frac{p_{k_1}}{p_{k_2}} \cdot \underbrace{\frac{\int_{\Theta_{k_1}} f_{k_1}^{\text{prior}}(\boldsymbol{\theta}) l_{k_1}(\mathbf{y}_1,\ldots,\mathbf{y}_n|\boldsymbol{\theta})\, d\boldsymbol{\theta}}{\int_{\Theta_{k_2}} f_{k_2}^{\text{prior}}(\boldsymbol{\theta}) l_{k_2}(\mathbf{y}_1,\ldots,\mathbf{y}_n|\boldsymbol{\theta})\, d\boldsymbol{\theta}}}_{\text{“Bayes factor”}}.$$

Having computed the posterior distribution over the models we can now either consider the model with the highest posterior probability $\mathbb{P}(\mathcal{M}_k|\mathbf{y}_1,\ldots,\mathbf{y}_n)$ (a simple form of *model selection*) or perform *model averaging* using $\mathbb{P}(\mathcal{M}_k|\mathbf{y}_1,\ldots,\mathbf{y}_n)$ as weights and computing estimates as the weighted average of the estimate provided under each model individually.

In order to compute the above probabilities we could run a separate MCMC algorithm for each model (“within model simulation”). Alternatively we could construct a single algorithm that can jump between the different models (“transdimensional simulation”). In order to do this we have to sample from the joint posterior

$$f^{\text{post}}(k,\boldsymbol{\theta}) = \frac{p_k f_k^{\text{prior}}(\boldsymbol{\theta}) l_k(\mathbf{y}_1,\ldots,\mathbf{y}_n|\boldsymbol{\theta})}{\sum_\kappa p_\kappa \int_{\Theta_\kappa} f_\kappa^{\text{prior}}(\boldsymbol{\vartheta}) l_\kappa(\mathbf{y}_1,\ldots,\mathbf{y}_n|\boldsymbol{\vartheta})\, d\boldsymbol{\vartheta}}$$

defined on the *disjoint union* space

$$\Theta := \bigcup_k (\{k\} \times \Theta_k).$$

Such spaces take a little getting used to, it's useful to think of the first coordinate as specifying a model and the remaining coordinates, *the number and nature of which* typically depend upon that first coordinate, are then the parameters of that model.

Unfortunately, we cannot use the approaches we have already seen (Metropolis-Hastings and Gibbs sampling) to sample from Θ, as Θ is not as well-behaved as the Θ_k: Θ is a union of spaces of different dimensions, to which — due to measure-theoretic subtleties — the theory which justifies these methods fails to apply. These subtleties can be understood by noting that the comparison of densities which are defined on spaces of different dimensions isn't a natural operation.

Bayesian multi-model inference is an example of a *variable dimension model*. A variable dimension model is a model “where one of the things you do not know is the number of things you do not know” (Green, 2003). In the following two sections we will try to extend the Metropolis-Hastings method to this more general setting.

8.2 Another look at the Metropolis-Hastings algorithm

Recall the random walk Metropolis-Hastings algorithm (algorithm 5.2) in which we set $\mathbf{X} := \mathbf{X}^{(t-1)} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim g$. In this section we will generalise this, by allowing the proposed value to be obtained by a general transformation of the previous value and some auxiliary random variable:

$$\mathbf{X} = \boldsymbol{\tau}(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)}), \text{ with } \mathbf{U}^{(t-1)} \sim g_{1\to 2}.$$

For our further developments it will be necessary that the transformation is a bijective map (i.e. a one-to-one mapping), which requires that the image and the domain of the transformation have the same dimension. Thus we consider

$$\mathbf{T}_{1\to 2} : (\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)}) \mapsto (\mathbf{X}, \mathbf{U}),$$

such that $\mathbf{X} = \boldsymbol{\tau}(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)})$. Furthermore we shall assume that $\mathbf{T}_{1\to 2}$ is a *diffeomorphism*[1] (i.e. $\mathbf{T}_{1\to 2}$ is has as inverse $\mathbf{T}_{2\to 1}$, and both $\mathbf{T}_{1\to 2}$ and $\mathbf{T}_{1\to 2}^{-1}$ are differentiable) with inverse $\mathbf{T}_{2\to 1} = \mathbf{T}_{1\to 2}^{-1}$.

If we generate a newly proposed value $\mathbf{X}$ as mentioned above, how do we have to choose the probability of acceptance $\alpha(\mathbf{X}|\mathbf{X}^{(t-1)})$ such that the resulting MCMC algorithm fulfills the detailed balance condition?

If we set the probability of acceptance (in this module we use the convention $|\mathbf{A}| = |\det(\mathbf{A})|$) to

$$\alpha(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = \min\left\{1, \frac{f(\mathbf{x}^{(t)})g_{2\to 1}(\mathbf{u}^{(t)})}{f(\mathbf{x}^{(t-1)})g_{1\to 2}(\mathbf{u}^{(t-1)})}\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})}{\partial(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})}\right|\right\},$$

then we can establish that detailed balance holds, provided we choose uniformly at random between the two available moves, as we will see below. Note that we use the commonon notation

$$\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})}{\partial(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})}\right|$$

to refer to the absolute value of the Jacobian determinant associated with $\mathbf{T}_{1\to 2}$, i.e. the absolute value of the determinant of a matrix of the partial derivatives of $\mathbf{T}_{1\to 2}$ with respect to its arguments (such that element ij is the partial derivative of coordinate i of $\mathbf{T}_{1\to 2}$ with respect to coordinate j of its argument) evaluated at $(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})$.

Example 8.1 (Random walk Metropolis). In order to clarify what we have just derived we will state the random walk Metropolis algorithm in terms of this new approach.
In the random walk Metropolis algorithm with a symmetric proposal $g_{1\to 2}$ we considered

$$\mathbf{X} = \mathbf{X}^{(t-1)} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim g_{1\to 2},$$

which corresponds to using

$$(\mathbf{X}, \mathbf{U}) = \mathbf{T}_{1\to 2}(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)}) = (\mathbf{X}^{(t-1)} + \mathbf{U}^{(t-1)}, \mathbf{U}^{(t-1)}), \qquad \mathbf{U}^{(t-1)} \sim g_{1\to 2}.$$

For the backward move we generate $\mathbf{U} \sim g_{1\to 2}$ as well, i.e. we have $g_{1\to 2} = g_{2\to 1}$. Further $\mathbf{T}_{2\to 1}(\mathbf{X}^{(t)}, \mathbf{U}^{(t)}) = (\mathbf{X}^{(t)} - \mathbf{U}^{(t)}, \mathbf{U}^{(t)})$. Note that due to the symmetry of $g_{1\to 2}$ this is equivalent (in the sense of equality of distribution) to setting $\mathbf{X}^{(t)} + \mathbf{U}^{(t)}$, and the forward move (based on $T_{1\to 2}$) and backward move (based on $T_{2\to 1}$) are identical.
We accept the newly proposed $\mathbf{X}$ with probability

$$\alpha(\mathbf{X}|\mathbf{X}^{(t-1)}) = \min\left\{1, \frac{f(\mathbf{X})g_{2\to 1}(\mathbf{U})}{f(\mathbf{X}^{(t-1)})g_{1\to 2}(\mathbf{U}^{(t-1)})}\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)})}{\partial(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)})}\right|\right\} = \min\left\{1, \frac{f(\mathbf{X})}{f(\mathbf{X}^{(t-1)})}\right\},$$

as $g_{1\to 2} = g_{2\to 1}$, $\mathbf{U} = \mathbf{U}^{(t-1)}$, and

[1] Not a very widely-used term in statistics, but one which is often used in the literature relating to these methods for some reason.

$$\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)})}{\partial(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)})}\right| = \begin{vmatrix} 1 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 1 \\ 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{vmatrix} = 1.$$

◁

Assume that for the corresponding backward move we draw $\mathbf{U}^{(t)} \sim g_{2\to 1}$ and set $(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)}) = \mathbf{T}_{2\to 1}(\mathbf{X}^{(t)}, \mathbf{U}^{(t)})$. Then the probability of accepting the backward move is

$$\alpha(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) = \min\left\{1, \frac{f(\mathbf{x}^{(t-1)})g_{1\to 2}(\mathbf{u}^{(t-1)})}{f(\mathbf{x}^{(t)})g_{2\to 1}(\mathbf{u}^{(t)})}\left|\frac{\partial \mathbf{T}_{2\to 1}(\mathbf{x}^{(t)}, \mathbf{u}^{(t)})}{\partial(\mathbf{x}^{(t)}, \mathbf{u}^{(t)})}\right|\right\},$$

We then obtain that

$$
\begin{aligned}
& \int_{\mathbf{x}^{(t-1)}\in A}\int_{\{\mathbf{u}^{(t-1)}\colon \mathbf{x}^{(t)}\in B\}} \alpha(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})g_{1\to 2}(\mathbf{u}^{(t-1)})f(\mathbf{x}^{(t-1)})\, d\mathbf{u}^{(t-1)}d\mathbf{x}^{(t-1)} \\
= & \int_{\mathbf{x}^{(t-1)}\in A}\int_{\{\mathbf{u}^{(t-1)}\colon \mathbf{x}^{(t)}\in B\}} \min\left\{1, \frac{f(\mathbf{x}^{(t)})g_{2\to 1}(\mathbf{u}^{(t)})}{f(\mathbf{x}^{(t-1)})g_{1\to 2}(\mathbf{u}^{(t-1)})}\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})}{\partial(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})}\right|\right\} g_{1\to 2}(\mathbf{u}^{(t-1)})f(\mathbf{x}^{(t-1)})\, d\mathbf{u}^{(t-1)}d\mathbf{x}^{(t-1)} \\
= & \int_{\mathbf{x}^{(t-1)}\in A}\int_{\{\mathbf{u}^{(t-1)}\colon \mathbf{x}^{(t)}\in B\}} \min\left\{f(\mathbf{x}^{(t-1)})g_{1\to 2}(\mathbf{u}^{(t-1)}), f(\mathbf{x}^{(t)})g_{2\to 1}(\mathbf{u}^{(t)})\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})}{\partial(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})}\right|\right\} d\mathbf{u}^{(t-1)}d\mathbf{x}^{(t-1)} \\
= & \int_{\{\mathbf{u}^{(t)}\colon \mathbf{x}^{(t-1)}\in A\}}\int_{\mathbf{x}^{(t)}\in B} \min\left\{f(\mathbf{x}^{(t-1)})g_{1\to 2}(\mathbf{u}^{(t-1)}), f(\mathbf{x}^{(t)})g_{2\to 1}(\mathbf{u}^{(t)})\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})}{\partial(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})}\right|\right\}\left|\frac{\partial \mathbf{T}_{2\to 1}(\mathbf{x}^{(t)},\mathbf{u}^{(t)})}{\partial(\mathbf{x}^{(t)},\mathbf{u}^{(t)})}\right| d\mathbf{x}^{(t)}d\mathbf{u}^{(t)} \\
= & \int_{\{\mathbf{u}^{(t)}\colon \mathbf{x}^{(t-1)}\in A\}}\int_{\mathbf{x}^{(t)}\in B} \min\left\{\frac{f(\mathbf{x}^{(t-1)})g_{1\to 2}(\mathbf{u}^{(t-1)})}{f(\mathbf{x}^{(t)})g_{2\to 1}(\mathbf{u}^{(t)})}\left|\frac{\partial \mathbf{T}_{2\to 1}(\mathbf{x}^{(t)},\mathbf{u}^{(t)})}{\partial(\mathbf{x}^{(t)},\mathbf{u}^{(t)})}\right|, 1\right\} g_{2\to 1}(\mathbf{u}^{(t)})f(\mathbf{x}^{(t)})\, d\mathbf{x}^{(t)}d\mathbf{u}^{(t)} \\
= & \int_{\{\mathbf{u}^{(t)}\colon \mathbf{x}^{(t-1)}\in A\}}\int_{\mathbf{x}^{(t)}\in B} \alpha(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})g_{2\to 1}(\mathbf{u}^{(t)})f(\mathbf{x}^{(t)})\, d\mathbf{x}^{(t)}d\mathbf{u}^{(t)}
\end{aligned}
$$

The fourth row is obtained from the third row by using the change of variable formula. Note that $\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})}{\partial(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})}\right| \cdot \left|\frac{\partial \mathbf{T}_{2\to 1}(\mathbf{x}^{(t)},\mathbf{u}^{(t)})}{\partial(\mathbf{x}^{(t)},\mathbf{u}^{(t)})}\right| = 1$ as $\mathbf{T}^{2\to 1} = \mathbf{T}_{1\to 2}^{-1}$. Equation 8.1 implies by, analogy with proposition 5.1, detailed balance, i.e. the Markov chain generated by the above method has indeed f as invariant distribution.

Remark 8.1 (Detailed Balance). In a general state space detailed balance holds if

$$
\int_{\mathbf{x}^{(t-1)}\in A}\int_{\mathbf{x}^{(t)}\in B} \pi(d\mathbf{x}^{(t-1)})K(\mathbf{x}^{(t-1)}, d\mathbf{x}^{(t)}) = \int_{\mathbf{x}^{(t-1)}\in A}\int_{\mathbf{x}^{(t)}\in B} \pi(d\mathbf{x}^{(t)})K(\mathbf{x}^{(t)}, d\mathbf{x}^{(t-1)})
$$

for all Borel sets A, B, where $K(\mathbf{x}^{(t-1)}, B) = \mathbb{P}(\mathbf{X}^{(t)} \in B|\mathbf{X}^{(t-1)} = \mathbf{x}^{(t-1)})$. Now we have that

$$
\int_{\mathbf{x}^{(t)}\in B} K(\mathbf{x}^{(t-1)}, d\mathbf{x}^{(t)}) = K(\mathbf{x}^{(t-1)}, B) = \mathbb{P}(\mathbf{X}^{(t)} \in B|\mathbf{X}^{(t-1)} = \mathbf{x}^{(t-1)}) = \int_{\{\mathbf{u}^{(t-1)}\colon \mathbf{x}^{(t)}\in B\}} \alpha(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})g_{1\to 2}(\mathbf{u}^{(t-1)})\, d\mathbf{u}^{(t-1)} + \mathbb{I}_B(\mathbf{x}^{(t-1)})(1 - a(\mathbf{x}^{(t-1)})).
$$

As

$$
\int_{\mathbf{x}^{(t-1)}\in A} \mathbb{I}_B(\mathbf{x}^{(t-1)})(1 - a(\mathbf{x}^{(t-1)}))\, \pi(d\mathbf{x}^{(t-1)}) = \int_{\mathbf{x}\in A\cap B} (1 - a(\mathbf{x}))\, \pi(d\mathbf{x}) = \int_{\mathbf{x}^{(t)}\in B} \mathbb{I}_A(\mathbf{x}^{(t)})(1 - a(\mathbf{x}^{(t)}))\, \pi(d\mathbf{x}^{(t)})
$$

detailed balance is equivalent to

$$
\int_{\mathbf{x}^{(t-1)}\in A}\int_{\{\mathbf{u}^{(t-1)}\colon \mathbf{x}^{(t)}\in B\}} \alpha(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})g_{1\to 2}(\mathbf{u}^{(t-1)})f(\mathbf{x}^{(t-1)})\, d\mathbf{u}^{(t-1)}d\mathbf{x}^{(t-1)} = \int_{\{\mathbf{u}^{(t)}\colon \mathbf{x}^{(t-1)}\in A\}}\int_{\mathbf{x}^{(t)}\in B} \alpha(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})g_{2\to 1}(\mathbf{u}^{(t)})f(\mathbf{x}^{(t)})\, d\mathbf{x}^{(t)}d\mathbf{u}^{(t)}
$$

which is what we have shown in (8.1). (On the left hand side $\mathbf{x}^{(t)} := \mathbf{x}^{(t)}(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})$ is defined implicitly such that $(\mathbf{x}^{(t)}, \mathbf{u}^{(t)}) = \mathbf{T}_{1\to 2}(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})$. On the right hand side $\mathbf{x}^{(t-1)} := \mathbf{x}^{(t-1)}(\mathbf{x}^{(t)}, \mathbf{u}^{(t)})$ is defined implicitly such that $(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)}) = \mathbf{T}_{2\to 1}(\mathbf{x}^{(t)}, \mathbf{u}^{(t)})$.)

Note that this argument holds even if $\mathbf{x}^{(t-1)}$ and $\mathbf{x}^{(t)}$ have different dimension, as long as the joint vectors $(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})$ and $(\mathbf{x}^{(t)}, \mathbf{u}^{(t)})$ have common dimension. Thus we can use the above approach for sampling from variable dimension models, as the next section shows.

8.3 The Reversible Jump Algorithm

Returning to the developments of Section 8.1 we need to draw an MCMC sample from the joint posterior distribution

$$f^{\text{post}}(k, \boldsymbol{\theta}) = \frac{p_k f_k^{\text{prior}}(\boldsymbol{\theta}) l_k(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta})}{\sum_\kappa p_\kappa \int_{\Theta_\kappa} l_\kappa^{\text{prior}}(\boldsymbol{\vartheta}) l_\kappa(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\vartheta})\, d\boldsymbol{\vartheta}} \tag{8.1}$$

defined on

$$\Theta := \bigcup_k (\{k\} \times \Theta_k)$$

A slight modification of the approach discussed in Section 8.2 allows us to draw samples from $f^{\text{post}}(k, \boldsymbol{\theta})$ by jumping between the models. This leads to the reversible jump algorithm proposed by Green (1995):

Algorithm 8.1 (Reversible jump). Starting with $k^{(0)}$ and $\boldsymbol{\theta}^{(0)}$ iterate for $t = 1, 2, \ldots$

1. Select new model $\mathcal{M}_k$ with probability $\rho_{k^{(t-1)} \to k}$ (where $\sum_k \rho_{k^{(t-1)} \to k} = 1$).
   (With probability $\rho_{k^{(t-1)} \to k^{(t-1)}}$ update the parameters of $\mathcal{M}_{k^{(t-1)}}$ and skip the remaining steps.)
2. Generate $\mathbf{u}^{(t-1)} \sim g_{k^{(t-1)} \to k}$
3. Set $(\boldsymbol{\theta}, \mathbf{u}) := T_{k^{(t-1)} \to k}(\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})$.
4. Compute

$$\alpha := \min \left\{ 1, \frac{f^{\text{post}}(k, \boldsymbol{\theta}) \rho_{k \to k^{(t-1)}} g_{k \to k^{(t-1)}}(\mathbf{u})}{f^{\text{post}}(k^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) \rho_{k^{(t-1)} \to k} g_{k^{(t-1)} \to k}(\mathbf{u}^{(t-1)})} \left| \frac{\partial \mathbf{T}_{k^{(t-1)} \to k}(\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})}{\partial(\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})} \right| \right\}$$

5. With probability α set $k^{(t)} = k$ and $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}$, otherwise keep $k^{(t)} = k^{(t-1)}$ and $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.

Note that as in Section 8.2 we need that $\mathbf{T}_{k \to l}$ is a diffeomorphism with $\mathbf{T}_{l \to k} = \mathbf{T}_{k \to l}^{-1}$. Note that this implies that $(\boldsymbol{\theta}, \mathbf{u})$ has the same dimension as $(\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})$ ("dimension matching"). It is possible (and a rather popular choice) that $\mathbf{u}$ or $\mathbf{u}^{(t-1)}$ is zero-dimensional, as long as the dimensions of $(\boldsymbol{\theta}, \mathbf{u})$ and $(\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})$ match. Often $\rho^{k \to l}$ is only positive if the models $\mathcal{M}_k$ and $\mathcal{M}_l$ are close in some sense. Note however that $\rho_{k \to l} = 0$ implies $\rho_{l \to k} = 0$. In general, the transdimensional moves should be designed such that they yield a high probability of acceptance.

Remark 8.2. The probability of acceptance of the reversible jump algorithm does not depend on the normalising constant of the joint posterior $f^{(post)}(k, \boldsymbol{\theta})$ (i.e. the denominator of (8.1)).

Proposition 8.1. *The joint posterior $f^{post}(k, \boldsymbol{\theta})$ is under the above conditions the invariant distribution of the reversible jump algorithm.*

Proof. From Remark 8.1 we have using $\mathbf{x} := (k, \boldsymbol{\theta})$ and the fact that k is discrete (i.e. an integral with respect to k is a sum) that detailed balance holds, as

$$
\begin{aligned}
& \sum_{k^{(t-1)} \in \mathcal{A}} \int_{\boldsymbol{\theta}^{(t-1)} \in A_{k^{(t-1)}}} \sum_{k^{(t)} \in \mathcal{B}} \int_{\{\mathbf{u}^{(t-1)} :\, \boldsymbol{\theta}^{(t)} \in B_{k^{(t)}}\}} \alpha((k^{(t)}, \boldsymbol{\theta}^{(t)}) | (k^{(t-1)}, \boldsymbol{\theta}^{(t-1)})) \rho_{k^{(t-1)} \to k^{(t)}} g_{k^{(t-1)} \to k^{(t)}}(\mathbf{u}^{(t-1)}) f^{\text{post}}(k^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) \, d\mathbf{u}^{(t-1)} d\boldsymbol{\theta}^{(t-1)} \\
= & \sum_{k^{(t-1)} \in \mathcal{A}} \int_{\boldsymbol{\theta}^{(t-1)} \in A_{k^{(t-1)}}} \sum_{k^{(t)} \in \mathcal{B}} \int_{\{\mathbf{u}^{(t-1)} :\, \boldsymbol{\theta}^{(t)} \in B_{k^{(t)}}\}} \min \left\{ 1, \frac{f^{\text{post}}(k^{(t)}, \boldsymbol{\theta}^{(t)}) \rho_{k^{(t)} \to k^{(t-1)}} g_{k^{(t)} \to k^{(t-1)}}(\mathbf{u}^{(t)})}{f^{\text{post}}(k^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) \rho_{k^{(t-1)} \to k^{(t)}} g_{k^{(t-1)} \to k^{(t)}}(\mathbf{u}^{(t-1)})} \left| \frac{\partial \mathbf{T}_{k^{(t-1)} \to k^{(t)}}(\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})}{\partial (\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})} \right| \right\} \\
& \rho_{k^{(t-1)} \to k^{(t)}} g_{k^{(t-1)} \to k^{(t)}}(\mathbf{u}^{(t-1)}) f^{\text{post}}(k^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) \, d\mathbf{u}^{(t-1)} d\boldsymbol{\theta}^{(t-1)} \\
= & \sum_{k^{(t-1)} \in \mathcal{A}} \int_{\boldsymbol{\theta}^{(t-1)} \in A_{k^{(t-1)}}} \sum_{k^{(t)} \in \mathcal{B}} \int_{\{\mathbf{u}^{(t-1)} :\, \boldsymbol{\theta}^{(t)} \in B_{k^{(t)}}\}} \min \left\{ f^{\text{post}}(k^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) \rho_{k^{(t-1)} \to k^{(t)}} g_{k^{(t-1)} \to k^{(t)}}(\mathbf{u}^{(t-1)}) \right. , \\
& \left. f^{\text{post}}(k^{(t)}, \boldsymbol{\theta}^{(t)}) \rho_{k^{(t)} \to k^{(t-1)}} g_{k^{(t)} \to k^{(t-1)}}(\mathbf{u}^{(t)}) \left| \frac{\partial \mathbf{T}_{k^{(t-1)} \to k^{(t)}}(\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})}{\partial (\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})} \right| \right\} \, d\mathbf{u}^{(t-1)} d\boldsymbol{\theta}^{(t-1)} \\
= & \sum_{k^{(t-1)} \in \mathcal{A}} \int_{\{\mathbf{u}^{(t)} :\, \boldsymbol{\theta}^{(t-1)} \in A_{k^{(t-1)}}\}} \sum_{k^{(t)} \in \mathcal{B}} \int_{\boldsymbol{\theta}^{(t)} \in B_{k^{(t)}}} \min \left\{ f^{\text{post}}(k^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) \rho_{k^{(t-1)} \to k^{(t)}} g_{k^{(t-1)} \to k^{(t)}}(\mathbf{u}^{(t-1)}) \right. , \\
& \left. f^{\text{post}}(k^{(t)}, \boldsymbol{\theta}^{(t)}) \rho_{k^{(t)} \to k^{(t-1)}} g_{k^{(t)} \to k^{(t-1)}}(\mathbf{u}^{(t)}) \left| \frac{\partial \mathbf{T}_{k^{(t-1)} \to k^{(t)}}(\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})}{\partial (\boldsymbol{\theta}^{(t-1)}, \mathbf{u}^{(t-1)})} \right| \right\} \left| \frac{\partial \mathbf{T}_{k^{(t)} \to k^{(t-1)}}(\boldsymbol{\theta}^{(t)}, \mathbf{u}^{(t)})}{\partial (\boldsymbol{\theta}^{(t)}, \mathbf{u}^{(t)})} \right| \, d\boldsymbol{\theta}^{(t)} d\mathbf{u}^{(t)} \\
= & \sum_{k^{(t-1)} \in \mathcal{A}} \int_{\{\mathbf{u}^{(t)} :\, \boldsymbol{\theta}^{(t-1)} \in A_{k^{(t-1)}}\}} \sum_{k^{(t)} \in \mathcal{B}} \int_{\boldsymbol{\theta}^{(t)} \in B_{k^{(t)}}} \min \left\{ f^{\text{post}}(k^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) \rho_{k^{(t-1)} \to k^{(t)}} g_{k^{(t-1)} \to k^{(t)}}(\mathbf{u}^{(t-1)}) \left| \frac{\partial \mathbf{T}_{k^{(t)} \to k^{(t-1)}}(\boldsymbol{\theta}^{(t)}, \mathbf{u}^{(t)})}{\partial (\boldsymbol{\theta}^{(t)}, \mathbf{u}^{(t)})} \right| \right. , \\
& \left. f^{\text{post}}(k^{(t)}, \boldsymbol{\theta}^{(t)}) \rho_{k^{(t)} \to k^{(t-1)}} g_{k^{(t)} \to k^{(t-1)}}(\mathbf{u}^{(t)}) \right\} \, d\boldsymbol{\theta}^{(t)} d\mathbf{u}^{(t)} \\
= & \sum_{k^{(t-1)} \in \mathcal{A}} \int_{\{\mathbf{u}^{(t)} :\, \boldsymbol{\theta}^{(t-1)} \in A_{k^{(t-1)}}\}} \sum_{k^{(t)} \in \mathcal{B}} \int_{\boldsymbol{\theta}^{(t)} \in B_{k^{(t)}}} \min \left\{ \frac{f^{\text{post}}(k^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) \rho_{k^{(t-1)} \to k^{(t)}} g_{k^{(t-1)} \to k^{(t)}}(\mathbf{u}^{(t-1)})}{f^{\text{post}}(k^{(t)}, \boldsymbol{\theta}^{(t)}) \rho_{k^{(t)} \to k^{(t-1)}} g_{k^{(t)} \to k^{(t-1)}}(\mathbf{u}^{(t)})} \left| \frac{\partial \mathbf{T}_{k^{(t)} \to k^{(t-1)}}(\boldsymbol{\theta}^{(t)}, \mathbf{u}^{(t)})}{\partial (\boldsymbol{\theta}^{(t)}, \mathbf{u}^{(t)})} \right| , 1 \right\} \\
& \rho_{k^{(t)} \to k^{(t-1)}} g_{k^{(t)} \to k^{(t-1)}}(\mathbf{u}^{(t)}) f^{\text{post}}(k^{(t)}, \boldsymbol{\theta}^{(t)}) \, d\boldsymbol{\theta}^{(t)} d\mathbf{u}^{(t)} \\
= & \sum_{k^{(t-1)} \in \mathcal{A}} \int_{\{\mathbf{u}^{(t)} :\, \boldsymbol{\theta}^{(t-1)} \in A_{k^{(t-1)}}\}} \sum_{k^{(t)} \in \mathcal{B}} \int_{\boldsymbol{\theta}^{(t)} \in B_{k^{(t)}}} \alpha((k^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) | (k^{(t)}, \boldsymbol{\theta}^{(1)})) \rho_{k^{(t)} \to k^{(t-1)}} g_{k^{(t)} \to k^{(t-1)}}(\mathbf{u}^{(t)}) f^{\text{post}}(k^{(t)}, \boldsymbol{\theta}^{(t)}) \, d\boldsymbol{\theta}^{(t)} d\mathbf{u}^{(t)}
\end{aligned}
$$

for all $A = \bigcup_{k \in \mathcal{A}} \{k\} \times A_k \subset \Theta$, $B = \bigcup_{k \in \mathcal{B}} \{k\} \times A_k \subset \Theta$ □

Example 8.2. Consider a problem with two possible models $\mathcal{M}_1$ and $\mathcal{M}_2$. The model $\mathcal{M}_1$ has a single parameter $\theta_1 \in [0,1]$. The model $\mathcal{M}_2$ has two parameters $\theta_1, \theta_2 \in D$ with triangular domain $D = \{(\theta_1, \theta_2) : 0 \leq \theta_2 \leq \theta_1 \leq 1\}$. The joint posterior of $(k, \boldsymbol{\theta})$ is

$$f^{\text{post}}(k, \boldsymbol{\theta}) \propto p_k f_k^{\text{prior}}(\boldsymbol{\theta}) l_k(\mathbf{y}_1, \ldots, \mathbf{y}_n | \boldsymbol{\theta})$$

We need to propose two moves $\mathbf{T}_{1\to2}$ and $\mathbf{T}_{2\to1}$ such that $\mathbf{T}_{1\to2} = \mathbf{T}_{2\to1}^{-1}$. Assume that we want to get from model $\mathcal{M}_2$ to model $\mathcal{M}_1$ by dropping θ_2, i.e.

$$\mathbf{T}_{2\to1}(\theta_1, \theta_2) = (\theta_1, \star)$$

A move that is compatible (in the sense that $\mathbf{T}_{1\to2} = \mathbf{T}_{2\to1}^{(-1)}$) with $\mathbf{T}_{2\to1}$ is

$$\mathbf{T}_{1\to2}(\theta, u) = (\theta, u\theta).$$

When setting $\star$ to θ_2/θ_1 we have that $\mathbf{T}_{1\to2} = \mathbf{T}_{2\to1}^{-1}$. If we draw $U \sim \mathrm{U}[0,1]$ we have that $\mathbf{T}_{1\to2}(\theta, U) \in D$. The Jacobian is

$$\left| \frac{\partial \mathbf{T}_{1\to2}(\theta, u)}{\partial(\theta, u)} \right| = \begin{vmatrix} 1 & 0 \\ u & \theta \end{vmatrix} = |\theta| = \theta.$$

Using the formula for the derivative of the inverse we obtain that

$$\left| \frac{\partial \mathbf{T}_{2\to1}(\theta_1, \theta_2)}{\partial(\theta_1, \theta_2)} \right| = \left| \left(\left. \frac{\partial \mathbf{T}_{1\to2}(\theta, u)}{\partial(\theta, u)} \right|_{(\theta,u)=\mathbf{T}_{2\to1}(\theta_1,\theta_2)} \right)^{-1} \right| = 1 / \left| \left. \frac{\partial \mathbf{T}_{1\to2}(\theta, u)}{\partial(\theta, u)} \right|_{(\theta,u)=\mathbf{T}_{2\to1}(\theta_1,\theta_2)} \right| = 1/\theta_1$$

The moves between the models $\mathcal{M}_1$ and $\mathcal{M}_2$ (and vice versa) keep θ_1 constant. An algorithm based only on these two moves will not yield an irreducible chain. Thus we need to include fixed-dimensional moves. In this simple example it is enough to include a single fixed-dimensional move: If we are in model $\mathcal{M}_1$ then with probability $1/2$ we carry out a Metropolis update (e.g. using an independent proposal from $\mathrm{U}[0,1]$). In order to obtain an irreducible and fast mixing chain in more complex models, it is typically necessary to allow for fixed-dimensional moves in all models.

This setup corresponds to $\rho_{1\to1} = 1/2$, $\rho_{1\to2} = 1/2$, $\rho_{2\to1} = 1$, $\rho_{2\to2} = 0$.

The reversible jump algorithm specified above consists of iterating for $t = 1, 2, 3, \ldots$

– If the current model is $\mathcal{M}_1$ (i.e. $k^{(t-1)} = 1$):
  * With probability $1/2$ perform an update of $\theta^{(t-1)}$ within model $\mathcal{M}_1$, i.e.
    1. Generate $\theta_1 \sim \mathrm{U}[0,1]$.
    2. Compute the probability of acceptance
    $$\alpha = \min \left\{ 1, \frac{f_1^{\text{prior}}(\theta_1) l_1(\mathbf{y}_1, \ldots, \mathbf{y}_n | \theta_1)}{f_1^{\text{prior}}(\theta_1^{(t-1)}) l_1(\mathbf{y}_1, \ldots, \mathbf{y}_n | \theta_1^{(t-1)})} \right\}$$
    3. With probability α set $\theta^{(t)} = \theta$, otherwise keep $\theta^{(t)} = \theta^{(t-1)}$.
  * Otherwise attempt a jump to model $\mathcal{M}_2$, i.e.
    1. Generate $u^{(t-1)} \sim \mathrm{U}[0,1]$
    2. Set $(\theta_1, \theta_2) := T_{1\to2}(\theta^{(t-1)}, u^{(t-1)}) = (\theta^{(t-1)}, u^{(t-1)}\theta^{(t-1)})$.
    3. Compute
    $$\alpha = \min \left\{ 1, \frac{p_2 \cdot f_2^{\text{prior}}(\theta_1, \theta_2) l_2(\mathbf{y}_1, \ldots, \mathbf{y}_n | \theta_1, \theta_2) \cdot 1}{p_1 \cdot f_1^{\text{prior}}(\theta_1^{(t-1)}) l_1(\mathbf{y}_1, \ldots, \mathbf{y}_n | \theta_1^{(t-1)}) \cdot 1/2 \cdot 1} \cdot \theta_1^{(t-1)} \right\}$$
    4. With probability α set $k^{(t)} = 2$ and $\boldsymbol{\theta}^{(t)} = (\theta_1, \theta_2)$, otherwise keep $k = 1$ and $\theta^{(t)} = \theta^{(t-1)}$.

– Otherwise, if the current model is $\mathcal{M}_2$ (i.e. $k^{(t-1)} = 2$) attempt a jump to $\mathcal{M}_1$:

1. Set $(\theta, u) := T_{2\to1}(\theta_1^{(t-1)}, \theta_2^{(t-1)}) = (\theta_1^{(t-1)}, \theta_2^{(t-1)}/\theta_1^{(t-1)})$.
2. Compute

$$\alpha = \min\left\{1, \frac{p_1 \cdot f_1^{\text{prior}}(\theta_1) l_1(\mathbf{y}_1, \ldots, \mathbf{y}_n|\theta_1) \cdot 1/2 \cdot 1}{p_2 \cdot f_2^{\text{prior}}(\theta_1^{(t-1)}, \theta_2^{(t-1)}) l_2(\mathbf{y}_1, \ldots, \mathbf{y}_n|\theta_1^{(t-1)}, \theta_2^{(t-1)}) \cdot 1} \cdot \frac{1}{\theta_1^{(t-1)}}\right\}$$

3. With probability α set $k^{(t)} = 1$ and $\theta^{(t)} = \theta$, otherwise keep $k = 2$ and $\boldsymbol{\theta}^{(t)} = (\theta_1^{(t-1)}, \theta_2^{(t-1)})$. ◁

Having established that it's possible to construct algorithms within this framework, we now turn our attention to a more realistic example — revisiting the Gaussian mixture model we considered previously but now including the number of components amongst the unknown parameters.

Example 8.3 (Mixture of Gaussians with a variable number of components). Consider again the Gaussian mixture model from example 7.1, in which we assumed that the density of y_i is from a mixture of Gaussians

$$f(y_i|\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k, \tau_1, \ldots, \tau_k) = \sum_{\kappa=1}^{k} \pi_\kappa \phi_{(\mu_\kappa, 1/\tau_\kappa)}(y_i).$$

Suitable prior distributions are a Dirichlet distribution for $(\pi_1, \ldots, \pi_k)$, a Gaussian for μ_κ and a Gamma distribution for τ_κ. In order to ensure identifiability we assume the μ_κ are ordered, i.e. $\mu_1 < \ldots < \mu_k$ and make the corresponding change to the posterior density (to compensate for setting the density to zero for all configurations which fail to satisfy this ordering constraint, the density of all configurations compatible with the constraint must be increased by a factor of $k!$). In example 7.1 we assumed that the number of components k is known. In this example we assume that we want to estimate the number of components k as well. Note that the dimension of the parameter vector $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k, \tau_1, \ldots, \tau_k)$ depends on k, so we need to use the reversible jump algorithm to move between models with different numbers of components. Denote with p_k the prior distribution of the number of components.

The easiest way of moving between models, is to allow for two simple transdimensional moves: adding one new component ("birth move", $k \to k+1$) and dropping one component ("death move", $k+1 \to k$).

Consider the birth move first. We draw the mean and precision parameters of the new component, which we will call μ_{k+1} and τ_{k+1} for convenience, from the corresponding prior distributions. Furthermore we draw the prior probability of the new component $\pi_{k+1} \sim \text{Beta}(1, k)$. As we need that the sum of the prior probabilities $\sum_{\kappa=1}^{k+1} \pi_\kappa = 1$, we have to rescale the other prior probabilities to $\pi_\kappa = \pi_\kappa^{(t-1)}(1 - \pi_{k+1})$ $(\kappa = 1, \ldots, k)$. Putting this into the notation of the reversible jump algorithm, we draw

$$u_1^{(t-1)} \sim g_1, \qquad u_2^{(t-1)} \sim g_2, \qquad u_3^{(t-1)} \sim g_3,$$

and set

$$\pi_{k+1} = u_1^{(t-1)}, \qquad \mu_{k+1} = u_2^{(t-1)}, \qquad \tau_{k+1} = u_3^{(t-1)}$$

with g_1 being the density of the $\text{Beta}(1, k)$ distribution, g_2 being the density of the prior distribution on the μ_κ, and g_3 being the density of the prior distribution on τ_κ. The corresponding transformation $T_{k\to k+1}$ is

$$\begin{pmatrix} \pi_1 \\ \vdots \\ \pi_k \\ \pi_{k+1} \\ \vdots \\ \mu_{k+1} \\ \vdots \\ \tau_{k+1} \end{pmatrix} = T_{k\to k+1} \begin{pmatrix} \pi_1^{(t-1)} \\ \vdots \\ \pi_k^{(t-1)} \\ \vdots \\ u_1^{(t-1)} \\ u_2^{(t-1)} \\ u_3^{(t-1)} \end{pmatrix} = \begin{pmatrix} \pi_1^{(t-1)}(1-u_1^{(t-1)}) \\ \vdots \\ \pi_k^{(t-1)}(1-u_1^{(t-1)}) \\ u_1^{(t-1)} \\ \vdots \\ u_2^{(t-1)} \\ \vdots \\ u_3^{(t-1)} \end{pmatrix}$$

The determinant of the Jacobian of $T_{k\to k+1}$ is

$$(1-u_1^{(t-1)})^k$$

Next we consider the death move, which is the move in the opposite direction. Assume that we drop the κ-th component. To keep the notation simple we assume $\kappa = k+1$. In order to maintain the constraint $\sum_{\iota=1}^{k} \pi_\iota = 1$ we need to rescale the prior probabilities to $\pi_\iota = \pi_\iota/(1-\pi_{k+1}^{(t-1)})$ $(\iota = 1, \ldots, k)$. The corresponding transformation is

$$\begin{pmatrix} \pi_1 \\ \vdots \\ \pi_k \\ \vdots \\ u_1 \\ u_2 \\ u_3 \end{pmatrix} = T_{k+1\to k} \begin{pmatrix} \pi_1^{(t-1)} \\ \vdots \\ \pi_{k+1}^{(t-1)} \\ \vdots \\ \mu_{k+1}^{(t-1)} \\ \vdots \\ \tau_{k+1}^{(t-1)} \end{pmatrix} = \begin{pmatrix} \pi_1^{(t-1)}/(1-\pi_{k+1}^{(t-1)}) \\ \vdots \\ \pi_k^{(t-1)}/(1-\pi_{k+1}^{(t-1)}) \\ \vdots \\ \pi_{k+1}^{(t-1)} \\ \mu_{k+1}^{(t-1)} \\ \tau_{k+1}^{(t-1)} \end{pmatrix}$$

It is easy to see that $T_{k+1\to k} = T_{k\to k+1}^{-1}$, and thus the modulus of the Jacobian of $T_{k+1\to k}$ is

$$\frac{1}{(1-\pi_{k+1}^{(t-1)})^k}$$

Now that we have specified both the birth move and the complementary death move, we can state the probability of accepting a birth move from a model with k components to a model with $k+1$ components. It is

$$\min\left\{1, \frac{p_{k+1} f_{k+1}^{\text{prior}}(\boldsymbol{\theta}) l(y_1, \ldots, y_n|\boldsymbol{\theta})}{p_k f_k^{\text{prior}}(\boldsymbol{\theta}^{(t-1)}) l(y_1, \ldots, y_n|\boldsymbol{\theta}^{(t-1)})} \cdot \frac{(k+1)!}{k!} \cdot \frac{\rho_{k+1\to k}/(k+1)}{\rho_{k\to k+1} g_1(u_1^{(t-1)}) g_2(u_2^{(t-1)}) g_3(u_3^{(t-1)})} \cdot (1-u_1^{(t-1)})^k\right\}$$

The factors $(k+1)!$ and $k!$ are required to account for the fact that the model is not uniquely parameterised, and any permutation of the indexes of the components yields the same model. $1/(k+1)$ in the probability of picking one of the $k+1$ components in the death step.

The probability of accepting the death step is the reciprocal of the above probability of acceptance of the birth step.

There are other (and more efficient) possibilities of moving between models of different orders. A very efficient pair of moves corresponds to splitting and (in the opposite direction) merging components. For a more detailed review of this model see (Richardson and Green, 1997). ◁

9. Simulated Annealing

9.1 A Monte-Carlo method for finding the mode of a distribution

So far we have studied various methods that allow for approximating expectations $\mathbb{E}(\varphi(\mathbf{X}))$ by ergodic averages $\frac{1}{T}\sum_{t=1}^{T}\varphi(\mathbf{X}_i^{(t)})$. This section presents an algorithm for finding the (global) mode(s) of a distribution. For definiteness, in this chapter we define the mode(s) of a distribution to be the set of global maxima of the density, i.e. $\{\boldsymbol{\xi}:\ f(\boldsymbol{\xi}) \geq f(\mathbf{x})\ \forall \mathbf{x}\}$. In Section 9.2 we will extend this idea to finding global extrema of arbitrary functions.

We could estimate the mode of a distribution by the $\mathbf{X}^{(t)}$ with maximal density $f(\mathbf{X}^{(t)})$, this is however a not very efficient strategy. A sample from a Markov chain with $f(\cdot)$ samples from the whole distribution and not only from the mode(s).

This suggests modifying the distribution such that it is more concentrated around the mode(s). One way of achieving this is to consider

$$f_{(\beta)}(x) \propto (f(x))^{\beta}$$

for very large values of β.

Example 9.1 (Normal distribution). Consider the $\mathsf{N}(\mu, \sigma^2)$ distribution with density

$$f_{(\mu,\sigma^2)}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \propto \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

It is easy to see that the mode of the $\mathsf{N}(\mu, \sigma^2)$ distribution is μ. We have that

$$\left(f_{(\mu,\sigma^2)}(x)\right)^{\beta} \propto \left(\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\right)^{\beta} = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2/\beta}\right) \propto f_{(\mu,\sigma^2/\beta)}(x).$$

In other words, the larger β is chosen, the more concentrated the distribution will be around the mode μ. Figure 9.1 illustrates this idea. ◁

The result we have obtained of the Gaussian distribution in the above example actually holds in general. For $\beta \to \infty$ the distribution defined by the density $f_{(\beta)}(x)$ converges to a distribution that has all mass on the mode(s) of f (see figure 9.2 for an example). It is instructive to see informally why this is the case when considering a discrete random variable with probability mass function $p(\cdot)$ and finite support E. Denote with E^* the set of modes of p, i.e. $p(\xi) \geq p(x)$ for all $\xi \in E^*$ and $x \in E$, and with $m := p(\xi)$ with $\xi \in E^*$. Then

$$p_{(\beta)}(x) = \frac{(p(x))^{\beta}}{\sum_{x\in E^*}(p(x))^{\beta} + \sum_{x\in E\setminus E^*}(p(x))^{\beta}} = \frac{(p(x)/m)^{\beta}}{\sum_{x\in E^*} 1 + \sum_{x\in E\setminus E^*}(p(x)/m)^{\beta}} \overset{\beta\to+\infty}{\longrightarrow} \begin{cases} 1/|E^*| & \text{if } x \in E^* \\ 0 & \text{if} x \notin E^* \end{cases}$$

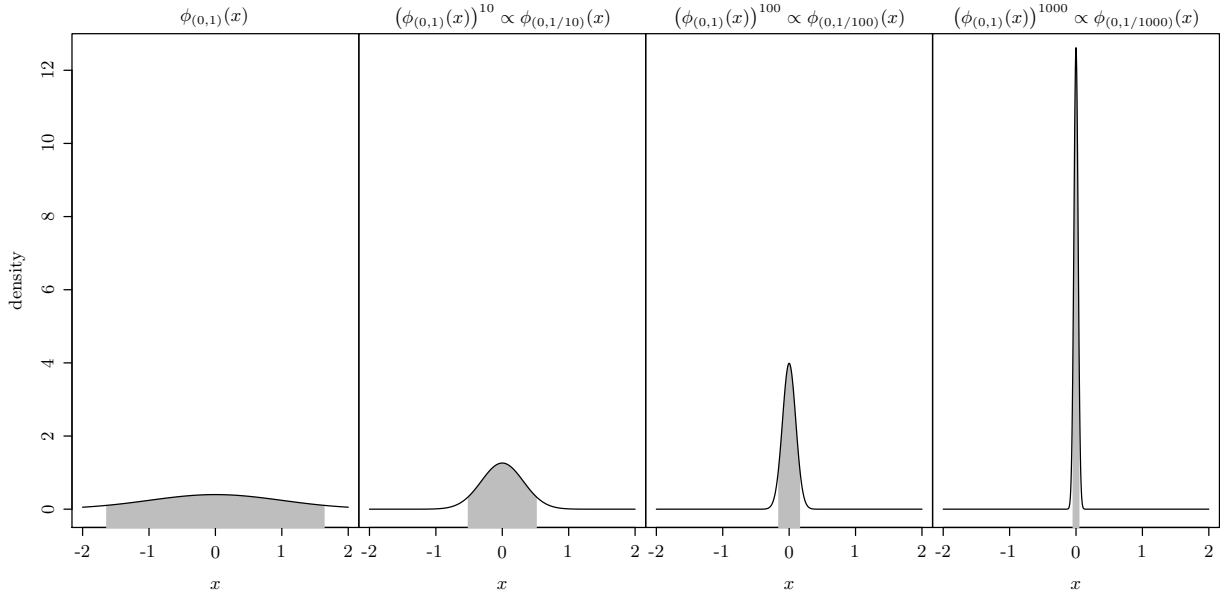


Figure 9.1. Density of the $\mathsf{N}(0,1)$ raised to increasing powers. The areas shaded in grey represent 90% of the probability mass.

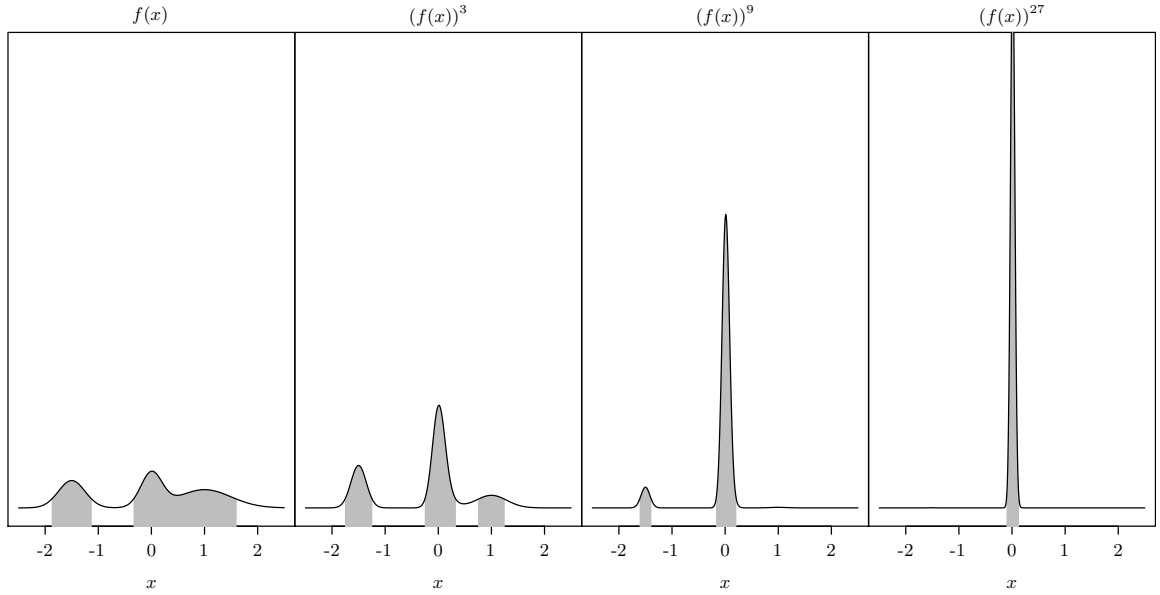


Figure 9.2. An arbitrary multimodal density raised to increasing powers. The areas shaded in grey reach from the 5% to the 95% quantiles.

In the continuous case the distribution is not uniform on the modes (see Hwang, 1980, for details).

We can use a random-walk Metropolis algorithm to sample from $f_{(\beta)}(\cdot)$. The probability of accepting a move from $\mathbf{X}^{(t-1)}$ to $\mathbf{X}$ would be

$$\min\left\{1, \frac{f_{(\beta)}(\mathbf{X})}{f_{(\beta)}(\mathbf{X}^{(t-1)})}\right\} = \min\left\{1, \left(\frac{f(\mathbf{X})}{f(\mathbf{X}^{(t-1)})}\right)^{\beta}\right\}.$$

Note that this probability does not depend on the (generally unknown) normalisation constant of $f_{(\beta)}(\cdot)$. It is however difficult to directly sample from $f_{(\beta)}$ for large values of β: for $\beta \to \infty$ the probability of accepting a newly proposed X becomes 1 if $f(X) > f(X^{(t-1)})$ and 0 otherwise. Thus $X^{(t)}$ converges to a *local* extrema of the density f, however not necessarily a mode of f (i.e. a *global* extremum of the density). Whether $X^{(t)}$ gets caught in a local extremum or not, depends on whether we can reach the mode from the *local* extrema of the density within one step. The following example illustrates this problem.

Example 9.2. Consider the following simple optimisation problem of finding the mode of the distribution defined on $\{1, 2, \ldots, 5\}$ by

$$p(x) = \begin{cases} 0.4 & \text{for } x = 2 \\ 0.3 & \text{for } x = 4 \\ 0.1 & \text{for } x = 1, 3, 5. \end{cases}$$

Figure 9.3 illustrates this distribution. Clearly, the (global) mode of $p(x)$ is at $x = 2$. Assume we want to sample from $p_{(\beta)}(x) \propto p(x)^{\beta}$ using a random walk Metropolis algorithm with proposed value $X = X^{(t-1)} + \varepsilon$ with $\mathbb{P}(\varepsilon = \pm 1) = 0.5$ for $X^{(t-1)} \in \{2, 3, 4\}$, $\mathbb{P}(\varepsilon = +1) = 1$ for $X^{(t-1)} = 1$, and $\mathbb{P}(\varepsilon = -1) = 1$ for $X^{(t-1)} = 5$. In other words, we can either move one to the left, stay in the current value (when the proposed value is rejected), or move one to the right. Note that for $\beta \to +\infty$ the probability for accepting a move from 4 to 3 converges to 0, as $p(4) > p(3)$. As the Markov of chain can only move from 4 to 2 only via 3, it cannot escape the local extremum at 4 for $\beta \to +\infty$. ◁

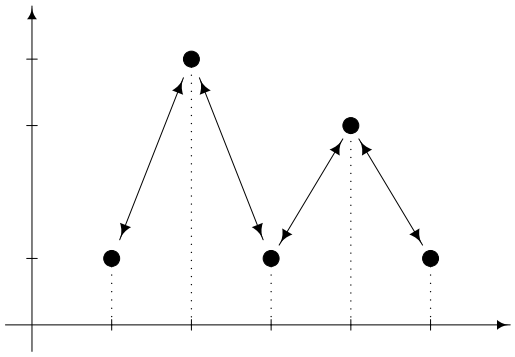


Figure 9.3. Illustration of Example 9.2

For large β the distribution $f_{(\beta)}(\cdot)$ is concentrated around the modes, however at the price of being difficult to sample from: the resulting Markov chain has very poor mixing properties: for large β the algorithm can hardly move away from a local extremum surrounded by areas of low probability (the density of such a distribution would have many local extrema separated by areas where the density is effectively 0).

The key idea of simulated annealing[1] (Kirkpatrick et al., 1983) is to sample from a target distribution that changes over time: $f_{(\beta_t)}(\cdot)$ with $\beta_t \to +\infty$. Before we consider different strategies for choosing the sequence (β_t), we generalise the framework developed so far to finding the global extrema of arbitrary functions.

9.2 Minimising an arbitrary function

Consider that we want to find the global minimum of a function $h : E \to \mathbb{R}$. Finding the global minimum of $H(x)$ is equivalent to finding the mode of a distribution

$$f(x) \propto \exp(-H(x)) \text{ for } x \in E,$$

if such a distribution exists. In this framework, finding the mode of a density f corresponds to finding the minimum of $-\log(f(x))$. As in the previous section we can raise f to large powers to obtain a distribution

$$f_{(\beta_t)}(x) = (f(x))^{\beta_t} \propto \exp(-\beta_t \cdot H(x)) \text{ for } x \in E.$$

We hope to find the (global) minimum of $H(x)$, which is the (global) mode of the distribution defined by $f_{\beta_t}(x)$, by sampling from a Metropolis-Hastings algorithm. As suggested above we let $\beta_t \to +\infty$. This yields the following algorithm:

Algorithm 9.1 (Simulated Annealing). Starting with $\mathbf{X}^{(0)} := (X_1^{(0)}, \ldots, X_p^{(0)})$ and $\beta^{(0)} > 0$ iterate for $t = 1, 2, \ldots$

1. Increase $\beta^{(t-1)}$ to $\beta^{(t)}$ (see below for different annealing schedules)
2. Draw $\mathbf{X} \sim q(\cdot|\mathbf{X}^{(t-1)})$.
3. Compute
$$\alpha(\mathbf{X}|\mathbf{X}^{(t-1)}) = \min\left\{1, \exp\left(-\beta_t(H(\mathbf{X}) - H(\mathbf{X}^{(t-1)}))\right) \cdot \frac{q(\mathbf{X}^{(t-1)}|\mathbf{X})}{q(\mathbf{X}|\mathbf{X}^{(t-1)})}\right\}.$$
4. With probability $\alpha(\mathbf{X}|\mathbf{X}^{(t-1)})$ set $\mathbf{X}^{(t)} = \mathbf{X}$, otherwise set $\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)}$.

If a random walk Metropolis update is used (i.e. $\mathbf{X} = \mathbf{X}^{(t-1)} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim g(\cdot)$ for a symmetric g), then the probability of acceptance becomes

$$\alpha(\mathbf{X}|\mathbf{X}^{(t-1)}) = \min\left\{1, \exp\left(-\beta_t(H(\mathbf{X}) - H(\mathbf{X}^{(t-1)}))\right)\right\}.$$

Using the same arguments as in the previous section, it is easy to see that the simulated annealing algorithm converges to a *local* minimum of $H(\cdot)$. Whether it will be able to find the *global* minimum depends on how slowly we let the inverse temperature β go to infinity.

Logarithmic tempering When choosing $\beta_t = \frac{\log(1+t)}{\beta_0}$, the inverse temperature increases slow enough that global convergence results can be established for certain special cases. Hajek (1988) established global convergence when $H(\cdot)$ is optimised over a *finite* set using a proposal which is uniform over E and logarithmic tempering with a suitably large β_0.

Assume we choose $\beta_0 = \Delta H$ with $\Delta H := \max_{x,x' \in E} |H(x) - H(x')|$. Then the probability of reaching state x in the t-th step is

[1] The term *annealing* comes from metallurgy and refers to the technique of melting a metal before allowing that metal to cool down slowly in order to reach a lower energy state and consequently produce a tougher metal. Following this analogy, $1/\beta$ is typically referred to as temperature, β as inverse temperature.

$$\mathbb{P}(X^{(t)} = x) = \sum_{\xi} \underbrace{\mathbb{P}(X^{(t)} = x | X^{(t-1)} = \xi)}_{\geq \exp(-\beta_t \Delta H)/|E|} \mathbb{P}(X^{(t-1)} = \xi) \geq \exp(-\beta_t \Delta H)/|E|$$

Using the logarithmic tempering schedule we obtain $\mathbb{P}(X^{(t)} = x) \geq 1/\left((1+t)|E|\right)$ and thus the expected number of visits to state x is

$$\sum_{t=0}^{\infty} \mathbb{P}(X^{(t)} = x) \geq \sum_{t=0}^{\infty} [(1+t)|E|]^{-1} = +\infty.$$

Thus every state is recurrent. As β increases we however spend an ever increasing amount of time in the global minima of x.

On the one hand visiting very state x infinitely often implies that we can escape from local minima. On the other hand, this implies as well that we visit every state x (regardless of how large $H(x)$ is) infinitely often. In other words, the reason why simulated annealing with logarithmic tempering works, is that it still behaves very much like an exhaustive search. However, the only reason why we consider simulated annealing is that exhaustive search would be too slow! For this reason, logarithmic tempering has little practical relevance.

Geometric tempering A popular choice is $\beta_t = \alpha^t \cdot \beta_0$ for some $\alpha > 1$.

Example 9.3. Assume we want to find the maximum of the function

$$H(x) = \left((x-1)^2 - 1\right)^2 + 3 \cdot s(11.56 \cdot x^2), \text{ with } s(x) = \begin{cases} |x| \mod 2 & \text{for } 2k \leq |x| \leq 2k+1 \\ 2 - |x| \mod 2 & \text{for } 2k+1 \leq |x| \leq 2(k+1) \end{cases}$$

for $k \in \mathbb{N}_0$. Figure 9.4 (a) shows $H(x)$ for $x \in [-1, 3]$. The global minimum of $H(x)$ is at $x = 0$. We simulated annealing with a geometric tempering with $\beta_0 = 1$ and $\beta_t = 1.001\beta_{t-1}$ and a random walk Metropolis algorithm with $\varepsilon \sim \text{Cauchy}(0, \sqrt{0.1})$. Figure 9.4 (b) shows the first 1,000 iterations of the Markov chain yielded by the simulated annealing algorithm. Note that when using a Gaussian distribution with small enough a variance the simulated annealing algorithm is very likely to remain in the local minimum at $x \approx 1.8$. ◁

Note that there is no guarantee that the simulated annealing algorithm converges to the global minimum of $H(x)$ in finite time. In practice, it would unrealistic to expect simulated annealing to converge to a *global* minimum, however in most cases it will find a "good" *local* minimum.

9.3 Data Augmentation for Simulated Annealing NOT EXAMINABLE

Two common optimisation problems arise in the evaluation of statistical estimators: *Maximum Likelihood Estimation:* Given $l(\theta; \mathbf{x}) = f_{\mathbf{X}}(\mathbf{x}; \theta)$ compute $\hat{\theta}_{\text{ML}} = \arg\max_{\theta \in \Theta} l(\theta; \mathbf{x})$ and *Maximum a Posteriori Estimation:* Given $l(\theta; \mathbf{x}) = f_{\mathbf{X}}(\mathbf{x}; \theta)$ and prior $f^{\text{prior}}(\theta)$ compute $\hat{\theta}_{\text{MAP}} = \arg\max_{\theta \in \Theta} f^{\text{prior}}(\theta) l(\theta; \mathbf{x})$.

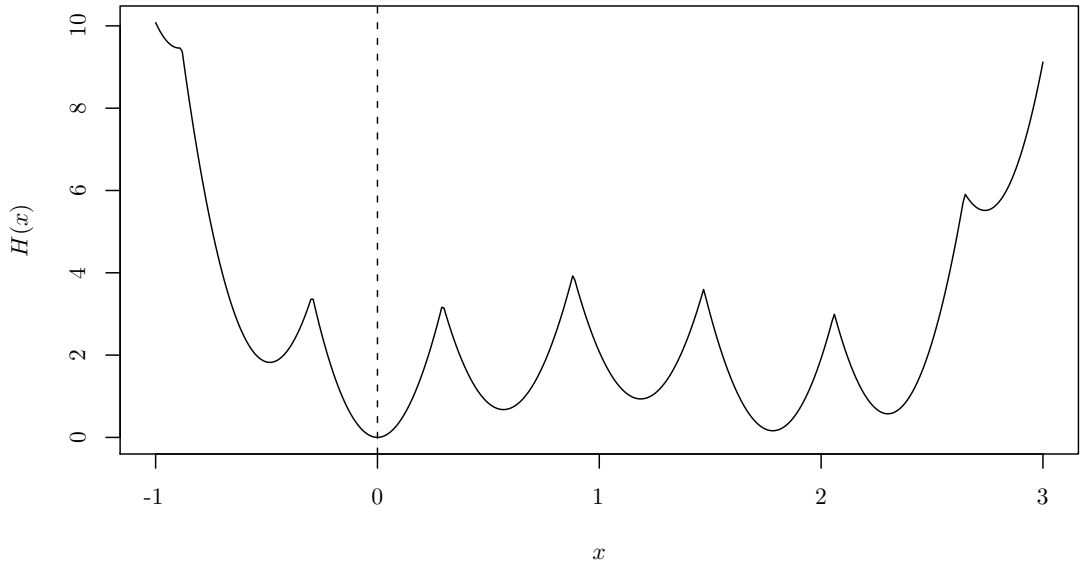
Both can fit our simple optimisation framework, and we can see a further illustration of the workings of the annealing method by considering the sequence of distributions obtained for simple problems.

Example 9.4 (Gaussian MAP Estimation). – If $l(\mu; \mathbf{x}) = \prod_{i=1}^{n} \phi_{\mu, \sigma^2}(x_i)$ with σ^2 known.
– And $\pi(\mu) = \phi_{\mu_0, \sigma_0^2}(\mu)$ then
– The posterior is

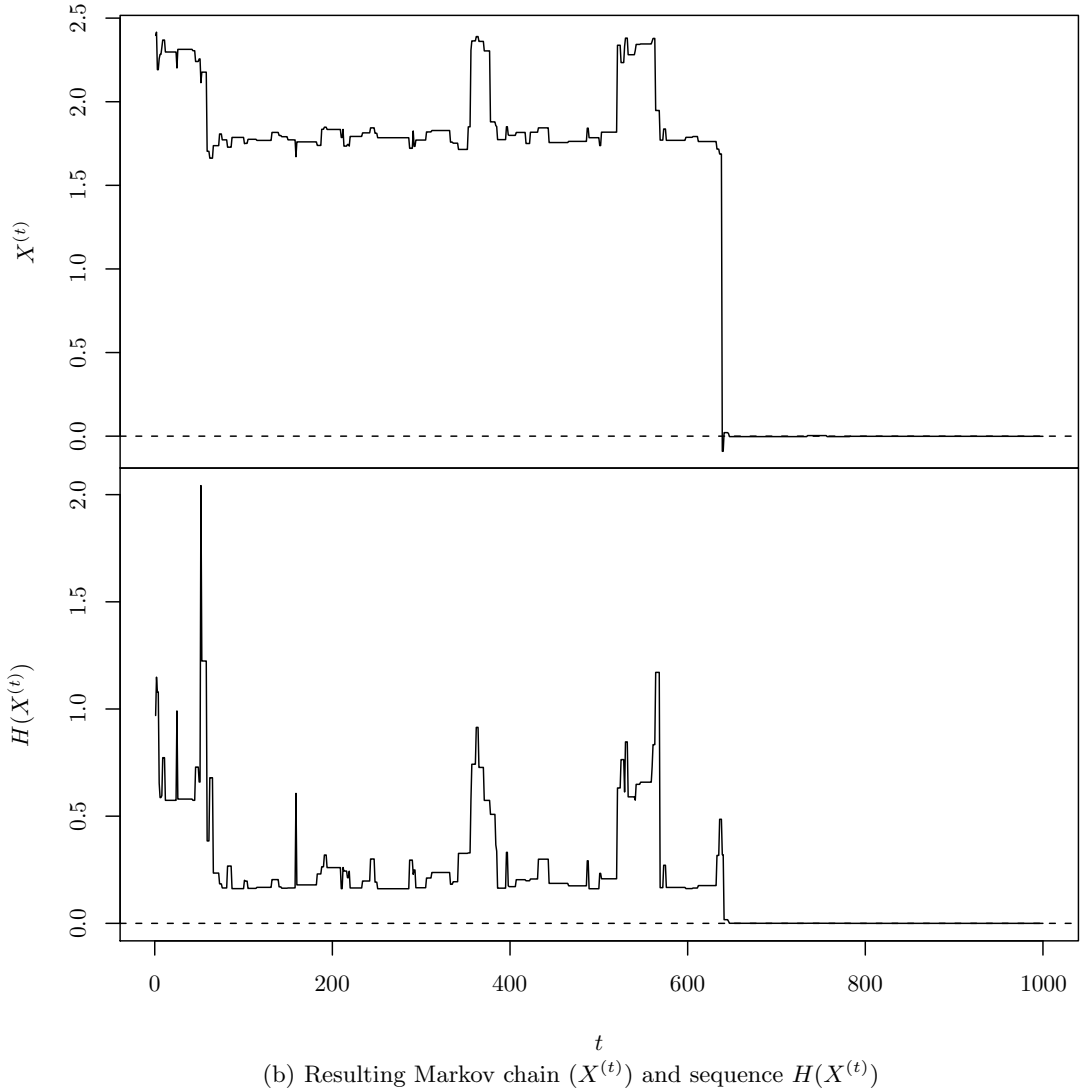
$$f^{\text{post}}(\mu) = \mathsf{N}\left(\mu; \frac{\sigma^2 \mu + n\sigma_0^2 \bar{x}}{\sigma^2 + n\sigma_0^2}, \frac{\sigma^2 \sigma_0^2}{\sigma^2 + n\sigma_0^2}\right)$$

– And we could aim to sample from

$$f_{(\beta)}^{\text{MAP}}(\mu) \propto (f^{\text{post}}(\mu))^{\beta} \propto \mathsf{N}\left(\mu; \frac{\sigma^2 \mu + n\sigma_0^2 \bar{x}}{\sigma^2 + n\sigma_0^2}, \frac{\sigma^2 \sigma_0^2}{\beta(\sigma^2 + n\sigma_0^2)}\right)$$



(a) Objective function



(b) Resulting Markov chain $(X^{(t)})$ and sequence $H(X^{(t)})$

Figure 9.4. Objective function $H(x)$ from Example 9.3 and first 1,000 iterations of the Markov chain yielded by simulated annealing.

Example 9.5 (Example: Normal ML Estimation). – If $l(\mu;\mathbf{x}) = \prod_{i=1}^{n} \phi_{\mu,\sigma^2}(x_i)$ with σ^2 known.
– We could view the likelihood as being proportional to a distribution over μ:

$$f(\mu) = \mathsf{N}\left(\mu; \bar{x}, \sigma^2/n\right).$$

– And we could aim to sample from

$$f_{(\beta)}^{\mathsf{MLE}}(\mu) \propto (f(\mu))^{\beta} \propto \mathsf{N}\left(\mu; \bar{x}, \sigma^2/\beta n\right).$$

In both of these cases, the sequence of distributions concentrates on the maximiser of the original objective function and so any algorithm able to sample from these distributions (for large enough β) will provide good approximations of the optimiser of the objective function.

Two closely related problems often arise when dealing with complicated statistical models. *Marginal Maximum Likelihood Estimation:* Given $l(\theta;\mathbf{x}) = \int f_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z};\theta)d\mathbf{z}$ compute $\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta\in\Theta} l(\theta;\mathbf{x})$ and *Marginal Maximum a Posteriori Estimation:* Given $l(\theta;\mathbf{x}) = \int f_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z};\theta)d\mathbf{z}$ and prior $f^{\mathrm{prior}}(\theta)$ compute $\hat{\theta}_{\mathrm{MMAP}} = \arg\max_{\theta\in\Theta} f^{\mathrm{prior}}(\theta)l(\theta;\mathbf{x})$. Such problems often arise when one can write down a complete generative model for the process by which the data arose in terms of the parameters, but one only observes a subset of the random quantities generated within that model. For example, consider a mixture model in which we don't observe the association of observations with mixture components or a genetic model in which we observe the DNA of only the current generation of individuals: we don't observe the DNA of their ancestors or their *family trees*. If it's possible to integrate out the unobserved random quantities then we can proceed as usual, but unfortunately, we can't typically evaluate the marginal likelihoods.

Recall the *demarginalisation* technique for sampling from $f_{\mathbf{X}}(\mathbf{x})$ by defining a convenient joint distribution $f_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z})$ which admits the distribution of interest as a marginal. In order to do this, we saw that we could introduce a set of auxiliary random variables $Z_1,\ldots,Z_r$ such that $f_{\mathbf{X}}$ is the marginal density of $(X_1,\ldots,X_p)$ under the joint distribution of $(X_1,\ldots,X_p,Z_1,\ldots,Z_r)$, i.e.

$$f(x_1,\ldots,x_p) = \int f(x_1,\ldots,x_n,z_1,\ldots,z_r)\, d(z_1,\ldots,z_r).$$

The idea of introducing some auxiliary random variables in such a way that $f_{(\beta)}(\mathbf{x})$ is the marginal distribution seems a natural extension of this idea.

In order to do this, we consider

$$l(\mathbf{x},\mathbf{z}|\theta) = f_{X,Z}(\mathbf{x},\mathbf{z}|\theta) = f_Z(\mathbf{z}|\theta)f_X(\mathbf{x}|\mathbf{z},\theta).$$

and introduce a whole collection of *vectors* of auxiliary variables:

$$f_{\beta}^{MMAP}(\theta,\mathbf{z}_1,\ldots,\mathbf{z}_\beta|\mathbf{x}) \propto \prod_{i=1}^{\beta}\left[\pi(\theta)f_Z(\mathbf{z}_i)f_X(\mathbf{x}|\mathbf{z}_i,\theta)\right]$$

and we can easily establish that, by exploiting the conditional independence structure of our augmented likelihood:

$$\begin{aligned} f_{\beta}^{MMAP}(\theta|\mathbf{x}) &\propto \int f_{\beta}^{MMAP}(\theta,\mathbf{z}_1,\ldots,\mathbf{z}_\beta|\mathbf{x})d\mathbf{z}_1,\ldots d\mathbf{z}_\beta \\ &\propto \pi(\theta)^{\beta} f_X(\mathbf{x}|\theta)^{\beta} = f^{\mathrm{post}}(\theta|\mathbf{x})^{\beta} \end{aligned}$$

This idea is the basis of the *State Augmentation for Maximisation of Expectations* (SAME) algorithm (Doucet et al., 2002b).

In the case of maximising the likelihood rather than the posterior we need to be slightly more careful. The likelihood is a probability density over the data, but need not even be integrable if viewed as a function of the parameters. We can address this problem by introducing an *instrumental* prior distribution (one used exclusively for computational reasons which is not intended to have any influence on the resulting inference.

Considering

$$l(\theta; \mathbf{x}, \mathbf{z}) = f_{X,Z}(\mathbf{x}, \mathbf{z}|\theta) = f_Z(\mathbf{z}|\theta) f_X(\mathbf{x}|\mathbf{z}, \theta),$$

we can again consider multiple augmentation — this time for for MMLE estimation — by setting

$$f_\beta^{MMLE}(\theta, \mathbf{z}_1, \ldots, \mathbf{z}_\beta|\mathbf{x}) \propto \pi(\theta) \prod_{i=1}^{\beta} [f_Z(\mathbf{z}_i) f_X(\mathbf{x}|\mathbf{z}_i, \theta)]$$

which ensures that

$$\begin{aligned} f_\beta^{MMLE}(\theta|\mathbf{x}) &\propto \int f_\beta^{MMLE}(\theta, \mathbf{z}_1, \ldots, \mathbf{z}_\beta|\mathbf{x}) d\mathbf{z}_1, \ldots d\mathbf{z}_\beta \\ &\propto \left[\pi(\theta)^{(1/\beta)} f_X(\mathbf{x}|\theta)\right]^\beta \approx l(\theta; \mathbf{x})^\beta \end{aligned}$$

for large enough β under support and regularity conditions on $\pi(\cdot)$, the *instrumental* prior.

Both of these augmentation strategies can give rise to a sequence of target distributions if we replace β with β_t, a non-decreasing sequence of numbers of replicates of the augmenting variables (in the SAME case it can be sensible to keep β_t fixed at a particular value for several iterations to give the chain time to reach equilibrium before further increasing it). And given such a sequence of target distributions we can apply MCMC kernels for which each is invariant in essentially the same manner as we did when considering simulated annealing. In the particular case in which we can sample from all of the relevant full conditional distributions, this gives rise to Algorithm 9.2, more general cases can be dealt with via obvious extensions.

Algorithm 9.2 (The SAME Gibbs Sampler). Starting with $\boldsymbol{\theta}^{(0)}$ iterate for $t = 1, 2, \ldots$

1. Increase $\beta^{(t-1)}$ to $\beta^{(t)}$ (if necessary).
2. For $k = 1, \ldots, \beta_t$, sample:
$$\mathbf{z}_k^{(t)} \sim f_Z(\mathbf{z}_k^{(t)}|x, \boldsymbol{\theta}^{(t-1)})$$
3. Sample:
$$\boldsymbol{\theta}^{(t)} \sim f_{(\beta_t)}^{\ldots}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{z}_1^{(t)}, \ldots, \mathbf{z}_{\beta_t}^{(t)})$$

The following toy example shows the SAME Gibbs sampler in action.

Example 9.6. Consider finding the parameters which maximise the likelihood in a setting in which the likelihood is a student t-distribution of unknown location parameter θ with 0.05 degrees of freedom. Four observations are available, $\mathbf{x} = (-20, 1, 2, 3)$.

In this case, the marginal likelihood is known (and we can use this knowledge to verify that the algorithm works as expected):

$$\log p(\mathbf{x}|\theta) = -0.525 \sum_{i=1}^{4} \log\left(0.05 + (x_i - \theta)^2\right).$$

This marginal likelihood is illustrated in Figure 9.5.

However, it is also possible to write down an augmented complete likelihood which admits this as a marginal distribution by exploiting the fact that the student t-distribution may be written as a *scale mixture* of normal densities:

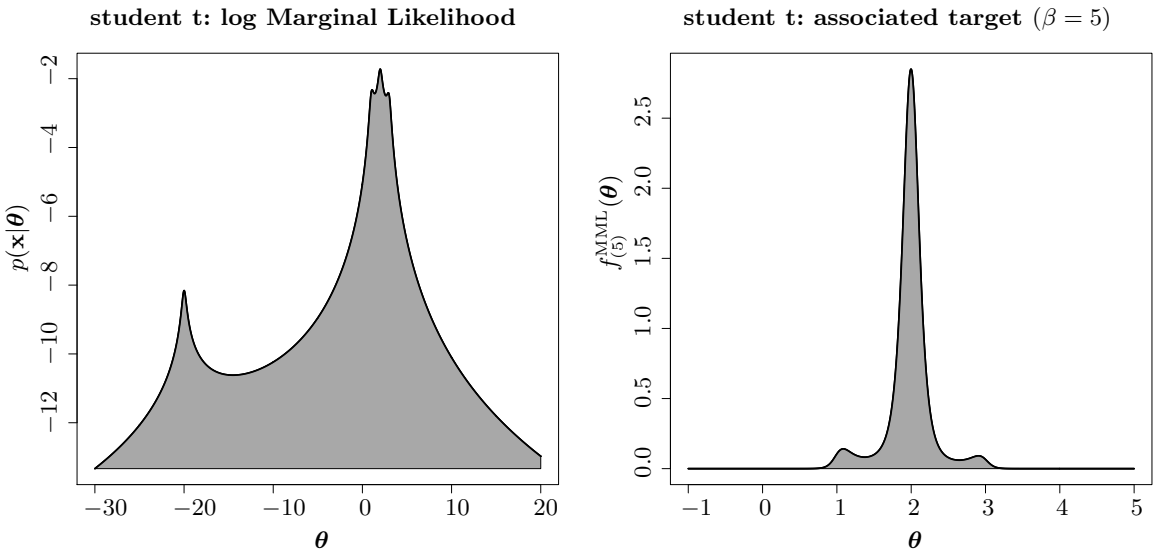


Figure 9.5. The log marginal likelihood (left) and the target distribution obtained by the annealing approach at $\beta = 5$ (right) for Example 9.6.

$$\log p(\mathbf{x}, \mathbf{z}|\theta) = -\sum_{i=1}^{4} \left[0.475 \log z_i + 0.025 z_i + 0.5 z_i (x_i - \theta)^2\right]$$

$$p_{(\beta_t)}(\mathbf{z}_{1:\beta_t}|\theta, \mathbf{x}) = \prod_{i=1}^{\beta_t} \prod_{j=1}^{4} \mathsf{Gamma}\left(z_{i,j} \left| 0.525, 0.025 + \frac{(x_j - \theta)^2}{2}\right.\right),$$
$$p_{(\beta_t)}(\theta|\mathbf{z}_{1:\beta_t}) \propto \mathcal{N}\left(\theta \left| \mu_t^{(\theta)}, \Sigma_t^{(\theta)}\right.\right),$$

where the parameters,

$$\Sigma_t^{(\theta)} = \left[\sum_{i=1}^{\beta_t} \sum_{j=1}^{4} z_{i,j}\right]^{-1} \qquad \mu_t^{(\theta)} = \Sigma_t^{(\theta)} \sum_{i=1}^{\beta_t} y^T z_i$$

We can straightforwardly implement the SAME Gibbs sampler for this problem.

It's perhaps more interesting to return to the familiar mixture model for which we have already considered several forms of inference and to apply the data augmentation approach to the problem of maximising the posterior density (note that one cannot use maximum likelihood estimation, at least directly, in this setting as the likelihood is not bounded above: if a cluster mean coincides exactly with an observation then making the variance of that component arbitrarily small leads to an arbitrarily high likelihood).

Example 9.7 (MAP Estimation for a Gaussian Mixture Model). Returning to the configuration considered in Example 8.3 but treating the number of components as known, we have:

– n iid observations, $x_1, \ldots, x_n$.
– Likelihood $f_{X,Z}(x_i, z_i|\omega, \mu, \sigma) = \omega_{z_i} \mathsf{N}(x_i|\mu_{z_i}, \sigma_{z_i}^2)$.
– Marginal likelihood $f_X(x_i|\omega, \mu, \sigma) = \sum_{j=1}^{K} \omega_j \mathsf{N}(x_i|\mu_j, \sigma_j^2)$.
– Diffuse conjugate priors were employed:

$$\omega \sim \text{Dirichlet}(\chi, \ldots, \chi)$$
$$\sigma_i^2 \sim \text{IG}\left(\frac{\lambda_i + 3}{2}, \frac{b_i}{2}\right)$$
$$\mu_i | \sigma_i^2 \sim \text{N}(a_i, \sigma_i^2/\lambda_i)$$

All full conditional distributions of interest are available, which allows us to use our Gibbs sampling strategy. This gives rise to an iterative algorithm in which step t comprises the following steps:

– Sample:

$$\omega \leftarrow \text{Dirichlet}\left(\beta_t(\chi - 1) + 1 + n_1^{\beta_t}, \ldots, \beta_t(\chi - 1) + 1 + n_K^{\beta_t}\right)$$
$$\sigma_i^2 \leftarrow \text{IG}(A_i, B_i)$$
$$\mu_i | \sigma_i^2 \leftarrow \text{Normal}\left(\frac{\beta_t \lambda_i a_i + \bar{\mathbf{x}}_i^{\beta_t}}{\beta_t \lambda_i + n_i^{\beta_t}}, \frac{\sigma_i^2}{\beta_t \lambda_i + n_i^{\beta_t}}\right)$$

where

$$n_i^{\beta_t} = \sum_{l=1}^{\beta_t} \sum_{p=1}^{n} \mathbb{I}_i(Z_{l,p}^{(t-1)}) \qquad \bar{\mathbf{x}}_i^{\beta_t} = \sum_{l=1}^{\beta_t} \sum_{p=1}^{n} \mathbb{I}_i(Z_{l,p}^{(t-1)}) x_p$$
$$\overline{\mathbf{x}^2}_i^{\beta_t} = \sum_{l=1}^{\beta_t} \sum_{p=1}^{n} \mathbb{I}_i(Z_{l,p}) x_p^2$$

– and

$$A_i = \frac{\beta_t(\lambda_i + 1) + n_i^{\beta_t}}{2} + 1$$
$$B_i = \frac{1}{2}\left(\beta_t(b_i + \lambda_i a_i^2) + \bar{\mathbf{x}^2}_i^{\beta_t} - \sum_{g=1}^{\beta_t} \frac{(\bar{\mathbf{x}}_i^g - \bar{\mathbf{x}}_i^{g-1} + \lambda_i a_i)^2}{\lambda_i + n_i^g - n_i^{g-1}}\right)$$

– Sample, for $j = 1, \ldots, \beta_t$:

$$\mathbf{z}_j^{(t)} \sim f^{\text{posterior}}(\mathbf{z}|\mathbf{x}, \pi^{(t)}, \sigma^{(t)}, \mu^{(t)})$$

Marginal posterior can be calculated (which means that we don't *need* such a complicated algorithm to deal with this problem, although it does perform well; the advantage of using such an example is that it allows us to assess the performance of the algorithm).

First we compare the performance of 50 runs of the algorithm with 50 (differently initialised) runs of a deterministic algorithm (expectation maximisation; EM) which is widely used to deal with problems of this type. *Cost* gives a rough indication of the computational cost of running each algorithm once.

Algorithm	T	**Cost**	**Mean**	**Std. Dev.**	**Min**	**Max**
EM	500	500	-158.06	3.23	-166.39	-153.85
EM	5000	5000	-157.73	3.83	-165.81	-153.83
SAME(6)	4250	8755	-155.32	0.87	-157.35	-154.03
SAME(50)	4250	112522	-155.05	0.82	-156.11	-153.98

Where two different sequences of the annealing parameter were considered:

SAME(6) set $\beta_t = 1$ for the first half of the iterations and then increasing linearly to a final maximum value of 6.

SAME(50) set $\beta_t = 1$ for the first 250 iterations, and then increasing linearly to 50

The log posterior density of the generating parameters was -155.87. These parameters were:

$$\pi = [0.2, 0.3, 0.5] \qquad \mu = [0, 2, 3] \qquad \text{and } \sigma = \left[1, \frac{1}{4}, \frac{1}{16}\right].$$

Although the EM algorithm occasionally produces good results, for this clean simulate data, some runs of the algorithm totally fail to find anything close to the global mode. The SAME algorithm is computationally more costly, but does behave more robustly. In real marginal optimisation problems, one typically cannot evaluate the objective function and so robust methods which can be relied upon to produce good solutions are required.

Next we turn ourselves to the much celebrated *Galaxy* data set of of (Roeder, 1990). This data set consists of the velocities of 82 galaxies, and it has been suggested that it consists of a mixture of between 3 and 7 distinct components – for example, see (Roeder and Wasserman, 1997) and (Escobar and West, 1995). For our purposes we have estimated the parameters of a 3 component Gaussian mixture model from which we assume the data was drawn. The following table summarises the marginal posterior of the solutions found by 50 runs of each algorithm, comparing the same algorithm with the EM algorithm. *Cost* gives a rough indication of the computational cost of running each algorithm once.

Algorithm	T	**Cost**	**Mean**	**Std. Dev.**	**Min**	**Max**
EM	500	500	-46.54	2.92	-54.12	-44.32
EM	5000	5000	-46.91	3.00	-56.68	-44.34
SAME(6)	4250	8755	-45.18	0.54	-46.61	-44.17
SAME(50)	4250	112522	-44.93	0.21	-45.52	-44.47

Again, two different sequences of annealing schedule were considered:

SAME(6) set $\beta_t = 1$ for the first half of the iterations and then increasing linearly to a final maximum value of 6.

SAME(50) set $\beta_t = 1$ for the first 250 iterations, and then increasing linearly to 50

A more sophisticated algorithm (Johansen et al., 2008) suggests that -43.96 ± 0.03 is about optimal.

and again, good robustness is demonstrated by the same algorithm.

10. Simulated Tempering and Related Methods

This chapter explores the use of annealing-like strategies within a more standard Monte Carlo context, attempting to create modified target distributions for which it is easier to construct fast-mixing Markov chains and to use these to approximate distributions of interest.

10.1 The Basis of Tempering

As we have seen in previous chapters, Markov chains sampling form distributions whose components are separated by areas of low probability typically have very poor mixing properties. We can use strategies very much like the ones used in the simulated annealing algorithm to bridge these "barriers" of low probability. The basic idea is to consider distributions with densities $f_{(\beta)}(x) \propto (f(x))^{(\beta)}$ for *small* β $(0 < \beta < 1)$, as opposed to large β as with simulated annealing. Choosing $\beta < 1$ makes the distribution more "spread out" and thus makes it easier to move from one part of the domain to another.

Whilst it might be easier to sample from $f_{(\beta)}(\cdot)$ for $\beta < 1$, we are actually not interested in this sample, but in a sample from $f(\cdot)$ itself (i.e. $f_{(\beta)}$ for $\beta = 1$). In order to address this difficulty, the strategies which are considered in subsequent sections consider extended target distributions which admit the distribution of interest as a marginal (yet more data augmentation) but which benefit from the better mixing properties of Markov chains targetting $f_{(\beta)}$ for $\beta \ll 1$. The idea was proposed by Marinari and Parisi (1992). Chapter 10 of (Liu, 2001) gives an overview over such approaches.

Before moving on to these more sophisticated approaches, it's useful to explore the behaviour of a Markov chain which targets $f_{(\beta)}$ for a range of values of $\beta \in (0, 1)$ and to consider using these samples within a simple importance sampling framework to approximate the distribution of interest.

In order to do this, we need to consider how to combine importance sampling with MCMC. Treating $f_{(\beta)}(x)$ as an importance distribution, if we could sample $X^{(1)}, \ldots, X^{(T)} \sim f_{(\beta)}$ then we could use

$$\int \varphi(x) f(x) dx \approx \frac{1}{T} \sum_{t=1}^{T} \varphi(X^{(t)}) \frac{f(x^{(t)})}{f_{(\beta)}(x^{(t)})}$$

or more realistically

$$\int \varphi(x) f(x) dx \approx \sum_{t=1}^{T} \varphi(X^{(t)}) \frac{f(X^{(t)})}{f_{(\beta)}(X^{(t)})} \Big/ \sum_{t=1}^{T} \frac{f(X^{(t)})}{f_{(\beta)}(X^{(t)})} \tag{10.1}$$

to estimate the expectation of φ under f. It's natural to ask what we might do if $X^{(1)}, \ldots, X^{(T)}$ come from an MCMC algorithm for which $f_{(\beta)}$ is the stationary distribution.

Actually, the justification of self-normalised importance sampling didn't depend upon the independence of the samples used. The ratio of two consistent estimators is itself a consistent estimator (of the ratio of the two quantities being estimated) under very weak conditions and the numerator and denominator of the right hand side of (10.1) are consistent estimators of

$$\mathbb{E}_{f_{(\beta)}}\left[\varphi(X)\frac{f(X)}{f_{(\beta)}(X)}\right] = \mathbb{E}_f[\varphi(X)]$$

and

$$\mathbb{E}_{f_{(\beta)}}\left[\frac{f(X)}{f_{(\beta)}(X)}\right] = \mathbb{E}_f[\mathbf{1}] = 1,$$

respectively, whenever the MCMC algorithm satisfies an ergodic theorem. As usual, multiplying the numerator and denominator by a common constant leaves their ratio unchanged and so we do not require access to the normalising constant of f or $f_{(\beta)}$.

Example 10.1 (Tempering and the Bimodal Mixture of Normals). We revisit Example 6.1 and consider approximating a mixture of two well-separated Gaussians

$$f(x) = 0.4 \cdot \phi_{(-1,0.2^2)}(x) + 0.6 \cdot \phi_{(2,0.3^2)}(x)$$

(see Figure 6.2 (a) for a plot of the density) using a random walk Metropolis algorithm with an $\mathrm{N}(0, 0.4^2)$ increment distribution. Figure 10.1 shows the behaviour of such a chain targetting $f(x) = f_{(1)}(x)$ and also of a chain targetting $f_{(0.06)}(x)$. The difference in their ability to explore the support of f is immediately striking.

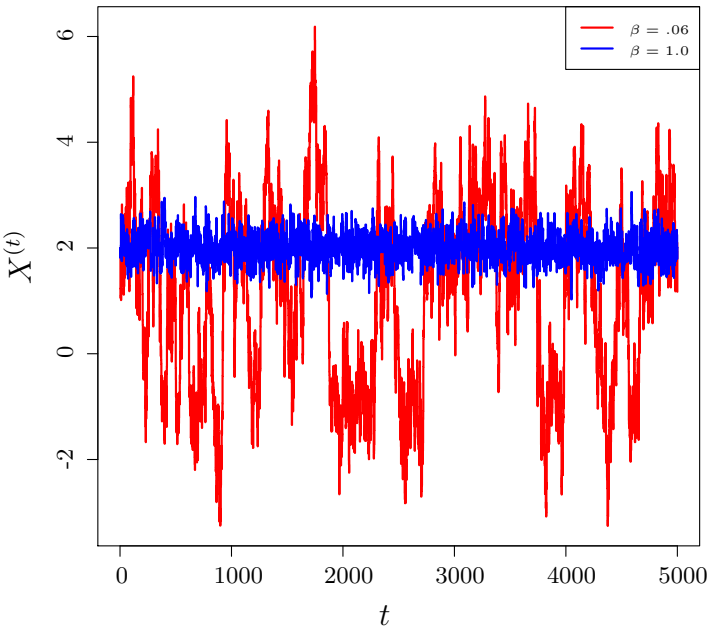


Figure 10.1. Two $f_{(\beta)}$-invariant Markov chains for the bimodal normal target density.

In order to assess the importance sampling strategy, we consider 50 replicates of the estimate obtained using Markov chains targetting $f_{(\beta)}$ for several different values of β with the appropriate importance sampling correction. Figure 10.2 illustrates the performance of this simple importance sampling estimator.

It's clear that tempering leads to more reliable estimators than the direct estimate and that it's reasonable robust to the precise choice of β. ◁

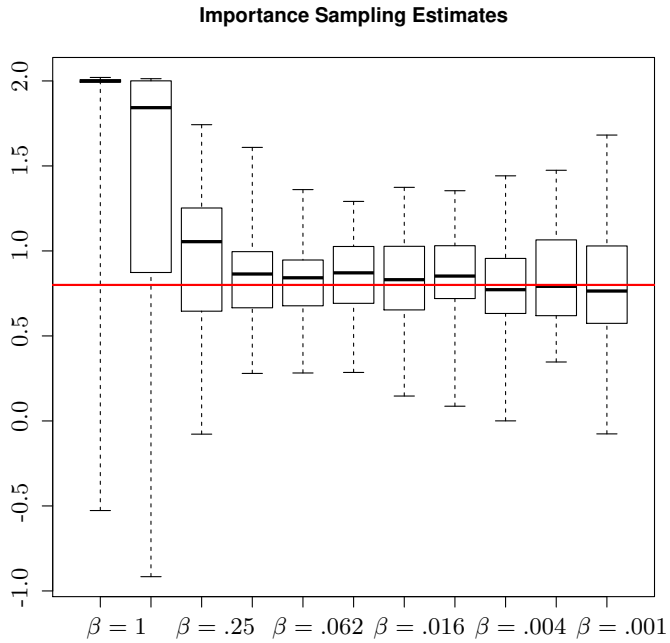


Figure 10.2. Performance of importance sampling estimator for the bimodal Gaussian mixture. For each value of β results for 50 independent runs are summarised here. In each case a chain of length $5,000$ was used.

The success of the simple importance sampling strategy in Example 10.1 is encouraging but we might worry that for more complex problems the variance of an importance sampling estimator using $f_{(\beta)}$ as a proposal might be larger for any value of β small enough to allow good mixing of our Markov chain. Indeed, for high dimensional problems, small values of β will lead to distributions $f_{(\beta)}$ for which almost all of the mass lies between the modes of the original distribution. We consequently turn our attention to slightly more sophisticated strategies.

10.2 Simulated Tempering

The first thing we might consider is simply augmenting the space and including a temperature parameter in the state space allowing us to construct a Markov chain which moves around in both temperature and state space, that is to construct an MCMC algorithm which targets $g(\beta)f_{(\beta)}(\mathbf{x})$ for some distribution g.

If we do this using a Metropolis-Hastings algorithm (and it's not clear how else we might construct such an algorithm in any generality) the the acceptance probability if we started at $(\beta^{(t-1)}, \mathbf{X}^{(t-1)})$ and sampled $(\beta, \mathbf{X})$ from $q(\beta, \mathbf{X}|\beta^{(t-1)}, \mathbf{X}^{(t-1)})$ is:

$$\min\left(1, \frac{g(\beta)f_{(\beta)}(\mathbf{X})}{g(\beta^{(t-1)})f_{(\beta^{(t-1)})}(\mathbf{X}^{(t-1)})} \frac{q(\beta^{(t-1)}, \mathbf{X}^{(t-1)}|\beta, \mathbf{X})}{q(\beta, \mathbf{X}|\beta^{(t-1)}, \mathbf{X}^{(t-1)})}\right)$$

and the normalizing constants *do not* cancel (we compare $f_{(\beta)}(\mathbf{X})$ and $f_{(\beta^{(t-1)})}(\mathbf{X}^{(t-1)})$ and these two distributions have *different* normalising constants).

In order to avoid this difficulty, the *simulated tempering* algorithm introduces a discrete sequence of n different inverse temperatures, $\beta_0 = 1 > \beta_1 > \ldots > \beta_n$ and considers the slightly different augmented probability distribution:

$$\bar{f}(\beta, \mathbf{x}) \propto \sum_{i=0}^{n} c_i \mathbb{I}_{\{\beta_i\}}(\beta) f^{\beta}(\mathbf{x}) \tag{10.2}$$

for some sequence of positive constants c_i. This distribution has a single unknown normalising constant which *will* cancel if a Metropolis-Hastings algorithm is applied but this comes at the cost of having to specify the sequence $c_0, \ldots, c_n$ which will have a significant influence upon the mixing of the resulting chain.

Assuming for the moment that we can specify some reason sequence $\{c_i\}$ we arrive at Algorithm 10.1 if we assume that during each iteration we choose one of two moves at random and update either the current value of the state variables, $\mathbf{X}$, *or* the current inverse temperature, β (which is identify by a variable I such that the current inverse temperature is $\beta = \beta_I$).

Algorithm 10.1 (Simulated Tempering). Starting with $(\mathbf{X}^{(0)}, I^{(0)})$, iterate for $t = 1, \ldots, T$:

1. With probability p:
   - Propose $I \sim \mathsf{Unif}\{I^{(t-1)} - 1, I^{(t-1)} + 1\}$.
   - Set $(\mathbf{X}^{(t)}, I^{(t)}) = (\mathbf{X}^{(t-1)}, I)$ with probability:

$$\alpha_I = \min\left(1, \frac{c_I}{c_{I^{(t-1)}}} \frac{f^{\beta_I}(X^{(t-1)})}{f^{\beta_{I^{(t-1)}}}(X^{(t-1)})}\right)$$

   - Otherwise set $(\mathbf{X}^{(t)}, I^{(t)}) = (\mathbf{X}^{(t-1)}, I^{(t-1)})$.
2. Otherwise:
   - Propose $\mathbf{X} \sim q_{I^{(t-1)}}(\mathbf{X}|\mathbf{X}^{(t-1)})$
   - Set $(\mathbf{X}^{(t)}, I^{(t)}) = (\mathbf{X}, I^{(t-1)})$ with probability:

$$\alpha_I = \min\left(1, \frac{f^{\beta_{I^{(t-1)}}}(\mathbf{X})}{f^{\beta_{I^{(t-1)}}}(\mathbf{X}^{(t-1)})} \frac{q_{I^{(t-1)}}(\mathbf{X}^{(t-1)}|\mathbf{X})}{q_{I^{(t-1)}}(\mathbf{X}|\mathbf{X}^{(t-1)})}\right)$$

   - Otherwise set $(\mathbf{X}^{(t)}, I^{(t)}) = (\mathbf{X}^{(t-1)}, I^{(t-1)})$.

Both of the moves employed within the simulated tempering algorithm are simple Metropolis-Hastings steps which update a subset of the variables, with the proposed moves being accepted with the usual Metropolis-Hastings acceptance probability using (10.2) as the target distribution.

A natural objective when designing a simulated tempering algorithm is that movement between different rungs of the *temperature ladder*, as the sequence of values of β which are use is often called, are not inhibited by the choice of sequence c_i and the usual way to try to achieve this is to make the marginal distribution over I uniform over $\{0, \ldots, \beta_n\}$.

In order to obtain a uniform distribution over this set, we need that (for normalising constant Z):

$$\int \bar{f}(\beta_j, \mathbf{x}) = \frac{1}{Z} \int \sum_{i=0}^{n} c_i \mathbb{I}_{\{\beta_i\}}(\beta) f^{\beta}(\mathbf{x}) d\mathbf{x} = \frac{1}{Z} \int c_j f^{\beta_j}(\mathbf{x}) d\mathbf{x}$$

is equal to $1/(n+1)$ for $j \in \{0, \ldots, n\}$. It's clear that this is equivalent to choosing $c_j^{-1} = \int f^{\beta_j}(\mathbf{x}) d\mathbf{x}$, which is typically infeasible as this amounts to knowing the normalising constants of a whole family of distributions related to f.

In practice, good performance can be obtained for any sequence of c_i which doesn't deviate from this "optimal" sequence by too much.

Example 10.2 (Tempering and the Bimodal Mixture of Normals (2)). Returning to the problem considered in Example 10.1, we now consider a simulated tempering strategy.

Applying Algorithm 10.1 to this problem with:

– Target Density:

$$f(x) = 0.4 \cdot \phi_{(-1,0.2^2)}(x) + 0.6 \cdot \phi_{(2,0.3^2)}(x)$$

– Let $n = 10$ and $\beta_i = 2^{-i}$.

– Let $c_i = Z_i$ where $Z_i = \int f^{(\beta_i)}(x)dx$.

– Let $q_i(x|x^{(t-1)}) = \mathsf{N}(x; x^{(t-1)}, \sigma^2)$ with $\sigma^2 = 0.4^2$.

Leads to a chain which explores the support of $f(x)$ well, as illustrated in Figure 10.3.

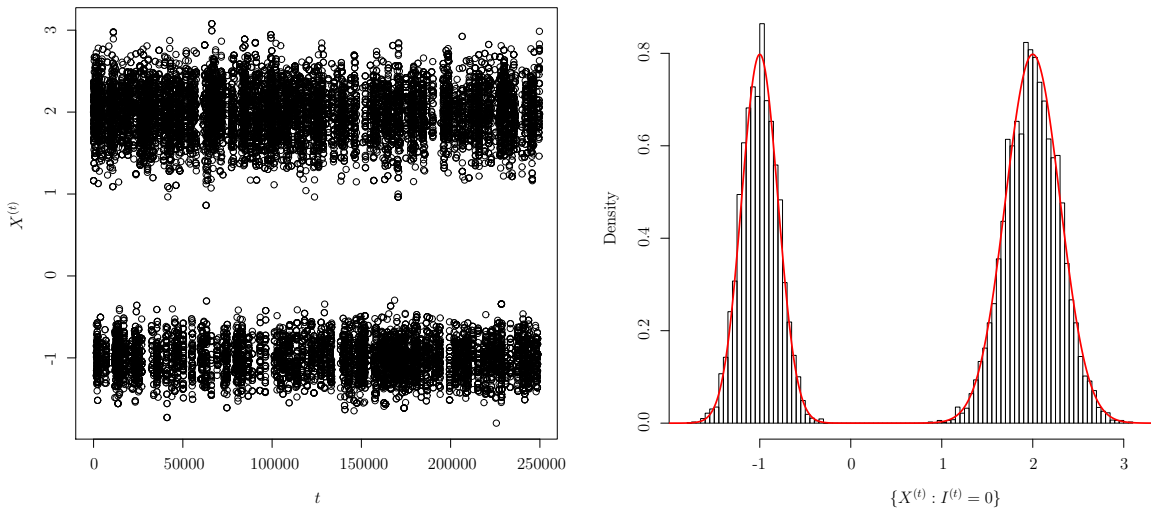


Figure 10.3. Behaviour of a simulated tempering chain targetting a simple bimodal mixture, considering only those time points, t, for which $I^{(t)} = 0$. Left panel: trace; right panel: histogram

.

Notice that the chain switches regularly between the two modes and the resulting histogram shows a good approximation of f; in contrast, we have seen that a simple Random Walk Metropolis algorithm targetting f with the same proposal kernel will become trapped within a single mode and fail to provide a good approximation of the target distribution. ◁

10.3 Parallel Tempering

An alternative strategy which partially sidesteps the need to specify a sequence c_i is employed within the *parallel tempering algorithm*. Again, a sequence of $n + 1$ values of β, with $\beta_0 = 1 > \beta_1 > \ldots > \beta_n > 0$, are employed, just as in the simulated tempering setting, but a different augmented target distribution is employed.

Rather than introducing an additional variable corresponding to inverse temperature, the parallel tempering strategy involves introducing an additional n copies of the entire state vector, and employs a target distribution in which each of these variables is independently distributed according to $f_{(\beta)}$ for a different value of β. That is, the augmented target distribution is:

$$\bar{f}(\mathbf{x}_0, \ldots, \mathbf{x}_n) \propto \prod_{i=0}^{n} f_{(\beta_i)}(\mathbf{x}_i)$$

Again two sorts of MCMC move are employed, in this case standard fixed-temperature moves and *exchange moves* which swap states between temperatures. The motivating idea is that the fixed temperature moves allow each of the individual copies of the state vector to explore the space, targetting different target distributions with those associated with small values of β able to mix freely while the exchange moves will allow the faster mixing of these changes to be transferred to the chain of interest. The chains targetting heavily tempered versions of the target can move freely around the space and whenever they visit regions of high density under the target distribution itself, any exchange move between the target and the tempered chains is likely to be accepted. The use of more than one tempered chain is necessary to ensure a reasonable proportion of exchange moves are accepted (typically, we consider exchange moves only between distributions in adjacent positions in the temperature ladder).

The specification of the target distribution in the parallel tempering case is such that unknown normalising constants need never be evaluated. Each standard fixed temperature move uses a proposal

$$q_i^{\mathrm{S}}(\mathbf{x}_0,\ldots,\mathbf{x}_n|\mathbf{x}_0^{(t-1)},\ldots,\mathbf{x}_n^{(t-1)}) = \tilde{q}_i(\mathbf{x}_i|\mathbf{x}_{(i)}^{(t-1)})\mathbb{I}_{\mathbf{x}_{-i}^{(t-1)}}(\mathbf{x}_{-i})$$

with acceptance probability

$$\alpha_i^{\mathrm{S}}(\mathbf{x}_0,\ldots,\mathbf{x}_n|\mathbf{x}_0^{(t-1)},\ldots,\mathbf{x}_n^{(t-1)}) = \min\left(1, \frac{f_{(\beta_i)}(\mathbf{x}_i)}{f_{(\beta_i)}(\mathbf{x}_i^{(t-1)})}\frac{\tilde{q}_i(\mathbf{x}_i^{(t-1)}|\mathbf{x}_i)}{\tilde{q}_i(\mathbf{x}_i|\mathbf{x}_i^{(t-1)})}\right)$$

to update component i of the extended state. The exchange moves employ proposals of the form

$$q_i^{\mathrm{E}}(\mathbf{x}_0,\ldots,\mathbf{x}_n|\mathbf{x}_0^{(t-1)},\ldots,\mathbf{x}_n^{(t-1)}) = \prod_{j\notin\{i-1,i\}}\mathbb{I}_{\mathbf{x}_j^{(t-1)}}(\mathbf{x}_j)\mathbb{I}_{\mathbf{x}_{i-1}^{(t-1)}}(\mathbf{x}_i)\mathbb{I}_{\mathbf{x}_i^{(t-1)}}(\mathbf{x}_{i-1})$$

for a randomly selected i between 1 and n and the associated acceptance probabilities are

$$\alpha_i^{\mathrm{E}}(\mathbf{x}_0,\ldots,\mathbf{x}_n|\mathbf{x}_0^{(t-1)},\ldots,\mathbf{x}_n^{(t-1)})\min\left(1, \frac{f_{(\beta_{i-1})}(\mathbf{x}_i^{(t-1)})}{f_{(\beta_{i-1})}(\mathbf{x}_{i-1}^{(t-1)})}\frac{f_{(\beta_i)}(\mathbf{x}_{i-1}^{(t-1)})}{f_{(\beta_i)}(\mathbf{x}_i^{(t-1)})}\right).$$

Algorithm 10.2 provides a simple implementation of a parallel tempering algorithm.

Algorithm 10.2 (Parallel Tempering). Starting with $\mathbf{X}_0^{(0)},\ldots,\mathbf{X}_n^{(0)}$, iterate for $t = 1,\ldots,T$:

1. For $i = 0:n$:
   - Propose $\mathbf{X}_i \sim \tilde{q}_i(\mathbf{X}_i|\mathbf{X}_i^{(t-1)})$.
   - Set $\widetilde{\mathbf{X}}_i^{(t)} = \mathbf{X}_i$ with probability
   $$\min\left(1, \frac{f_{(\beta_i)}(\mathbf{X}_i)}{f_{(\beta_i)}(\mathbf{X}_i^{(t-1)})}\frac{\tilde{q}_i(\mathbf{X}_i^{(t-1)}|\mathbf{X}_i)}{\tilde{q}_i(\mathbf{X}_i|\mathbf{X}_i^{(t-1)})}\right)$$
   - Otherwise set $\widetilde{\mathbf{X}}_i^{(t)} = \mathbf{X}_i^{(t-1)}$
2. Sample $I \sim \mathsf{Unif}\{1,\ldots,n\}$:
   - Set $\mathbf{X}^{(t)} = (\widetilde{\mathbf{X}}_{1:I-2}^{(t)}, \widetilde{\mathbf{X}}_I^{(t)}, \widetilde{\mathbf{X}}_{I-1}^{(t)}, \widetilde{\mathbf{X}}_{I+1:n}^{(t)})$ with probability:
   $$\min\left(1, \frac{f_{(\beta_{i-1})}(\mathbf{x}_{i-1})}{f_{(\beta_{i-1})}(\mathbf{x}_{i-1}^{(t)})}\frac{f_{(\beta_i)}(\mathbf{x}_i)}{f_{(\beta_i)}(\mathbf{x}_i^{(t)})}\right)$$
   - Otherwise, set $\mathbf{X}^{(t)} = \widetilde{\mathbf{X}}^{(t)}$.

Example 10.3 (Tempering and the Bimodal Mixture of Normals (3)). Returning once again to the problem considered in Example 10.1, we now consider a parallel tempering strategy.

We apply Algorithm 10.2 to this problem:

– Target Density:

$$f(x) = 0.4 \cdot \phi_{(-1,0.2^2)}(x) + 0.6 \cdot \phi_{(2,0.3^2)}(x)$$

– Set $n = 5$ and $\beta_i = 2^{-2i}$.

– Choose $\tilde{q}_i^{\mathsf{S}}(x_i|x_i^{(t-1)}) = \mathsf{N}(x_i; x_i^{(t-1)}, \sigma^2)$ with $\sigma^2 = 0.4^2/\beta^i$.

Again, we observe good mixing, as is shown in Figure 10.4. It is just possible to notice the large number of within mode moves (largely due to standard moves) between the less frequent jumps between modes (essentially exclusively due to exchange moves).

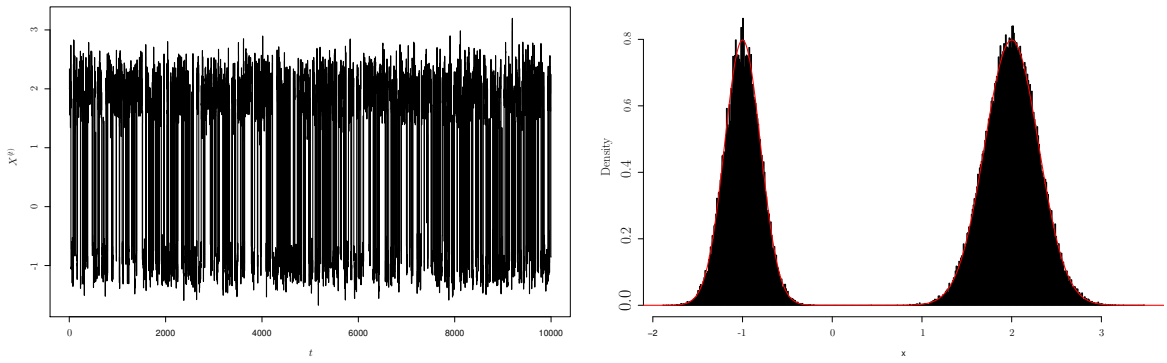


Figure 10.4. Behaviour of the first coordinate of a parallel tempering chain targetting a simple bimodal mixture. Left panel: trace; right panel: histogram

◁

Although parallel tempering frees us from the specification of one of the vectors of parameters required in simulated tempering (i.e. the $\{c_i\}$), we still need to specify the sequence of inverse temperature employed and here there are a number of important considerations. The number of distributions is important: too few and consecutive distributions will be too different and few exchange moves will be accepted; too many and it will take a long time for information to be exchanged between high and low temperature chains (as many exchange moves between adjacent chains will be required).

In addition to the number of temperatures, their actual values are also important. An optimal scaling analysis was developed by Atchadé et al. (2011) whose primary conclusion is that one should space the distributions in such a way that the average acceptance probability of exchange moves is approximately 0.234 (in this case, they employ an ESJD argument).

11. Current and Future Directions

NOT EXAMINABLE

Monte Carlo methodology is being actively developed. No module of this sort can hope to start from the beginning of the discipline and reach all of the frontiers of current research. This chapter contains a very a few words about some current research directions and attempts to provide references so that the interest reader can easily find out more. It isn't an exhaustive summary of interesting directions in this area, but I have attempted to provide a little information about at least the most widespread such topics (and, of course, those in which I am myself particularly interested).

11.1 Ensemble-based Methods and Sequential Monte Carlo

An area which I'm personally very interested in is ensemble-based methods. That is, using a collection of samples to approximate a distribution within an algorithm and performing operations on this ensemble rather than considering only a single point at a time as within MCMC algorithms.

Many of these methods come originally from the signal-processing literature in which a class of algorithms known as particle filters were introduced by Gordon et al. (1993) to approximate the solution of the discrete time optimal filtering problem using the weighted empirical distribution a collection of samples — see Doucet and Johansen (2011) for a recent survey of these and some related techniques.

Amongst others, Neal (2001) and Chopin (2001) proposed approaches based around this type of methodology (from quite different perspectives) which are applicable to more general problems. In Del Moral et al. (2006) a general framework for the implementation of this class of algorithms was presented, under the name of *Sequential Monte Carlo Samplers*. See Del Moral (2004) or Del Moral (2013) for book-length studies of the theoretical behaviour of this type of algorithm.

11.2 Pseudomarginal Methods and Particle MCMC

One area which has attracted a lot of recent attention is that in which one has access to a joint distribution but is interested in inference for only a (relatively low-dimensional) marginal of that distribution. It was demonstrated in Beaumont (2003) that with a clever spatial-extension scheme one could justify an approximation to the ideal marginal scheme.

Such an approach was further analysed by Andrieu and Roberts (2009) (and there is a trail of more recent work) who termed them *pseudomarginal* methods.

A closely related idea is the particle MCMC (PMCMC) approach of Andrieu et al. (2010). Here, SMC algorithms are used within an MCMC algorithm to integrate out large collections of latent variables. A number of schemes can be justified based upon a common extended-space view of these algorithms.

11.3 Approximate Approximate Methods

Pseudomarginal and related methods are often referred to as *exact approximate* methods: they are approximate in the sense that they emulate rather than implementing an idealised algorithm but are exact in the sense that they retain the correct invariant distribution. In recent years there has been some interest in making still further approximations and considering approximation schemes which lead to only approximations of the posterior. There is some weak theoretical support that such methods *can* work under certain circumstances (Alquier et al., 2016; Medina-Aguayo et al., 2015; Everitt et al., 2016, for example) but it remains difficult to justify their use in realistic settings.

11.4 Quasi-Monte Carlo

Quasi-random numbers, like pseudo-random numbers, are deterministic sequences of numbers which are intended to have, in an appropriate sense, similar statistical properties to pseudorandom numbers but that is the limit of the similarities between these two things. Quasi-random number sequences (QRNS) are intended to have a particular *maximum discrepancy* property. See Morokoff and Caflisch (1995) for an introduction to the Quasi Monte Carlo technique based around such numbers; or Niederreiter (1992) for a book-length introduction.

The use of quasi-random numbers within simulation procedures has received a burst of recent attention in large part due to the recent paper of Gerber and Chopin (2015) which presented an elegant approach to their incorporation within the SMC framework.

11.5 Hamiltonian/Hybrid MCMC

Hamiltonian Monte Carlo (HMC) is another approach to constructing MCMC chains which involves augmenting the state space to incorporate a *momentum* variable together with some clever numerical technology borrowed from the physics literature – like the HMC method itself – in order to produce transitions which can exhibit much better mixing than the small steps allowed by Metropolis Hastings type algorithms in high dimensions. Techniques adapted to a statistical for making use of higher order spatial structure was developed inGirolami and Calderhead (2011). See Livingstone (2015) for a good intuitive introduction and some theoretical developments or Neal (2011) for another introduction to these methods.

11.6 Methods for Big Data

An enormous research effort is currently being dedicated to the development of methods which scale sufficiently well with the size of a set of data that they allow inference with truly enormous data sets. This is too large, and too specialised, an area to dedicate much space to here, but Bardenet et al. (2015) provide an excellent comparative summary of the current state of the art.

Bibliography

Alquier, P., Friel, N., Everitt, R. and Boland, A. (2016) Noisy Monte Carlo: convergence of Markov chains with approximate transition kernels. *Statistics and Computing*, **26**, 29–47.

Andrieu, C., Doucet, A. and Holenstein, R. (2010) Particle Markov chain Monte Carlo. *Journal of the Royal Statistical Society B*, **72**, 269–342.

Andrieu, C. and Roberts, G. O. (2009) The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, **37**, 697–725.

Atchadé, Y. F., Roberts, G. O. and Rosenthal, J. S. (2011) Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. *Statistics and Computing*, **21**, 555–568.

Badger, L. (1994) Lazzarini's lucky approximation of π. *Mathematics Magazine*, **67**, 83–91.

Bardenet, R., Doucet, A. and Holmes, C. (2015) On markov chain monte carlo methods for tall data. *arXiv preprint arXiv:1505.02827*.

Barnard, G. A. (1963) Discussion of prof. bartlett's paper. **25**, 294.

Beaumont, M. (2003) Estimation of population growth or decline in genetically monitored populations. *Genetics*, **164**, 1139–1160.

Besag, J. and Diggle, P. (1977) Simple monte carlo tests for spatial pattern. **26**, 327–333.

Box, G. E. P. and Muller, M. E. (1958) A note on the generation of normal random deviates. *Annals of Mathematical Statistics*, **29**, 610–611.

Brockwell, P. J. and Davis, R. A. (1991) *Time series: theory and methods*. New York: Springer, 2 edn.

Brooks, S. and Gelman, A. (1998) General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, **7**, 434–455.

Buffon, G. (1733) Editor's note concerning a lecture given 1733 to the Royal Academy of Sciences in Paris. *Histoire de l'Académie Royale des Sciences*, 43–45.

— (1777) Essai d'arithmétique morale. *Histoire naturelle, générale et particulière*, **Supplément 4**, 46–123.

Chopin, N. (2001) Sequential inference and state number determination for discrete state-space models through particle filtering. *Working Paper 2001-34*, CREST, Laboratoire de Statistique, CREST, INSEE, Timbre J120, 75675 Paris cedex 14, France.

Del Moral, P. (2004) *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Probability and Its Applications. New York: Springer Verlag.

— (2013) *Mean Field Integration*. Chapman Hall.

Del Moral, P., Doucet, A. and Jasra, A. (2006) Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society B*, **63**, 411–436.

Doucet, A., Godsill, S. J. and Robert, C. P. (2002a) Marginal maximum a posteriori estimation using Markov chain Monte Carlo. *Statistics and Computing*, **12**, 77–84.

— (2002b) Marginal maximum a posteriori estimation using Markov chain Monte Carlo. *Statistics and Computing*, **12**, 77–84.

Doucet, A. and Johansen, A. M. (2011) A tutorial on particle filtering and smoothing: Fiteen years later. In *The Oxford Handbook of Nonlinear Filtering* (eds. D. Crisan and B. Rozovsky), 656–704.

Escobar, M. D. and West, M. (1995) Bayesian density estimation and inference using mixtures. **90**, 577–588.

Everitt, R. G., Johansen, A. M., Rowing, E. and Evdemon-Hogan, M. (2016) Bayesian model selection with un-normalised likelihoods. *Statistics and Computing*. In press.

Fahrmeir, L. and Tutz, G. (2001) *Multivariate Statistical Modelling Based on Generalised Linear Models*. New York: Springer, 2 edn.

Galassi, M. et al. (2002) *GNU Scientific Library Reference Manual*. Network Theory Ltd., 1.3 edn.

Gelfand, A. E. and Smith, A. F. M. (1990) Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, **85**, 398–409.

Gelman, A., Gilks, W. R. and Roberts, G. O. (1997) Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of. Applied Probability*, **7**, 110–120.

Gelman, A., Roberts, G. O. and Gilks, W. R. (1995) Efficient Metropolis jumping rules. In *Bayesian Statistics* (eds. J. M. Bernado, J. Berger, A. Dawid and A. Smith), vol. 5. Oxford: Oxford University Press.

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

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721–741.

Gerber, M. and Chopin, N. (2015) Sequential quasi monte carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **77**, 509–579.

Geweke, J. (1989) Bayesian inference in econometrics models using Monte Carlo integration. *Econometrica*, **57**, 1317–1339.

Gikhman, I. I. and Skorokhod, A. V. (1996) *Introduction to the Theory of Random Processes*. 31 East 2nd Street, Mineola, NY, USA: Dover.

Gilks, W. R., Richardson, S. and Spieghalter, D. J. (eds.) (1996) *Markov Chain Monte Carlo In Practice*. Chapman and Hall, first edn.

Girolami, M. and Calderhead, B. (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society B*, **73**, 123–214.

Gong, L. and Flegal, J. M. (2016) A practical sequential stopping rule for high- dimensional Markov chain Monte Carlo. *Journal of Computational and Graphical Statistics*, **25**, 684–700.

Gordon, N. J., Salmond, S. J. and Smith, A. F. M. (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, **140**, 107–113.

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

— (2003) Trans-dimensional Markov chain Monte Carlo. In *Highly Structured Stochastic Systems* (eds. P. J. Green, N. L. Hjort and S. Richardson), Oxford Statistical Science Series, chap. 6, 179–206. Oxford University Press.

Guihennec-Jouyaux, C., Mengersen, K. L. and Robert, C. P. (1998) Mcmc convergence diagnostics: A "reviewww". *Tech. Rep. 9816*, Institut National de la Statistique et des Etudes Economiques.

Hajek, B. (1988) Cooling schedules for optimal annealing. *Mathematics of Operations Research*, **13**, 311–329.

Halton, J. H. (1970) A retrospective and prospective survey of the Monte Carlo method. *SIAM Review*, **12**, 1–63.

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

Hwang, C.-R. (1980) Laplace's method revisited: Weak convergence of probability measures. *The Annals of Probability*, **8**, 1177–1182.

Johansen, A. M. (2009) Markov Chains. In *Encyclopaedia of Computer Science and Engineering* (ed. B. W. Wah), vol. 4, 1800–1808. 111 River Street, MS 8-02, Hoboken, NJ 07030-5774: John Wiley and Sons, Inc.

Johansen, A. M., Doucet, A. and Davy, M. (2008) Particle methods for maximum likelihood parameter estimation in latent variable models. *Statistics and Computing*, **18**, 47–57.

Jones, G. L. (2004) On the Markov chain central limit theorem. *Probability Surveys*, **1**, 299–320.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983) Optimization by simulated annealing. *Science*, **220, 4598**, 671–680.

Knuth, D. (1997) *The Art of Computer Programming*, vol. 1. Reading, MA: Addison-Wesley Professional.

Laplace, P. S. (1812) *Théorie Analytique des Probabilités*. Paris: Courcier.

Lazzarini, M. (1901) Un' applicazione del calcolo della probabilità alla ricerca esperimentale di un valore approssimato di π. *Periodico di Matematica*, **4**.

Liu, J. S. (2001) *Monte Carlo Strategies in Scientific Computing*. New York: Springer.

Liu, J. S., Wong, W. H. and Kong, A. (1995) Covariance structure and convergence rate of the Gibbs sampler with various scans. *Journal of the Royal Statistical Society B*, **57**, 157–169.

Livingstone, S. (2015) *Some contributions to the theory and methodology of Markov chain Monte Carlo*. Ph.D. thesis, University College London.

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

Marsaglia, G. (1968) Random numbers fall mainly in the planes. *Proceedings of the National Academy of Sciences of the United States of America*, **61**, 25–28.

Marsaglia, G. and Zaman, A. (1991) A new class of random number generators. *The Annals of Applied Probability*, **1**, 462–480.

Matsumoto, M. and Nishimura, T. (1998) Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, **8**, 3–30.

Medina-Aguayo, F. J., Lee, A. and Roberts, G. O. (2015) Stability of noisy Metropolis-Hastings. *ArXiv Mathematics e-prints 1503.0706*, ArXiv Mathematics e-prints.

Metropolis, N. (1987) The beginning of the Monte Carlo method. *Los Alamos Science*, **15**, 122–143.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. B., Teller, A. H. and Teller, E. (1953) Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, **21**, 1087–1092.

Metropolis, N. and Ulam, S. (1949) The Monte Carlo method. *Journal of the American Statistical Association*, **44**, 335–341.

Meyn, S. P. and Tweedie, R. L. (1993) *Markov Chains and Stochastic Stability*. Springer, New York, Inc. URL `http://probability.ca/MT/`.

Morokoff, W. J. and Caflisch, R. E. (1995) Quasi-Monte Carlo integration. *J. Comp. Phys.*, **122**, 218–230.

Neal, R. (2011) MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo* (eds. S. Brooks, A. Gelman, G. L. Jones and X.-L. Meng), 113–162. CRC Press.

Neal, R. M. (2001) Annealed importance sampling. *Statistics and Computing*, **11**, 125–139.

Niederreiter, H. (1992) *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics.

Nummelin, E. (1984) *General Irreducible Markov Chains and Non-Negative Operators*. No. 83 in Cambridge Tracts in Mathematics. Cambridge University Press, 1st paperback edn.

Philippe, A. and Robert, C. P. (2001) Riemann sums for mcmc estimation and convergence monitoring. *Statistics and Computing*, **11**, 103–115.

Richardson, S. and Green, P. J. (1997) On the bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society B*, **59**, 731–792.

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

Robert, C. P. and Casella, G. (2004) *Monte Carlo Statistical Methods*. Secaucus, NJ, USA: Springer, New York, Inc., 2 edn.

Roberts, G. and Tweedie, R. (1996) Geometric convergence and central limit theorems for multivariate Hastings and Metropolis algorithms. *Biometrika*, **83**, 95–110.

Roberts, G. O. (1996) Markov Chain concepts related to sampling algorithms. In Gilks et al. (1996), chap. 3, 45–54.

Roberts, G. O. and Rosenthal, J. S. (2004) General state space Markov chains and MCMC algorithms. *Probability Surveys*, **1**, 20–71.

Roeder, K. (1990) Density estimation with cofidence sets exemplified by superclusters and voids in galaxies. **85**, 617–624.

Roeder, K. and Wasserman, L. (1997) Practical Bayesian density estimation using mixtures of normals. **92**, 894–902.

Tierney, L. (1994) Markov Chains for exploring posterior distributions. *The Annals of Statistics*, **22**, 1701–1762.

— (1996) Introduction to general state space Markov Chain theory. In Gilks et al. (1996), chap. 4, 59–74.

Ulam, S. (1983) *Adventures of a Mathematician*. New York: Charles Scribner's Sons.