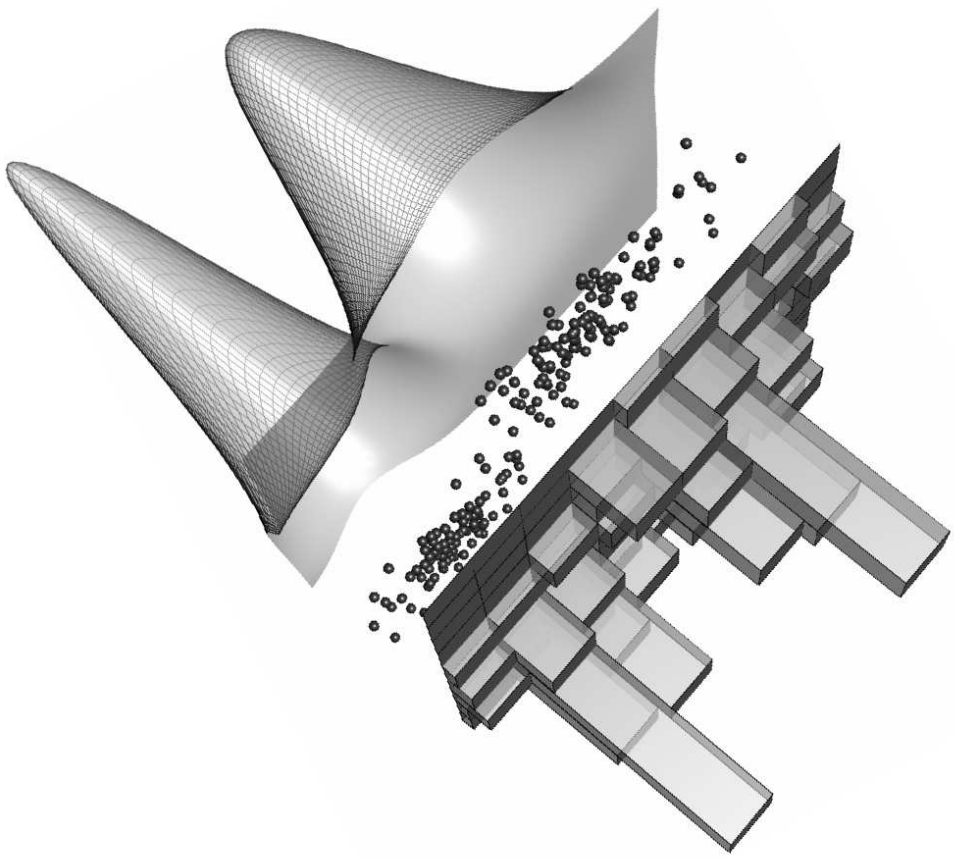
Adam M. Johansen and Ludger Evers

Monte Carlo Methods

Lecture Notes

November 15, 2007



University of BRISTOL
Department of Mathematics

Table of Contents

1. Introduction

1.1 What are Monte Carlo Methods?

This lecture course is concerned with Monte Carlo methods, which are sometimes referred to as *stochastic simulation* (Ripley (1987) for example only uses this term).

Examples of Monte Carlo methods include stochastic integration, where we use a simulation-based method to evaluate an integral, Monte Carlo tests, where we resort to simulation in order to compute the p-value, and Markov-Chain Monte Carlo (MCMC), where we construct a Markov chain which (hopefully) converges to the distribution of interest.

A formal definition of Monte Carlo methods was given (amongst others) by Halton (1970). He 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

Example 1.1 (A raindrop experiment for computing π). Assume we want to compute an Monte Carlo estimate of π using a simple experiment. Assume that we could 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 $X, Y \overset{\text{i.i.d.}}{\sim} \mathsf{U}[-1,1]$).
Now consider the probability that a raindrop falls into the unit circle (see figure 1.1). It is

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

In other words,

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

i.e. we found a way of expressing the desired quantity π as a function of a probability.
Of course we cannot compute $\mathbb{P}$(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:

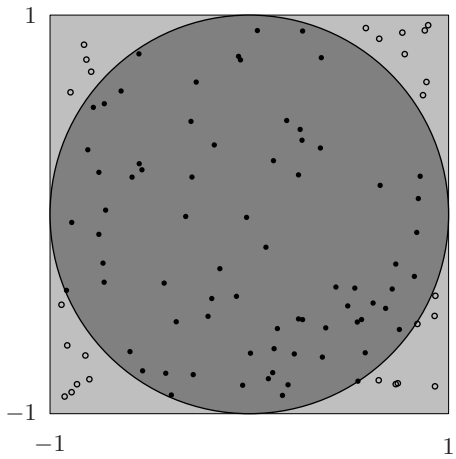


Fig. 1.1. Illustration of the raindrop experiment for estimating π

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

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 relatively poor.

However the *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 $X \sim \mathsf{B}(100, p)$, 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:

$$[2.750,\ 3.410]$$

In more general, 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

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

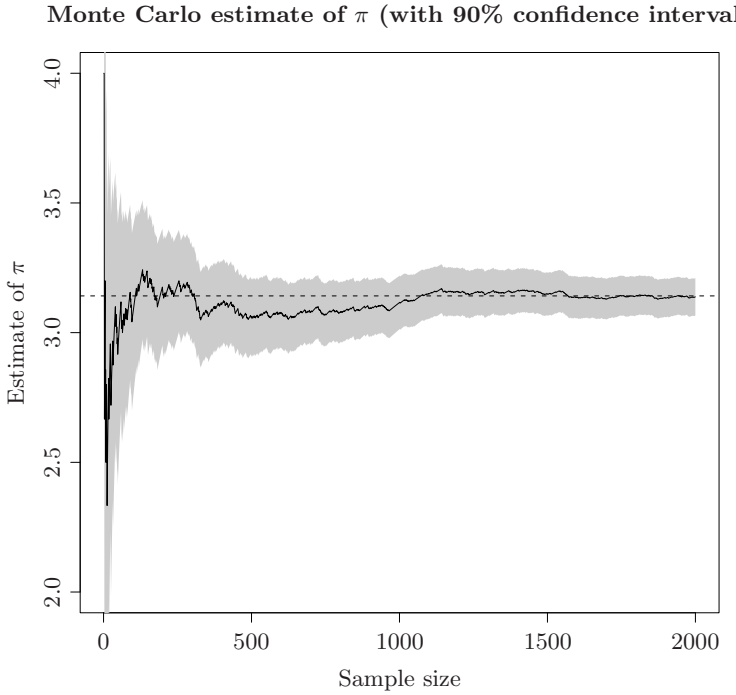


Fig. 1.2. Estimate of π resulting from the raindrop experiment

Let us recall again the different steps we have used in the example:

– We have written the quantity of interest (in our case π) as an expectation.[1]
– Second, we have replaced this algebraic representation of the quantity of interest by a sample approximation to it. The law of large numbers guaranteed that the sample approximation converges to the algebraic representation, and thus to the quantity of interest. Furthermore we used the central limit theorem 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.[2] 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 $f(x) = \int_0^{f(x)} 1\,dt$,

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

[1] A probability is a special case of an expectation as $\mathbb{P}(A) = \mathbb{E}(\mathbb{I}_A)$.

[2] As f is a polynomial we can obtain the result analytically, it is $\frac{4096}{8505} = \frac{2^{12}}{3^5\cdot 5\cdot 7} \approx 0.4816$.

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

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

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

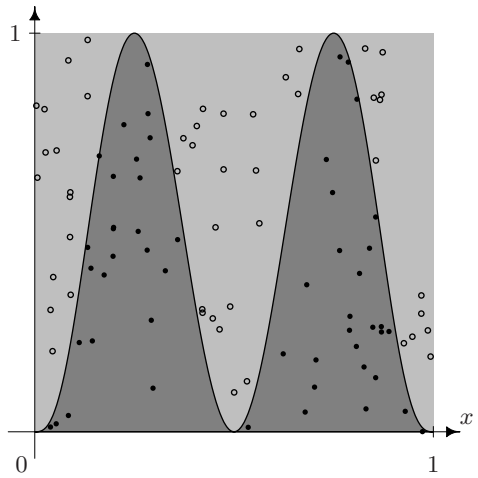


Fig. 1.3. Illustration of the raindrop experiment to compute $\int_0^1 f(x)dx$

When using Riemann sums (as in figure 1.4) to approximate the integral from example 1.2 the error is of order $O(n^{-1})$.[3,4]

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 $O(n^{-1/2})$. [5]

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.

---

[3] 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 $O(n^{-2})$. As there are n "bars", the total error is $O(n^{-1})$.

[4] The order of convergence can be improved when using the trapezoid rule and (even more) by using Simpson's rule.

[5] 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 $O(m^{-3})$ (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 $O(m^{-3})$). There are in total m^2 bars, so the total error is only $O(m^{-1})$, or equivalently $O(n^{-1/2})$.

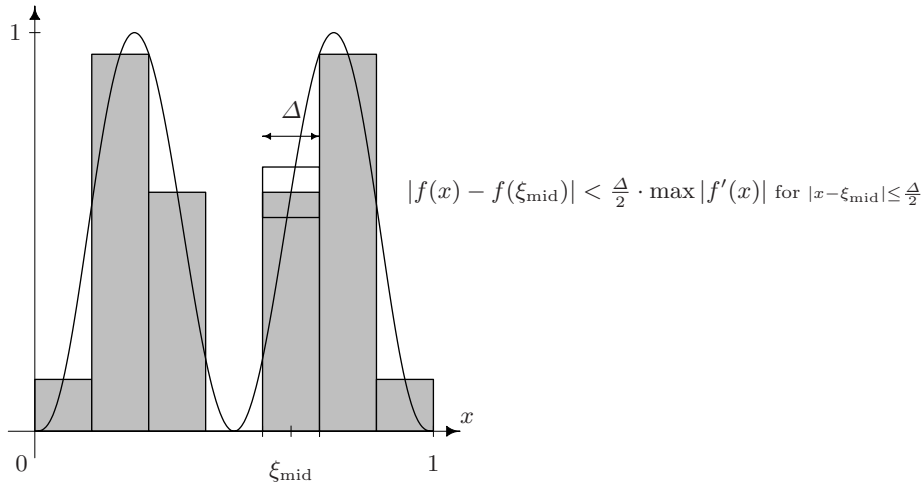
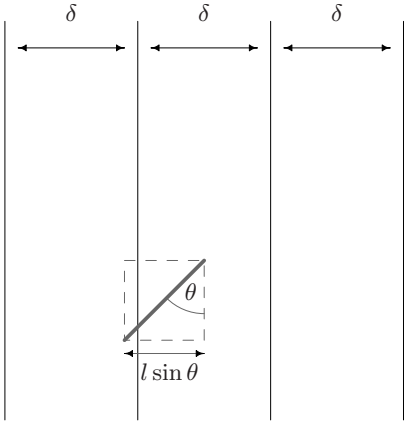


Fig. 1.4. Illustration of numerical integration by Riemann sums

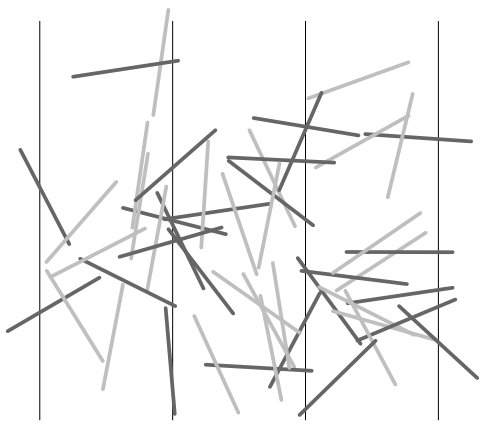
1.3 A Brief History of Monte Carlo Methods

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 are a somewhat more recent discipline. 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.

Fig. 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.[6] ◁

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 Neuman 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. In the following chapters we will assume that independent (pseudo-)random realisations from a uniform $\mathsf{U}[0, 1]$ distribution[7] are readily

[6] That Lazzarini's experiment was that precise, however, casts some doubt over the results of his experiments (see Badger, 1994, for a more detailed discussion).

[7] We will only use the $\mathsf{U}(0, 1)$ distribution as a source of randomness. Samples from other distributions can be derived from realisations of $\mathsf{U}(0, 1)$ random variables using deterministic algorithms.

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 (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}. This property is often referred to as the lack of predictability.
- 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 easy 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. Cosider 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[8], 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 for 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).

[8] The $(k+1)$-th point has the coordinates $(X_{nk+1}, \ldots, X_{nk+n-1})$.

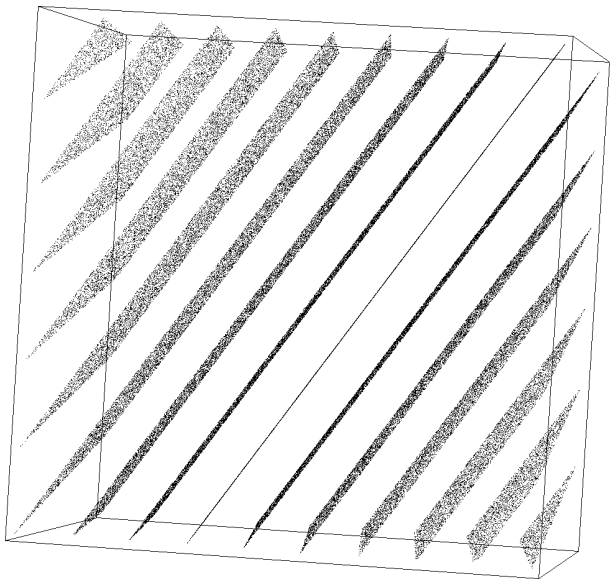
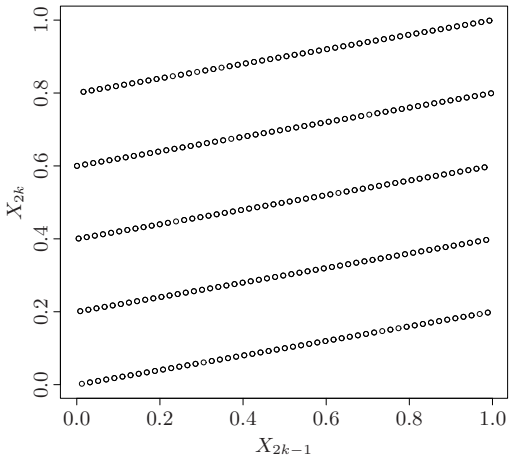
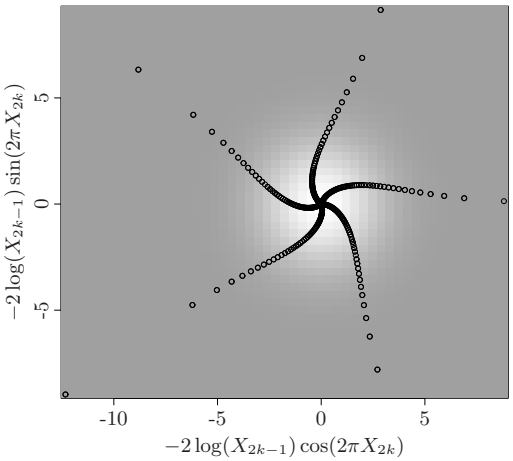


Fig. 1.6. 300,000 realisations of the RANDU pseudo-random number generator plotted in 3D. A point corresponds to a triplet $(x_{3k-2}, x_{3k-1}, x_{3k})$ for $k = 1, \ldots, 100000$. The data points lie on 15 hyperplanes.



(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).

Fig. 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 Gaussians (see example 2.2).

Due to this flaw of the congruential pseudo-random number generator, it should not be used in Monte Carlo experiments. 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.

2. Fundamental Concepts: Transformation, Rejection, and Reweighting

2.1 Transformation Methods

In section 1.4 we have seen how to create (pseudo-)random numbers from the uniform distribution $\mathsf{U}[0,1]$. 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.

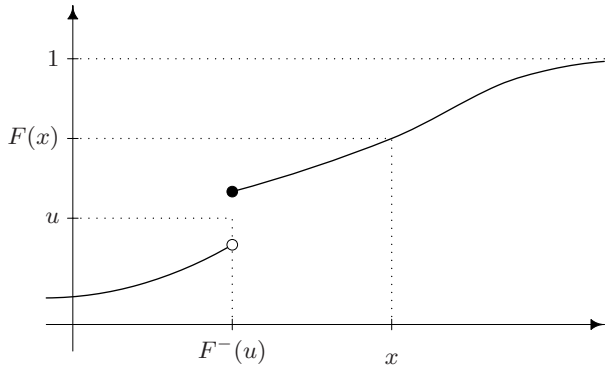


Fig. 2.1. Illustration of the definition of the generalised inverse F^- of a CDF F

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 $\mathsf{Expo}(\lambda)$ by applying the transformation $-\log(1-U)/\lambda$ to a uniform $\mathsf{U}[0,1]$ 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 very few distributions possess a CDF whose (generalised) inverse can be evaluated efficiently. Take the example of the Gaussian 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 the desired distribution. An example of such a method is the Box-Muller method for generating Gaussian random variables.

Example 2.2 (Box-Muller Method for Generating Gaussians). Using the transformation of density formula one can show that $X_1, X_2 \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0,1)$ 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 $R^2 \sim \mathsf{Expo}(1/2)$. Using $U_1, U_2 \overset{\text{i.i.d.}}{\sim} \mathsf{U}[0,1]$ 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 a 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. In many situations, we cannot find such a transformation in closed form. In these cases 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.

2.2 Rejection Sampling

The basic idea of rejection sampling is to sample from an *instrumental distribution*[1] and reject samples that are "unlikely" under the target distribution.

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

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

Thus $f(x)$ can be interpreted as the marginal density of a uniform distribution on the area under the density $f(x)$

$$\{(x,u) : \ 0 \le u \le f(x)\}.$$

Figure 2.2 illustrates this idea. This suggests that we can generate a sample from f by sampling from the area under the curve.

Example 2.3 (Sampling from a Beta distribution). The $\mathsf{Beta}(a,b)$ distribution $(a, b \ge 0)$ has the density

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

[1] The instrumental distribution is sometimes referred to as *proposal distribution*.

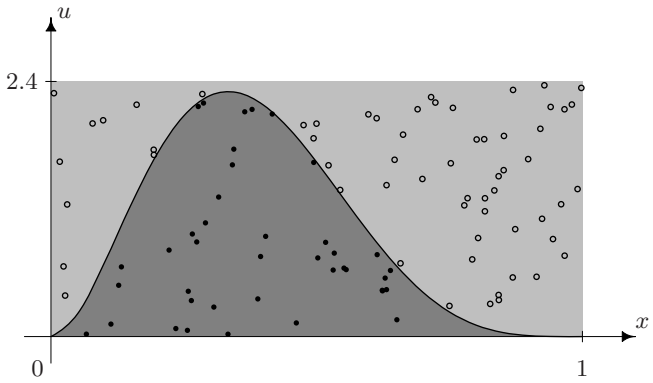


Fig. 2.2. Illustration of example 2.3. Sampling from the area under the curve (dark grey) corresponds to sampling from the $\mathsf{Beta}(3,5)$ density. In example 2.3 we use a uniform distribution of the light grey rectangle as proposal distribution. Empty circles denote rejected values, filled circles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

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

Using the above identity we can draw from $\mathsf{Beta}(3,5)$ 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 as in examples 1.1 and 1.2. We will sample from the light grey rectangle and and only keep the samples that fall in the area under the curve. Figure 2.2 illustrates this idea.

Mathematically speaking, we sample independently $X \sim \mathsf{U}[0,1]$ and $U \sim \mathsf{U}[0,2.4]$. 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 $\mathsf{U}[0,1]$ 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 that we can easily sample from.

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[2]

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

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 rejecting 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.[3] 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.

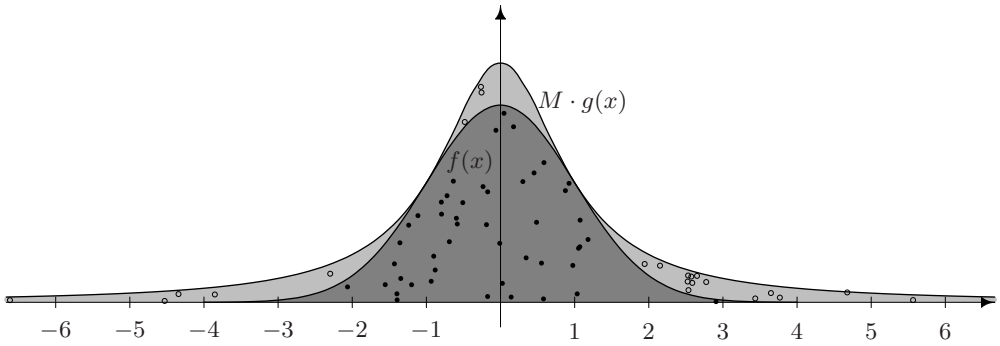


Fig. 2.3. Illustration of example 2.3. Sampling from the area under the density $f(x)$ (dark grey) corresponds to sampling from the $\mathsf{N}(0,1)$ density. The proposal $g(x)$ is a $\mathsf{Cauchy}(0,1)$.

[2] We denote by E the set of all possible values X can take.

[3] There is not much point is using this method is practise. The Box-Muller method is more efficient.

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 Gaussian distribution. ◁

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

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

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

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 $\mathrm{supp}(g) \supset \mathrm{supp}(f\cdot h)$.

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

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

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

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

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

(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 it is often the case in hierarchical Bayesian modelling. 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.[4] 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 $\mathrm{supp}(g) \supset \mathrm{supp}(f \cdot h)$ 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) for a finite variance of $\tilde{\mu}$:

– $f(x) < M \cdot g(x)$ and $\mathrm{Var}_f(h(X)) < \infty$.
– 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 $\mathrm{Var}(\tilde{\mu})$ 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*

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

[4] By complete analogy one can show that is enough to know g up to a multiplicative constant.

Proof. We have from theroem 2.2 (b) that

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

Thus we only have to minimise $\mathbb{E}_g\left(\left(\frac{h(X)\cdot f(X)}{g(X)}\right)^2\right)$. When plugging 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 the Jensen inequality to $\mathbb{E}_g\left(\left(\frac{h(X)\cdot f(X)}{g(X)}\right)^2\right)$ yielding

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

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 $h(X)$ 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 looking for. Further we need to be able to draw samples from $g^\star$ efficiently. The practically important corollary 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 $g_{\mathsf{t}_1}(x)$ 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 $\mathsf{N}(0,1)$ distribution as instrumental distribution.

Note that the third choice yields weights of infinite variance, as the instrumental distribution ($\mathsf{N}(0,1)$) 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 $\int|x|f(x)\,dx$.

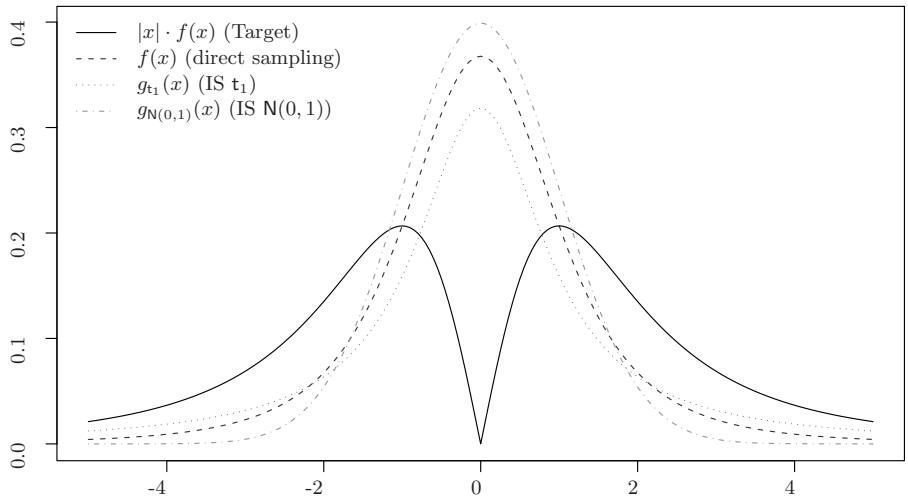


Fig. 2.4. Illustration of the different instrumental distributions in example 2.5.

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. So using a t_1 distribution as instrumental distribution is super-efficient (see figure 2.5).

Figure 2.6 somewhat explains why the t_1 distribution is a far better choice than the $\mathsf{N}(0,1)$ distributon. As the $\mathsf{N}(0,1)$ distribution does not have heavy enough tails, the weight tends to infinity as $|x| \to +\infty$. Thus large $|x|$ get large weights, causing the jumps of the estimate $\tilde{\mu}$ shown in figure 2.5. The t_1 distribution has heavy enough tails, so 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. ◁

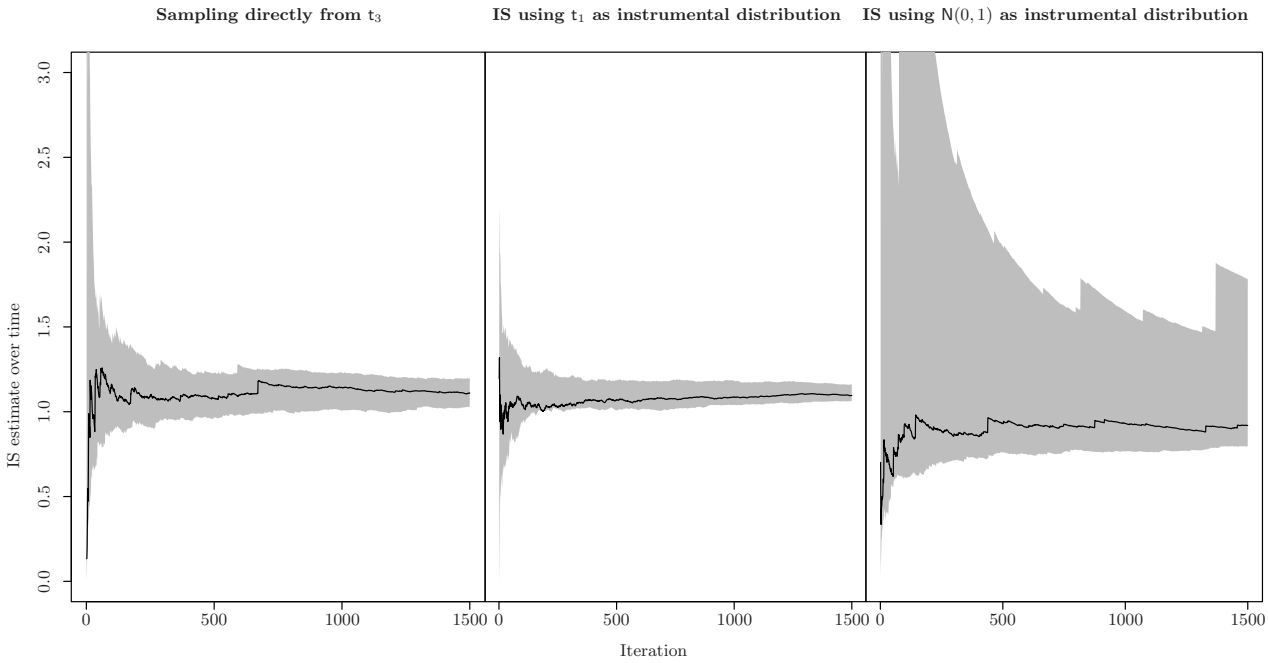


Fig. 2.5. Estimates of $\mathbb{E}|X|$ for $X \sim \mathsf{t}_3$ obtained after 1 to 1500 iterations. The three panels correspond to the three different sampling schemes used. The areas shaded in grey correspond to the range of 100 replications.

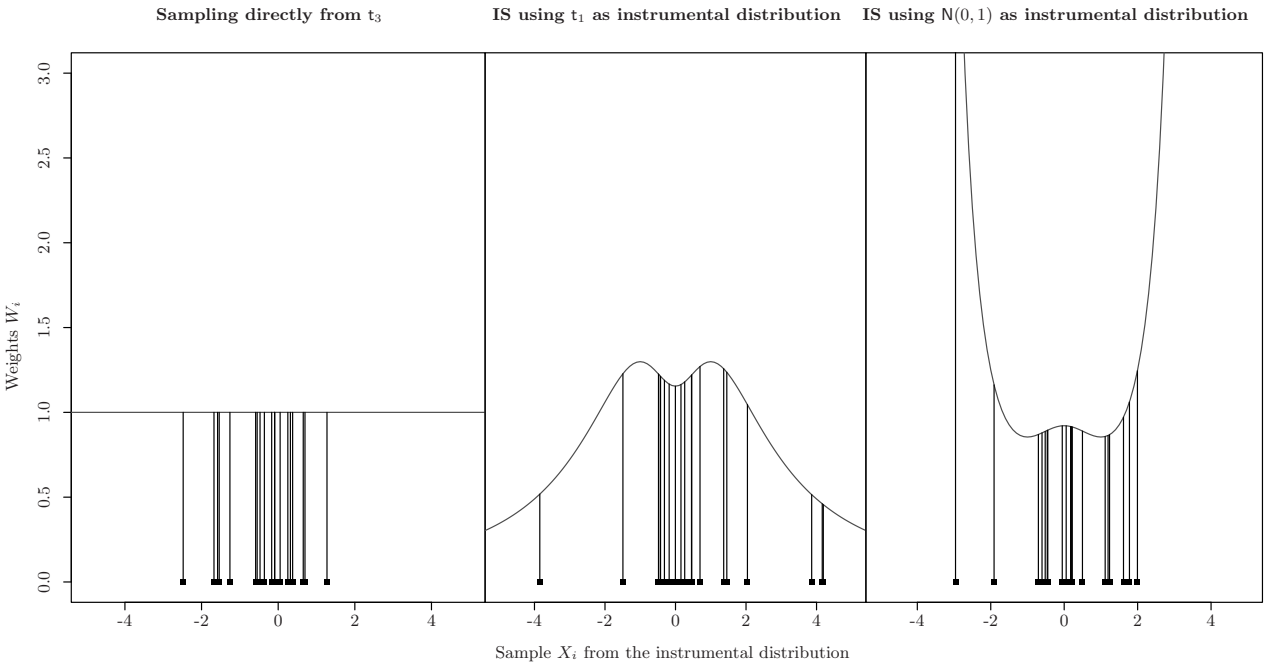


Fig. 2.6. Weights W_i obtained for 20 realisations X_i from the different instrumental distributions.

3. Markov Chains

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 all 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:

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

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:

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

Having established that 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 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 $i+1$ conditional upon the state of the process at time i) 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:

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

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

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

$$\mathbb{E}\left[\varphi(\xi_{t+1}, \ldots, \xi_{t+q}) | \xi_1 = x_1, \ldots \xi_t = x_t\right] = \mathbb{E}\left[\varphi(\xi_2, \ldots, \xi_{q+1}) | \xi_1 = x_t\right].$$

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 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. (2002). 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:

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

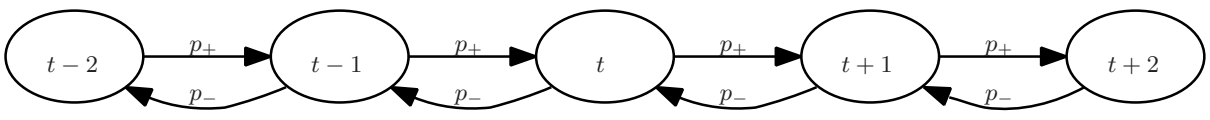


Fig. 3.1. A simple random walk on $\mathbb{Z}$.

◁

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 $p_+ + p_- < 0$ and $p_0 = 1 - (p_+ + p_-)$. In this case we may write the transition kernel as:

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

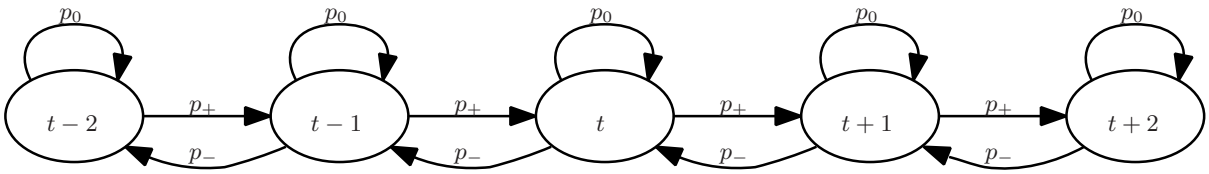


Fig. 3.2. A random walk on $\mathbb{Z}$ with $K_{tt} > 0$.

◁

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:

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

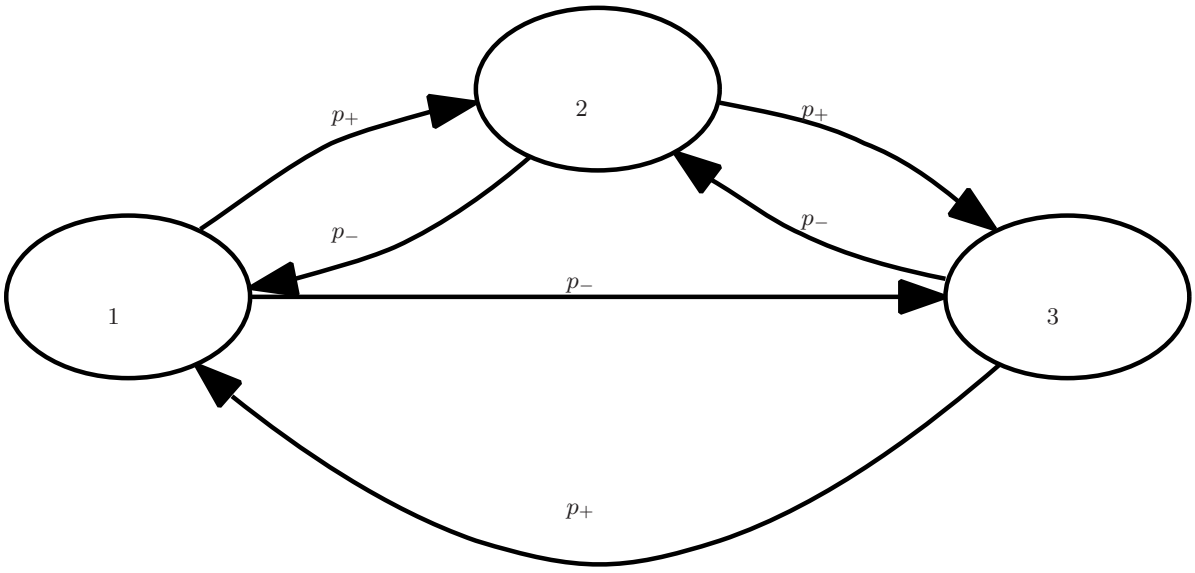


Fig. 3.3. A random walk on a triangle.

◁

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:

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

◁

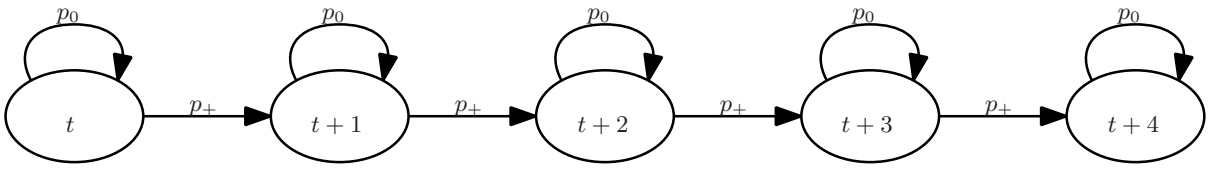


Fig. 3.4. A random walk on the positive integers.

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:*

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

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

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

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:*

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

whilst, if we have that,

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

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:

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

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. It requires a degree of technical sophistication which lies somewhat outside the scope of this course to do so rigorously. 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, (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 $A \subset E$:

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

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 $x \in E$:

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

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:

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

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

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

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

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

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 is constant over 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[2] – most interesting continuous state spaces do not possess atoms. The condition that the distribution in the next state is precisely the same, wherever the current state is rather strong. Another approach would be to require only that the conditional distribution has a common component, 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 m, some $\epsilon > 0$ and some non-trivial probability distribution, ν, such that:*

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

This tells us that the distribution m-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 for a number of sophisticated analytic 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 have a finite 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.

Definition 3.11 (Cycles). *A μ-irreducible Markov chain has a* cycle of length d *if there exists a small set C, an associated integer M and some probability distribution ν_s which has positive mass on C (i.e. $\int_C \nu_s(x)dx > 0$) such that:*

$$d = \gcd \{s \geq 1 : C \text{ is small for some } \nu_s \geq \delta_s \nu_s \text{ with } \delta_s > 0\}.$$

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:

– $\forall i \neq j : E_i \cap E_j = \emptyset$
– $\bigcup_{i=1}^{d} E_i = E$

[2] Note that this is much stronger than knowledge of the transition kernel, K, as in general all points in the space have zero probability.

–

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

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.:*

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

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 $\int_A \mu(y)dy > 0$, $\mathbb{E}_x\left[\eta_A\right] = \infty$ 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 $\mathbb{P}_x\left(\eta_A = \infty\right) = 1$ 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 $f : E \to \mathbb{R}$:*

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

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 $f : E \to \mathbb{R}$:*

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

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 $f : E \to \mathbb{R}$ 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 $2 + \delta$).*

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

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

where $\bar{f} = \int f(x)\pi(x)dx$.

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 (2008).
– (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 have seen that, using importance sampling, we can approximate an expectation $\mathbb{E}_f(h(X))$ without having to sample directly from f. However, finding an instrumental distribution which allows us to *efficiently* estimate $\mathbb{E}_f(h(X))$ 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 obtaining an *approximate* sample from 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.

Example 4.1 (Poisson change point model). Assume the following Poisson model of two regimes for n random variables $Y_1, \ldots, Y_n$.[1]

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

A suitable (conjugate) prior distribution for λ_j is the $\mathsf{Gamma}(\alpha_j, \beta_j)$ 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

[1] The probability distribution function of the $\mathsf{Poi}(\lambda)$ distribution is $p(y) = \frac{\exp(-\lambda)\lambda^y}{y!}$.

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

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

Now assume that we do not know the change point M and that we assume a uniform prior on the set $\{1,\dots,M-1\}$. 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 density function proportional to

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

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. The next section will however first of all state 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). Denote with $x_{-i} := (x_1,\dots,x_{i-1},x_{i+1},\dots,x_p)$.

Algorithm 4.1 ((Systematic sweep) 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 as used in the Gibbs sampler are often referred to as *full conditionals*. Note that the Gibbs sampler is *not* reversible. Liu et al. (1995) proposed the following algorithm that yields a reversible chain.

Algorithm 4.2 (Random sweep Gibbs sampler). Starting with $(X_1^{(0)},\dots,X_p^{(n)})$ 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 $\iota \neq j$.

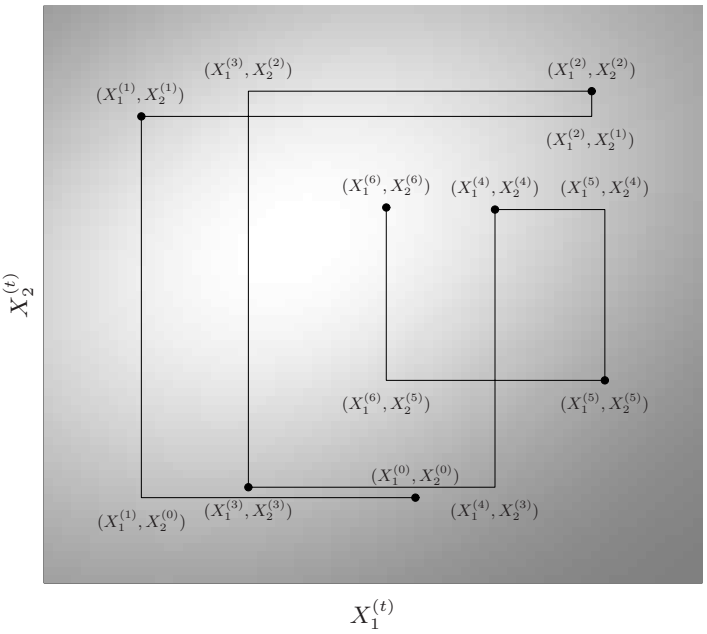


Fig. 4.1. Illustration of the Gibbs sampler for a two-dimensional distribution

4.3 The Hammersley-Clifford Theorem

An interesting property of the full conditionals, which the Gibbs sampler is based on, is that they fully specify the joint distribution, as Hammersley and Clifford proved in 1970[2]. 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 support 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 complete analogy

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

thus

[2] Hammersley and Clifford actually never published this result, as they could not extend the theorem to the case of non-positivity.

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

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 mostly arise when using improper prior distributions.

Example 4.2. Consider the following "model"

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

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 analyse whether the joint distribution $f(x_1,\ldots,x_p)$ is indeed the stationary distribution of the Markov chain generated by the Gibbs sampler[3]. For this 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

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

[3] 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).

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 analyse under which conditions 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[4]. 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

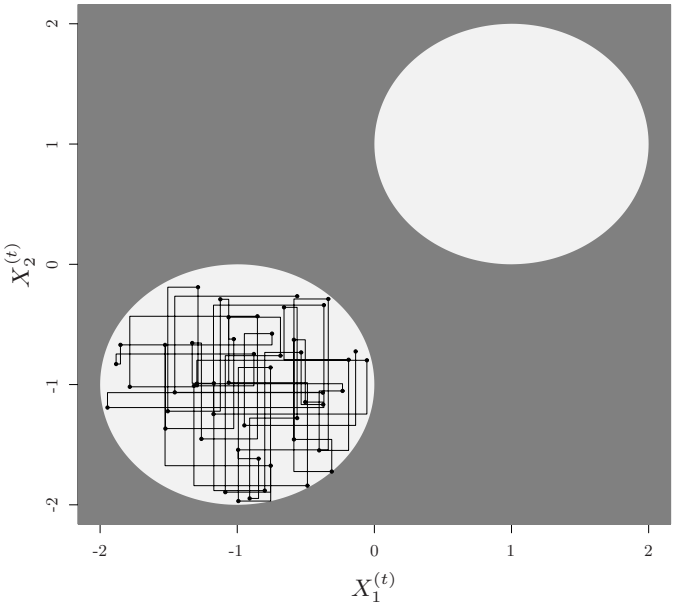


Fig. 4.2. Illustration of a Gibbs sampler failing to sample from a distribution with unconnected support (uniform distribution on $\{(x_1, x_2) : \|(x_1, x_2) - (1, 1)\| \leq 1|\} \cup \{(x_1, x_2) : \|(x_1, x_2) - (-1, -1)\| \leq 1|\}$)

and never reach 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 $\int_{\mathcal{X}} f(x_1^{(t)}, \ldots, x_p^{(t)}) d(x_1^{(t)}, \ldots, x_p^{(t)}) > 0$.

[4] Here and in the following we understand by "irreducibilty" irreducibility with respect to the target distribution f.

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

Thus the Markov Chain $(\mathbf{X}^{(t)})_t$ is strongly f-irreducible. As f is the invariant distribution of the Markov chain, it is as well recurrent (see the remark on page 36). □

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 theorems 3.1 and 3.2.

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 $h : E \to \mathbb{R}$*

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

for 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 $\mathbb{E}_f\left(h(\mathbf{X})\right)$ 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.[5] 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[6]

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

[5] 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 $Z_1, \ldots, Z_p \overset{\text{i.i.d.}}{\sim} N(0,1)$ and set $(X_1, \ldots, X_p)' = \boldsymbol{\Sigma}^{1/2}(Z_1, \ldots, Z_p)' + \boldsymbol{\mu}$

[6] 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. ◁

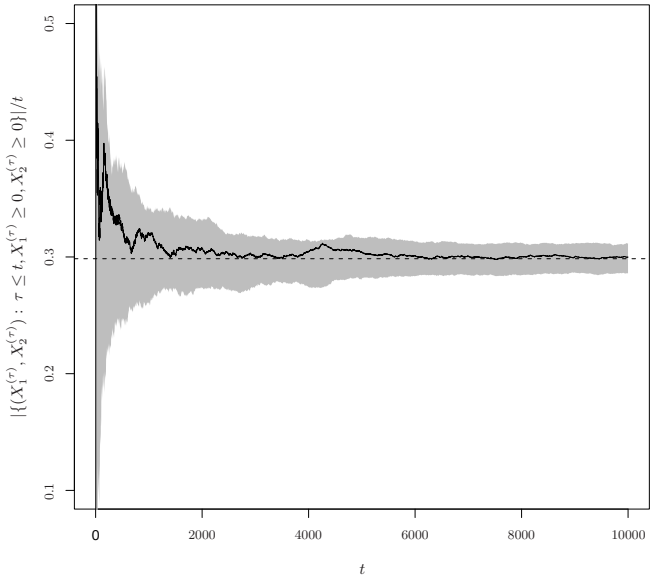
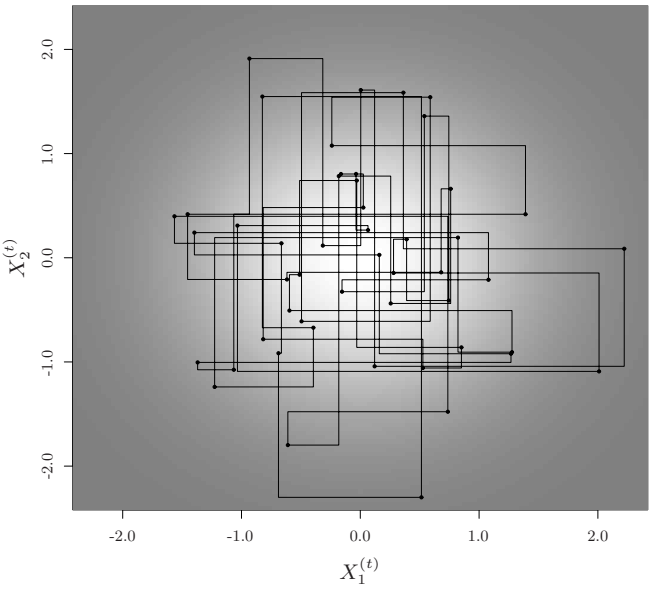


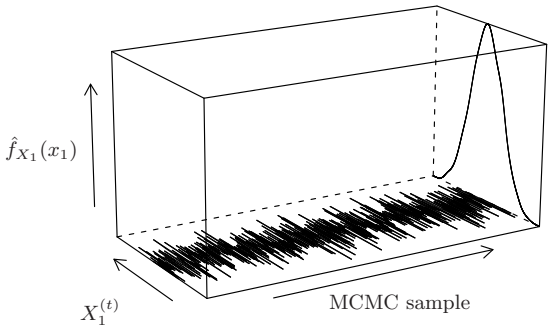
Fig. 4.3. Estimate of the $\mathbb{P}(X_1 \geq 0, X_2 \geq 0)$ obtained using a Gibbs sampler. The area shaded in grey corresponds to the range of 100 replications.

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.

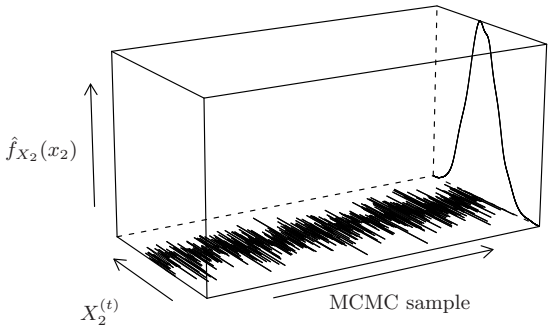
Example 4.5 (Sampling from a highly correlated bivariate Gaussian). Figure 4.5 shows the results obtained when sampling from a bivariate Normal distribution as in example 4.4, however with $\sigma_{12} = 0.99$. This yields a correlation of $\rho(X_1, X_2) = 0.99$. This Gibbs sampler is a lot slower mixing 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 $X_j^{(j)}$ highly correlated and thus yields to a 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 $(X_1^{(t)}, X_2^{(t)})$

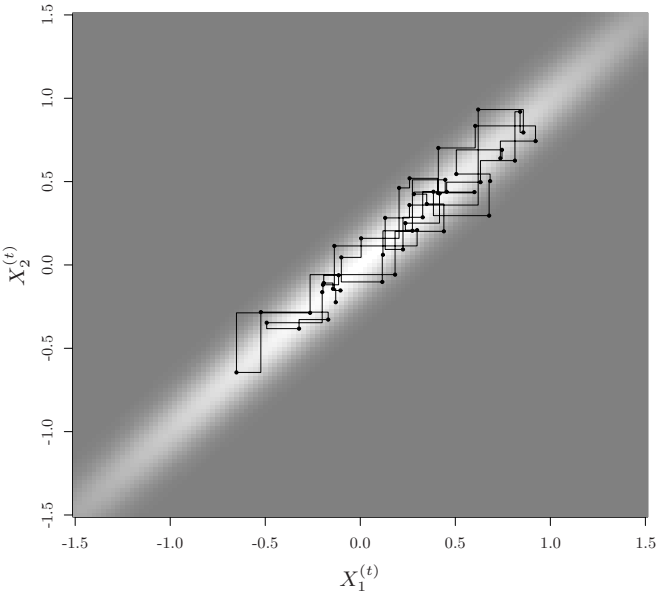


(b) Path of $X_1^{(t)}$ and estimated density of X after 1,000 iterations

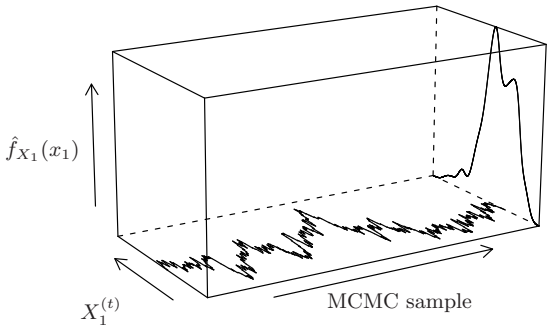


(c) Path of $X_2^{(t)}$ and estimated density of X_2 after 1,000 iterations

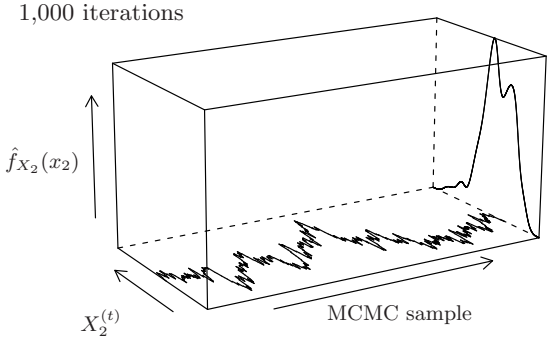
Fig. 4.4. Gibbs sampler for a bivariate standard normal distribution (correlation $\rho(X_1, X_2) = 0.3$)



(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

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

4.5 Data Augmentation

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 $Z_1, \ldots, Z_r$ such that f is the marginal density of $(X_1, \ldots, X_p, Z_1, \ldots, Z_r)$, i.e.

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

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

Example 4.6 (Mixture of Gaussians). Consider data $Y_1, \ldots, Y_n$, 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 which population the i-th observation is from, it is from a *mixture distribution*:

$$f(y_i) = \sum_{k=1}^{K} \pi_k \phi_{(\mu_k, 1/\tau)}(y_i). \tag{4.6}$$

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 $(\pi_1, \ldots, \pi_K)$ is the Dirichlet distribution[7] with parameters $\alpha_1, \ldots, \alpha_K > 0$ with density

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

for $\pi \geq 0$ and $\sum_{k=1}^K \pi_k = 1$. For the sake of simplicity we assume that the dispersion τ is known[8], as well as the number populations K.[9]

It is however difficult to sample from the posterior distribution of $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$ given data $Y_1, \ldots Y_n$ using a Gibbs sampler. This is due to the mixture nature of (4.6). This suggests introducing auxiliary variables $Z_1, \ldots Z_n$ 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 (4.6), i.e. the Z_i are indeed a completion. Now we have that

$$\begin{aligned} & f(y_1, \ldots, y_n, z_1, \ldots, z_n, \mu_1, \ldots, \mu_K, \pi_1, \ldots, \pi_K) \\ \propto \; & \left(\prod_{i=1}^n \pi_{z_i} \exp\left(-\tau (y_i - \mu_{z_i})^2/2\right) \right) \cdot \left(\prod_{k=1}^K \exp\left(-\tau_0 (\mu_k - \mu_0)^2/2\right) \right) \cdot \left(\prod_{k=1}^K \pi_k^{\alpha_k - 1} \right). \end{aligned} \tag{4.7}$$

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

$$\begin{aligned} \mathbb{P}(Z_i = k | Y_1, \ldots, Y_n, \mu_1, \ldots, \mu_K, \pi_1, \ldots, \pi_K) &= \frac{\pi_k \phi_{(\mu_k, 1/\tau)}(y_i)}{\sum_{\iota=1}^K \pi_\iota \phi_{(\mu_\iota, 1/\tau)}(y_i)} \qquad (4.8) \\ \mu_k | Y_1, \ldots, Y_n, Z_1, \ldots, Z_n, \pi_1, \ldots, \pi_K &\sim \mathsf{N}\left(\frac{\tau \left(\sum_{i:\, Z_i = k} Y_i\right) + \tau_o \mu_0}{|\{:\; Z_i = k\}|\tau + \tau_0}, \frac{1}{|\{i:\; Z_i = k\}|\tau + \tau_0} \right) \\ \pi_1, \ldots, \pi_K | Y_1, \ldots, Y_n, Z_1, \ldots, Z_n, \mu_1, \ldots, \mu_K &\sim \mathsf{Dirichlet}\left(\alpha_1 + |\{i:\; Z_i = 1\}|, \ldots, \alpha_K + |\{I:\; Z_i = K\}|\right). \end{aligned}$$

[7] The Dirichlet distribution is a multivariate generalisation of the Beta distribution.
[8] Otherwise, a Gamma distribution would be a suitable choice.
[9] For a model where the number of components is variable, see section 6.

To derive the full conditional of μ_k we have used that the joint density (4.7) 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 $\mu_1, \ldots, \mu_K$ and $\pi_1, \ldots, \pi_K$ given observations $Y_1, \ldots, Y_n$ using the following auxiliary variable Gibbs sampler: Starting with initial values $\mu_1^{(0)}, \ldots, \mu_K^{(0)}, \pi_1^{(0)}, \ldots, \pi_K^{(0)}$ iterate the following steps for $t = 1, 2, \ldots$

1. For $i = 1, \ldots, n$:

   Draw $Z_i^{(t)}$ from the discrete distribution on $\{1, \ldots, K\}$ specified by (4.8).

2. For $k = 1, \ldots, K$:

   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 $(\pi_1^{(t)}, \ldots, \pi_K^{(t)}) \sim \mathsf{Dirichlet}\left(\alpha_1 + |\{i:\ Z_i^{(t)}=1\}|, \ldots, \alpha_K + |\{i:\ Z_i^{(t)}=K\}|\right)$. ◁

5. The Metropolis-Hastings Algorithm

5.1 Algorithm

In the previous chapter we have 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 the 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 2.1 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 (open disks 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.

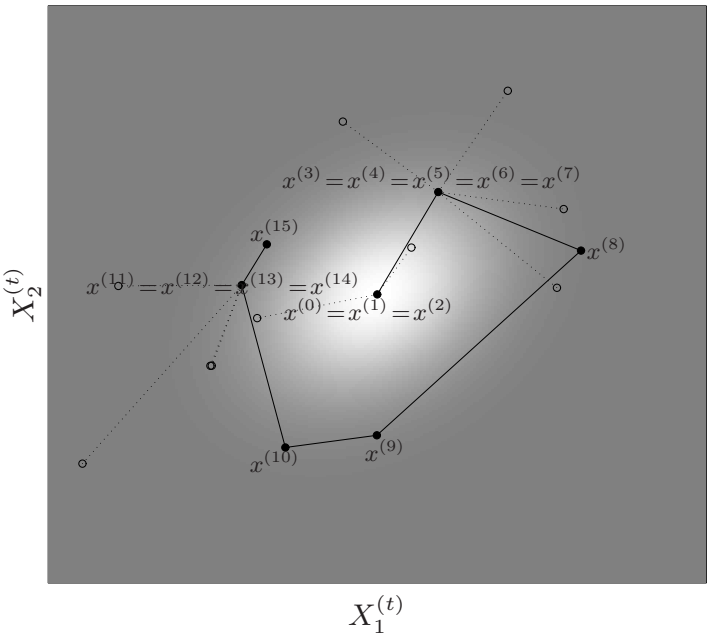


Fig. 5.1. Illustration of the Metropolis-Hastings algorithm. Filled dots denote accepted states, open circles rejected values.

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.[1]

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* continuous with respect to the Lebesgue measure.

Proof. We have

[1] On a similar note, it is enough to know $q(\mathbf{x}^{(t-1)}|\mathbf{x})$ up to a multiplicative constant independent of $\mathbf{x}^{(t-1)}$ and $\mathbf{x}$.

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

□

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

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

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 does not need to be 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.

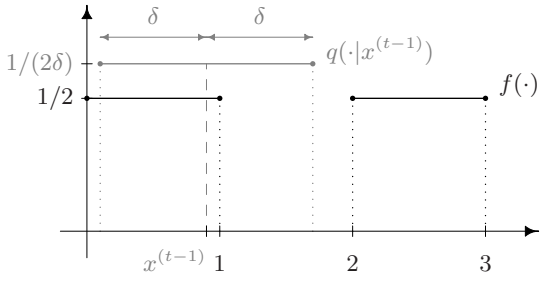


Fig. 5.2. Illustration of example 5.1

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

$$\forall\, \epsilon\, \exists\, \delta:\ q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) > \epsilon \text{ if } \|\mathbf{x}^{(t-1)} - \mathbf{x}^{(t)}\| < \delta$$

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 has f as invariant distribution: in this case we accept every proposed value with probability 1 (see e.g. remark 5.2).

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 36) 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 modeled 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 $0.08 \cdot \mathbb{I}$.
Figure 5.3 and table 5.2 show the results obtained using 50,000 samples[2]. Note that the convergence of the

		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

$\beta_j^{(t)}$ 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"). ◁

5.4 Choosing the proposal distribution

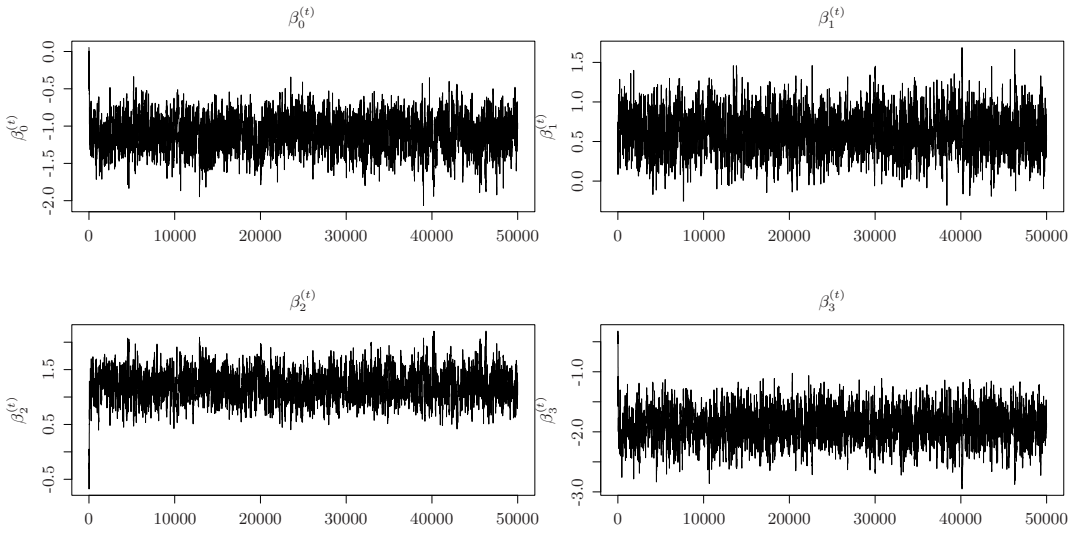
The efficiency of a Metropolis-Hastings sampler depends on the choice of the proposal distribution $q(\cdot|\mathbf{x}^{(t-1)})$. An ideal choice of proposal would lead to a small correlation of subsequent realisations $\mathbf{X}^{(t-1)}$ and $\mathbf{X}^{(t)}$. This correlation has two sources:

– 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.

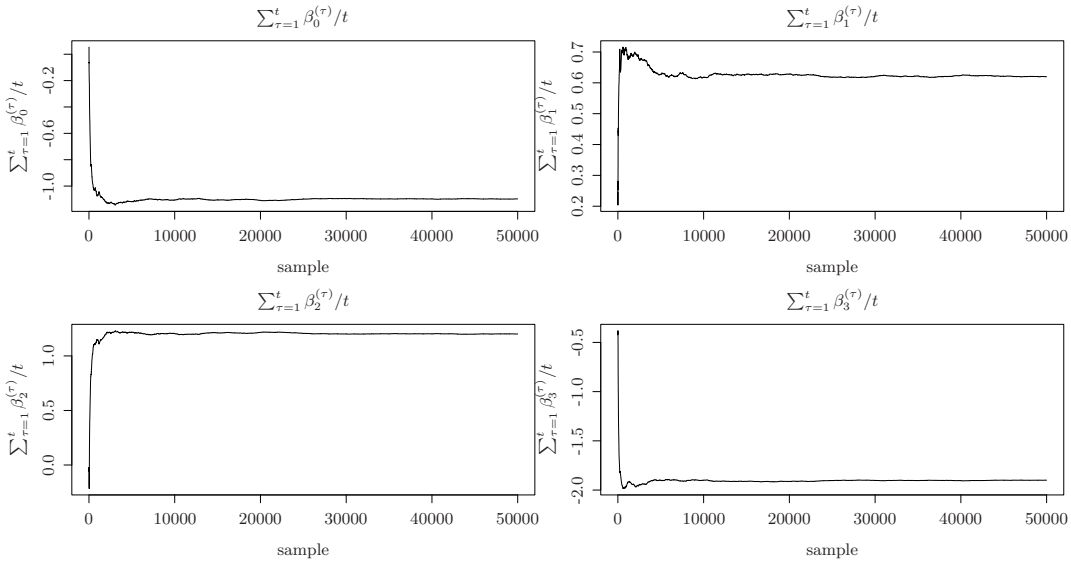
Thus we would ideally want a proposal distribution that both allows for fast changes in the $\mathbf{X}^{(t)}$ and yields a high probability of acceptance. Unfortunately these are two competing goals. If we choose a proposal distribution with a small variance, the probability of acceptance will be high, 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-Hastings 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-Hastings 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 too small yields a very high probability of acceptance, however at

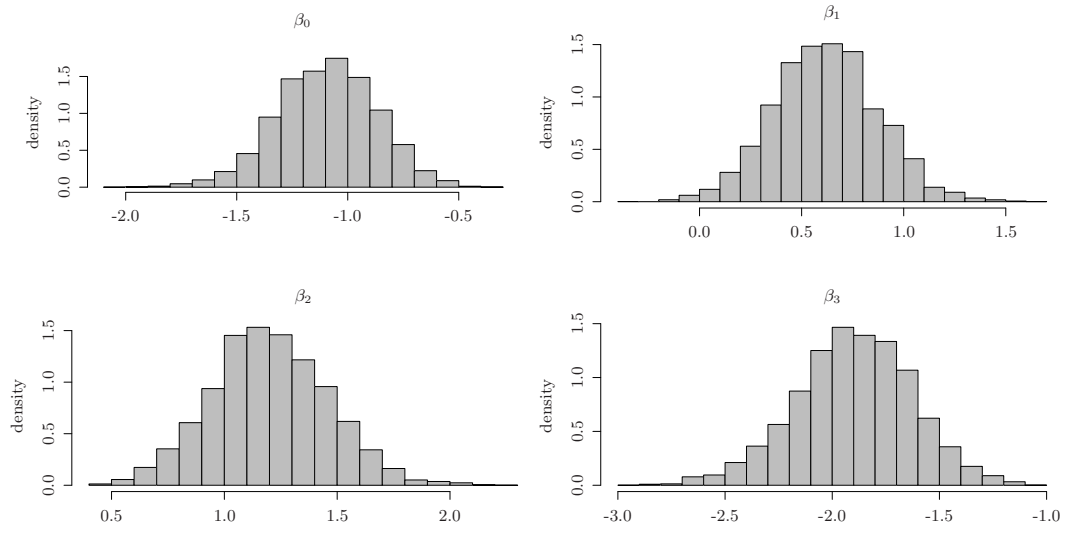
[2] You might want to consider a longer chain in practise.



(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

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

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.[3] This 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, making them a safer choice. The covariance structure of the proposal distribution should ideally reflect the expected covariance of the $(X_1, \ldots, X_p)$. 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 larger dimensions, which is in line with the results we obtained in the above simple example and the guidelines which motivate them. Note however that these are just rough guidelines.

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 modeled 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. ◁

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

[3] 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!

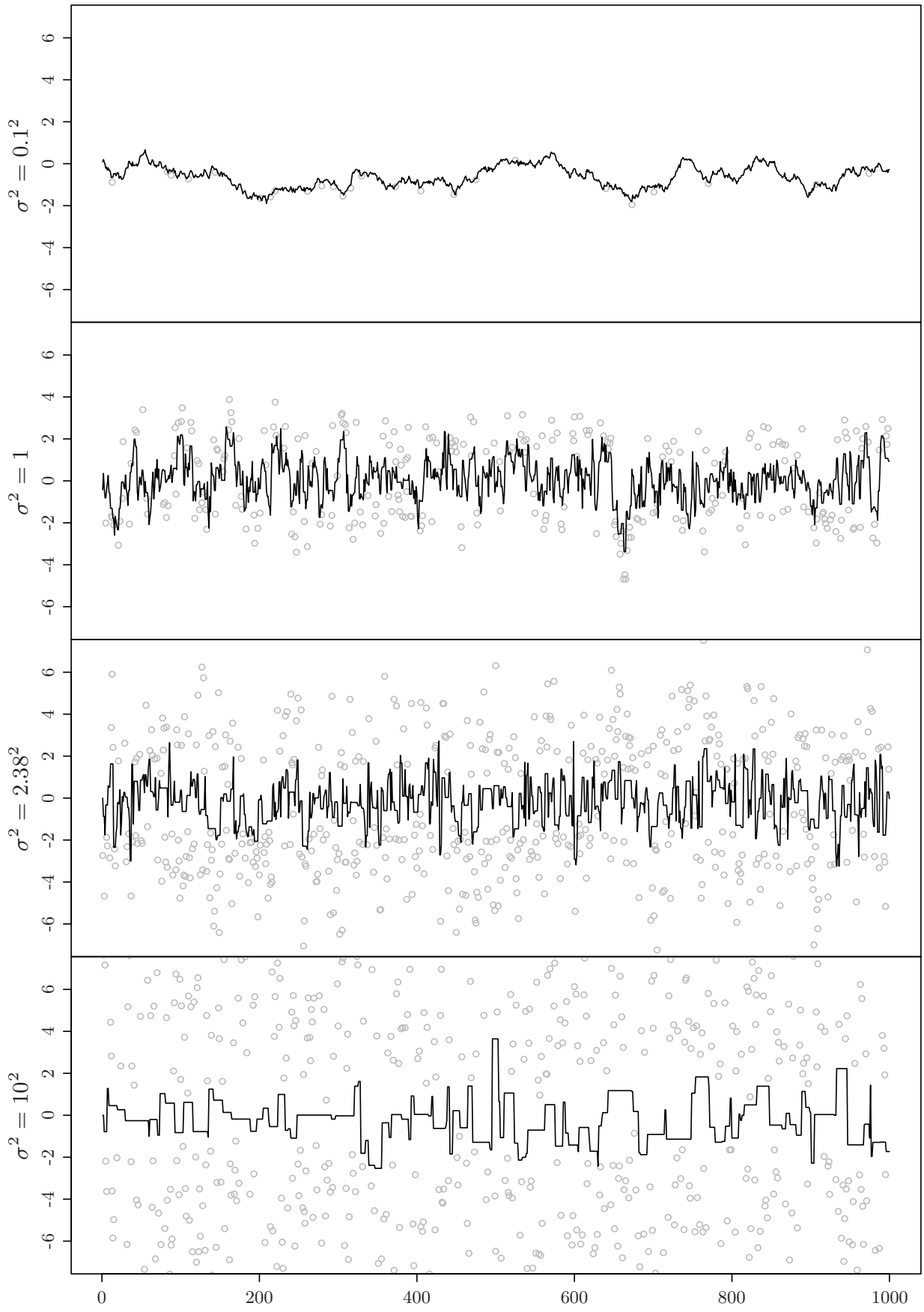


Fig. 5.4. Sample paths for example 5.3 for different choices of the proposal variance σ^2. Open grey discs represent rejected values.

(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 $\boldsymbol{\Sigma}$.

corresponds to a transition kernel $K^{(j)}$. As with the substeps of Gibbs sampler there are two ways of combining the transition kernels $K^{(1)}, \ldots, K^{(r)}$:

– 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

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

– 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 $\mathbf{X} = (X_1, \ldots, X_p)$. 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 $q_1, \dots, q_p$ 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 $\mathbf{X}^{(0)} = (X_1^{(0)}, \dots, X_p^{(0)})$ iterate

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

...

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

...

p. i. Draw $X_p \sim q_p(\cdot|X_1^{(t)}, \dots, X_{p-1}^{(t)}, X_p^{(t-1)})$.
   ii. Compute $\alpha_p = \min\left\{1, \frac{f(X_1^{(t)}, \dots, X_{p-1}^{(t)}, X_p) \cdot q_p(X_p^{(t-1)}|X_1^{(t)}, \dots, X_{p-1}^{(t)}, X_p)}{f(X_1^{(t)}, \dots, X_{p-1}^{(t)}, X_p^{(t-1)}) \cdot q_p(X_p|X_1^{(t)}, \dots, X_{p-1}^{(t)}, X_p^{(t-1)})}\right\}$.
   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 $\mathbf{X}^{(0)} = (X_1^{(0)}, \dots, X_p^{(0)})$ iterate

1. Draw an index j from a distribution on $\{1, \dots, p\}$ (e.g. uniform)
2. Draw $X_j \sim q_j(\cdot|X_1^{(t-1)}, \dots, X_p^{(t-1)})$.
3. Compute $\alpha_j = \min\left\{1, \frac{f(X_1^{(t-1)}, \dots, X_{j-1}^{(t-1)}, X_j, X_{j+1}^{(t-1)}, \dots, X_p^{(t-1)}) \cdot q_j(X_j^{(t-1)}|X_1^{(t-1)}, \dots, X_{j-1}^{(t-1)}, X_j, X_{j+1}^{(t-1)}, \dots, X_p^{(t-1)})}{f(X_1^{(t-1)}, \dots, X_{j-1}^{(t-1)}, X_j^{(t-1)}, X_{j+1}^{(t-1)}, \dots, X_p^{(t-1)}) \cdot q_j(X_j|X_1^{(t-1)}, \dots, X_{j-1}^{(t-1)}, X_j^{(t-1)}, X_{j+1}^{(t-1)}, \dots, X_p^{(t-1)})}\right\}$.
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

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

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. The Reversible Jump Algorithm

6.1 Bayesian multi-model inference

Examples 4.1, 4.6, and 5.2 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, 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 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 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 $\mathcal{M}_k$

$$\mathbb{P}(\mathcal{M}_k|\mathbf{y}_1, \ldots, \mathbf{y}_n) = \frac{p_k \int_{\Theta_k} f_k^{\text{prior}}(\boldsymbol{\theta}) l_k(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta})\, d\boldsymbol{\theta}}{\sum_\kappa p_\kappa \int_{\Theta_\kappa} f_\kappa^{\text{prior}}(\boldsymbol{\theta}) l_\kappa(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta})\, d\boldsymbol{\theta}}$$

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 on the models we can now either consider the model with the highest posterior probability $\mathbb{P}(\mathcal{M}_k|\mathbf{y}_1,\ldots,\mathbf{y}_n)$ or perform *model averaging* using $\mathbb{P}(\mathcal{M}_k|\mathbf{y}_1,\ldots,\mathbf{y}_n)$ as weights.

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

$$\Theta := \bigcup_k (\{k\} \times \Theta_k)\,.$$

Unfortunately, we cannot use the techniques presented in chapters 4 and 5 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 of chapters 4 and 5 fails to apply.

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.

6.2 Another look at the Metropolis-Hastings algorithm

Recall the random walk Metropolis-Hastings algorithm (algorithm 5.2) where we set $\mathbf{X} := \mathbf{X}^{(t-1)} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim g$. In this section we will generalise this to

$$\mathbf{X} = \boldsymbol{\tau}(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)}), \text{ with } \mathbf{U}^{(t-1)} \sim g_{1\to 2}.$$

For our further developments it will be necessary that the transformation is a bijective map, 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] 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 fulfils the detailed balance condition?

If we set the probability of acceptance to[2]

$$\alpha(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = \min\left\{1, \frac{f(\mathbf{x}^{(t)}) g_{2\to 1}(\mathbf{u}^{(t)})}{f(\mathbf{x}^{(t-1)}) g_{1\to 2}(\mathbf{u}^{(t-1)})} \left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})}{\partial(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})}\right|\right\},$$

then we can establish that detailed balance holds, as we will see below. 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

[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.

[2] In this lecture course we use the convention $|\mathbf{A}| = |\det(\mathbf{A})|$.

$$
\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 $\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})}{\partial(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)})}\right| \cdot \left|\frac{\partial \mathbf{T}_{2\to 1}(\mathbf{x}^{(t)}, \mathbf{u}^{(t)})}{\partial(\mathbf{x}^{(t)}, \mathbf{u}^{(t)})}\right| = 1$. Equation 6.1 implies by analogy with proposition 5.1 detailed balance[3], i.e. the Markov chain generated by the above method has indeed f as invariant distribution.

[3] 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 (6.1). (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 $(\mathbf{x}^{(t-1)}, \mathbf{u}^{(t-1)}) = \mathbf{T}_{2\to 1}(\mathbf{x}^{(t)}, \mathbf{u}^{(t)})$.)

Example 6.1 (Random walk Metropolis-Hastings). In order to clarify what we have just derived we will state the random walk Metropolis Hastings algorithm in terms of this new approach.
In the random walk Metropolis-Hastings 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 $\mathbf{T}_{2\to 1}(\mathbf{X}^{(t)}, \mathbf{U}^{(t)}) = (\mathbf{X}^{(t)} - \mathbf{U}^{(t)}, \mathbf{U}^{(t)})$.[4]
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

$$\left|\frac{\partial \mathbf{T}_{1\to 2}(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)})}{\partial(\mathbf{X}^{(t-1)}, \mathbf{U}^{(t-1)})}\right| = \begin{vmatrix} 1 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 1 \\ 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{vmatrix} = 1.$$

◁

Note that the above 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 the same dimension. Thus we can use the above approach for sampling from variable dimension models, as the next section shows.

6.3 The Reversible Jump Algorithm

Coming back to the developments of section 6.1 we need to draw an MCMC sample from the joint posterior distribution

$$f^{\text{post}}(k, \boldsymbol{\theta}) = \frac{p_k f_k^{\text{prior}}(\boldsymbol{\theta}) l_k(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta})}{\sum_\kappa p_\kappa \int_{\Theta_\kappa} l_\kappa^{\text{prior}}(\boldsymbol{\vartheta}) l_\kappa(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\vartheta})\, d\boldsymbol{\vartheta}} \tag{6.1}$$

defined on

$$\Theta := \bigcup_k (\{k\} \times \Theta_k)$$

A slight modification of the approach discussed in 6.2 allows us to draw samples from $f^{\text{post}}(k, \boldsymbol{\theta})$ by jumping between the models. This lea s to the reversible jump algorithm proposed by Green (1995):

[4] Due to the symmetry of $g_{1\to 2}$ this is equivalent 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.

Algorithm 6.1 (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 $\rho_{k^{(t-1)}\to k}$.[5]
   (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 $(\boldsymbol{\theta}, \mathbf{u}) := T_{k^{(t-1)}\to k}(\boldsymbol{\theta}^{(k-1)}, \mathbf{u}^{(k-1)})$.
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 $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}$, otherwise keep $k^{(t)} = k^{(t-1)}$ and $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)}$.

Note that as in section 6.2 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 6.1. The probability of acceptance of the reversible jump algorithm does not depend on the normalisation constant of the joint posterior $f^{(post)}(k, \boldsymbol{\theta})$ (i.e. the denominator of (6.1)).

Proposition 6.1. *The joint posterior $f^{post}(k, \boldsymbol{\theta})$ is under the above conditions the invariant distribution of the reversible jump algorithm.*

[5] $\sum_k \rho_{k^{(t-1)}\to k} = 1$.

Proof. From the footnote on page 65 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 6.2. 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 $D = \{(\theta_1, \theta_2) : 0 \le \theta_2 \le \theta_1 \le 1\}$. 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[6] 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 $U \sim \mathsf{U}[0,1]$ 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:[7] 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 $\mathsf{U}[0,1]$).

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 $\theta_1 \sim \mathsf{U}[0,1]$.
    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 $u^{(t-1)} \sim \mathsf{U}[0,1]$
    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
    $$\alpha = \min\left\{1, \frac{p_2 \cdot f_2^{\text{prior}}(\theta_1, \theta_2) l_2(\mathbf{y}_1, \ldots, \mathbf{y}_n | \theta_1, \theta_2) \cdot 1}{p_1 \cdot f_1^{\text{prior}}(\theta_1^{(t-1)}) l_1(\mathbf{y}_1, \ldots, \mathbf{y}_n | \theta_1^{(t-1)}) \cdot 1/2 \cdot 1} \cdot \theta_1^{(t-1)}\right\}$$

[6] in the sense that $\mathbf{T}_{1\to2} = \mathbf{T}_{2\to1}^{(-1)}$

[7] 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.

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 $\boldsymbol{\theta}^{(t)} = (\theta_1^{(t-1)}, \theta_2^{(t-1)})$. ◁

Example 6.3 (Mixture of Gaussians with a variable number of components). Consider again the Gaussian mixture model from example 4.6, 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.[8] In example 4.6 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 $\pi_{k+1} \sim \mathsf{Beta}(1, k)$. 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 $\mathsf{Beta}(1, k)$ 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

[8] In order to ensure identifiability we assume the μ_κ are ordered, i.e. $\mu_1 < \ldots < \mu_k$.

$$(1-u_1^{(t-1)})^k$$

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 parametrised, 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). ◁

7. Diagnosing convergence

7.1 Practical considerations

The theory of Markov chains we have seen in chapter 3 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 $\mathbb{E}_f(h(\mathbf{X}))$ by their the corresponding means

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

using the *entire* chain. In practise, 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 T_0 has to be chosen depends on how fast mixing the Markov chain $(\mathbf{X}^{(t)})_t$ is. Figure 7.1 illustrates the idea of a burn-in period.

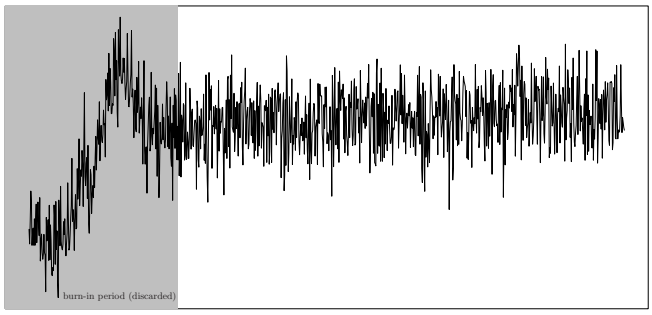


Fig. 7.1. Illustration of the idea of a burn-in period.

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 when the objective is estimating $\mathbb{E}_f(h(\mathbf{X}))$, as the following lemma shows.

Lemma 7.1. *Let $(\mathbf{X}^{(t)})_{t=1,\dots,T}$ be a sequence of random variables (e.g. from a Markov chain) with $\mathbf{X}^{(t)} \sim f$ and $(\mathbf{Y}^{(t)})_{t=1,\dots,\lfloor T/m\rfloor}$ a second sequence defined by $\mathbf{Y}^{(t)} := \mathbf{X}^{(m\cdot t)}$. If $\mathrm{Var}_f(h(\mathbf{X}^{(t)})) < +\infty$, then*

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

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

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

and

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

for $\tau_1, \tau_2 \in \{0,\dots,m-1\}$, we obtain that

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

Thus

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

□

The concept of thinning can be useful for other reasons. If the computer's memory cannot hold the entire chain $(\mathbf{X}^{(t)})_t$, thinning is a good choice. Further, it can be easier to assess the convergence of the thinned chain $(\mathbf{Y}^{(t)})_t$ as opposed to entire chain $(\mathbf{X}^{(t)})_t$.

7.2 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 practise.

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 nothing other than exploratory tools that help you judging 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 tasks of diagnosing different aspects of convergence:

Convergence to the target distribution. The first, and most important, question is whether $(\mathbf{X}^{(t)})_t$ yields a 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, and
- whether $(\mathbf{X}^{(t)})_t$ covers the entire support of the target distribution.

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

7.2.1 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), 4.5 (b) (c), 5.3 (a), and 5.4. 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 the plot seems (like for example figure 4.5), the slower mixing the resulting chain is.

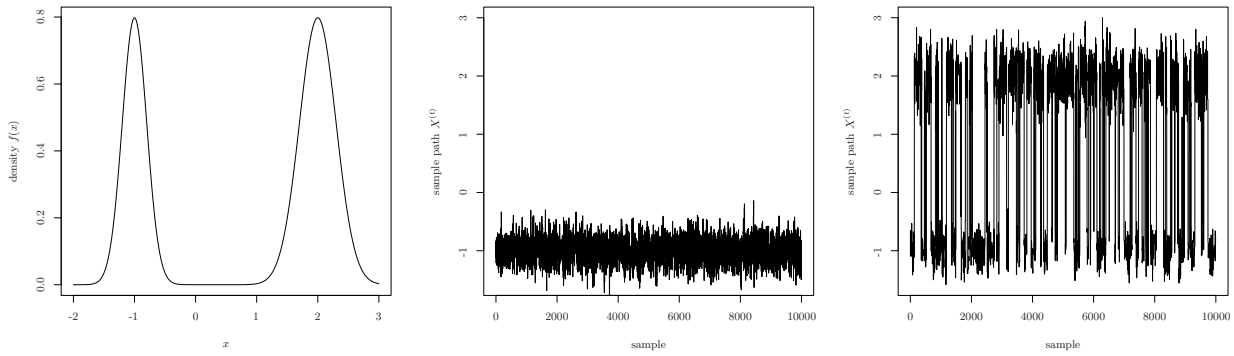
Note however that this plot suffers from the "you've only seen where you've been" problem. It is impossible to see from a plot of the sample path whether the chain has explored the entire support of the distribution.

Example 7.1 (A simple mixture of two Gaussians). In this example we sample 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 7.2 (a) for a plot of the density) using a random walk Metropolis algorithm with proposed value $X = X^{(t-1)} + \varepsilon$ with $\varepsilon \sim \mathsf{N}(0, \mathrm{Var}(\varepsilon))$. If we choose the proposal variance $\mathrm{Var}(\varepsilon)$ too small, we only sample from one population instead of both. Figure 7.2 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 7.2 (b) alone that the chain has not explored the entire support of the target. ◁

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



(a) Density $f(x)$

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

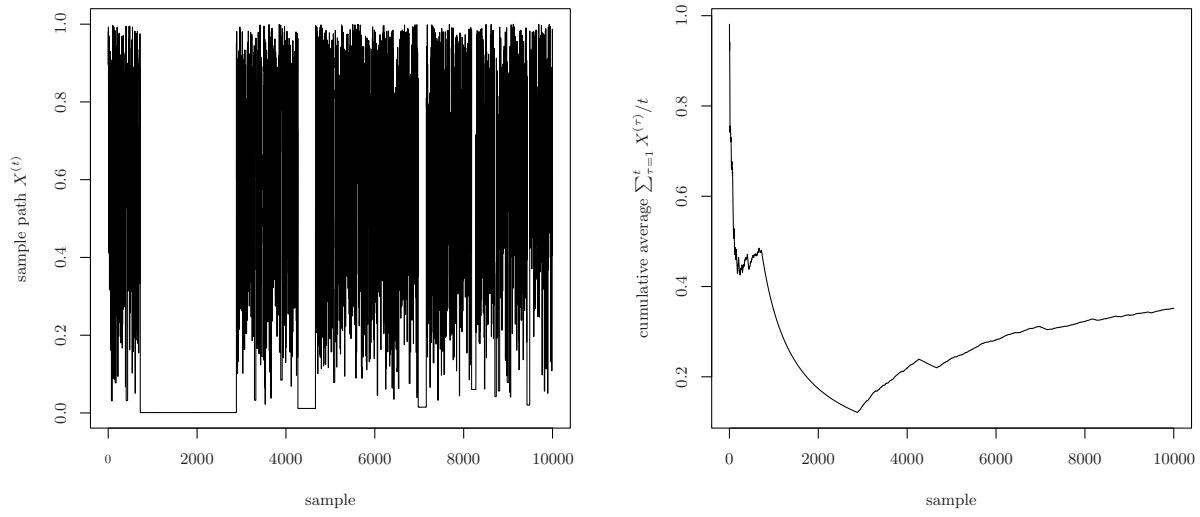
(c) Sample path of a random walk Metropolis algorithm with proposal variance $\mathrm{Var}(\varepsilon) = 1.2^2$

Fig. 7.2. Density of the mixture distribution with two random walk Metropolis samples using two different variances $\mathrm{Var}(\varepsilon)$ of the proposal.

Example 7.2 (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 $\mathsf{Beta}(\alpha, 1)$ 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 $X^{(t)} \sim \mathsf{Beta}(\alpha + 1, 1)$.

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 7.3 (b)). ◁



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

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

Fig. 7.3. 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.

7.2.2 Non-parametric tests of stationarity

This section presents the Kolmogorov-Smirnov test, which is an example of how nonparametric tests can be used as a tool for diagnosing whether a Markov chain has already converged.

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.

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. 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 [1]

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

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.[2] As long as a significant proportion of the values of the 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 7.3 (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 7.4 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. ◁

Note that 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 7.1 would not reveal anything unusual.

7.2.3 Riemann sums and control variates

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

[1] 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

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

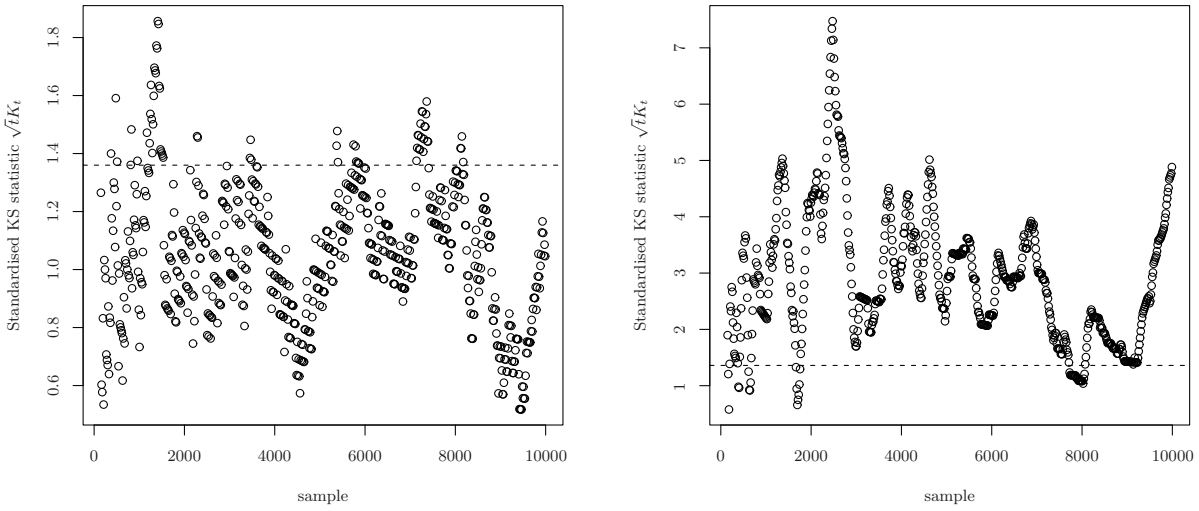
The Kolmogorov-Smirnov test statistic is the maximum difference between the two empirical CDFs:

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

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

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

[2] K_t is hereby the Kolmogorov-Smirnov statistic obtained from the sample consisting of the first t observations only.



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

Fig. 7.4. Standardised Kolmogorov-Smirnov statistic for $X_1^{(5 \cdot t)}$ from the Gibbs sampler from the bivariate Gaussian for two different correlations.

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

We can estimate this integral by the Riemann sum

$$\sum_{t=2}^{T} (X^{[t]} - X^{[t-1]}) f(X^{[t]}), \tag{7.1}$$

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 (7.1) should be around 1. 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.

Example 7.4 (A simple mixture of two Gaussians (continued)). In example 7.1 we considered two random-walk Metropolis algorithms: one ($\text{Var}(\varepsilon) = 0.4^2$) failed to explore the entire support of the target distribution, whereas the other one ($\text{Var}(\varepsilon) = 1.2^2$) 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 comparing several ways of estimating the same quantity. As long as 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 (7.1).

7.2.4 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[3] starting values $\mathbf{X}^{(1,0)}, \ldots, \mathbf{X}^{(L,0)}$, covering at least the support of the target distribution.

All L chains should converge to the same distribution, so comparing the plots from section 7.2.1 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.

[3] i.e. the variance of the starting values should be larger than the variance of the target distribution.

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 $\sum_{l=1}^{L} \delta_\alpha^{(L,\cdot)}/L$, where $\delta_\alpha^{(L,\cdot)}$ is the distance between the α and $(1-\alpha)$ 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 α and $(1-\alpha)$ 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

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

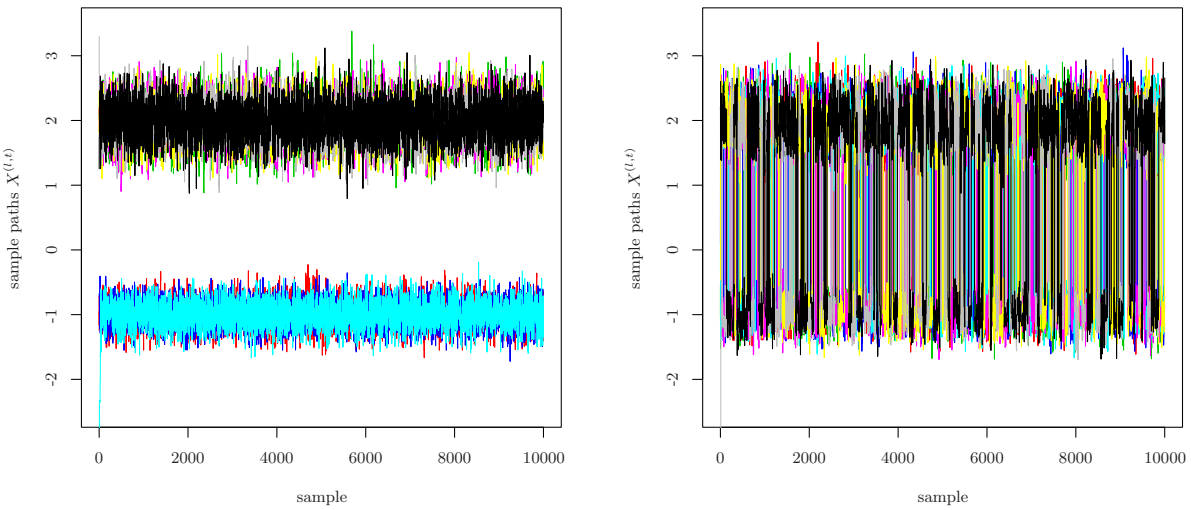
can be used as a tool to diagnose whether all chains sampled from the same distribution, in which case the ratio should be around 1.

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 7.5 (A simple mixture of two Gaussians (continued)). In the example of the mixture of two Gaussians we will consider $L = 8$ chains initialised from a $\mathsf{N}(0, 10^2)$ distribution. Figure 7.5 shows the sample paths of the 8 chains for both choices of $\text{Var}(\varepsilon)$. The corresponding values of $\hat{S}_{0.05}^{\text{interval}}$ are:

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

◁



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

Fig. 7.5. 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 7.6 (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 7.6 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

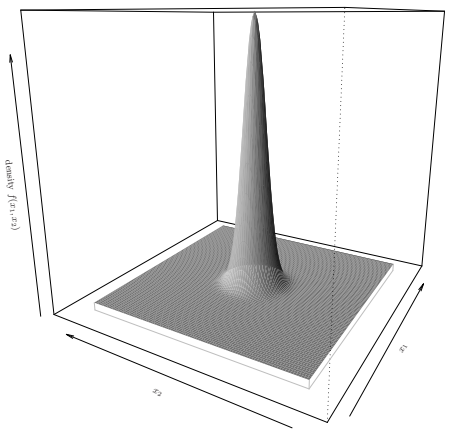
$$f(x_1|x_2) = \begin{cases} (1-\delta_{x_2})\phi_{(\boldsymbol{\mu},\sigma^2\cdot\mathbb{I})}(x_1,x_2) + \delta_{x_2} & \text{for } x_1 \in (0,1) \\ 0 & \text{else.} \end{cases}$$

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

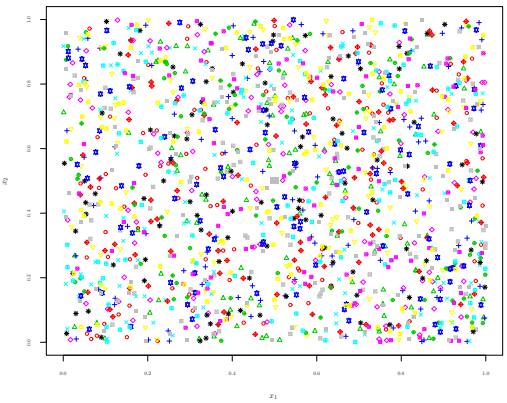
Assume we want to estimate $\mathbb{P}(0.49 < X_1, X_2 \leq 0.51)$ for $\delta = 10^{-3}$, $\boldsymbol{\mu} = (0.5, 0.5)'$, and $\sigma = 10^{-5}$ using a Gibbs sampler. Note that 99.9% of the mass of the distribution is concentrated in a very small area around $(0.5, 0.5)$, i.e. $\mathbb{P}(0.49 <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

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 7.6 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 7.6 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 $\sigma = 10^{-5}$)

Fig. 7.6. Density and sample from the witch's hat distribution.

7.2.5 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

$$\mathrm{Var}\left(\frac{1}{T}\sum_{t=1}^{T} h(\mathbf{X}^{(t)})\right) \tag{7.2}$$

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

The effective sample size (ESS) allows to quantify this loss of information caused by the positive correlation. The effective sample size is the size an i.i.d. would have to have in order to obtain the same variance (7.2) as the estimate from the Markov chain $(\mathbf{X}^{(t)})_{t=1,\ldots,T}$.

In order to compute the variance (7.2) we make the simplifying assumption that $(h(\mathbf{X}^{(t)}))_{t=1,\ldots,T}$ is from a second-order stationary time series, i.e. $\mathrm{Var}(h(\mathbf{X}^{(t)})) = \sigma^2$, and $\rho(h(\mathbf{X}^{(t)}), h(\mathbf{X}^{(t+\tau)})) = \rho(\tau)$. Then

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

If $\sum_{\tau=1}^{+\infty}|\rho(\tau)| < +\infty$, then we can obtain from the dominated convergence theorem[4] that

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

as $T\to\infty$. Note that the variance 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 $(h(\mathbf{X}^{(t)}))_{t=1,\ldots,T}$ is a first-order autoregressive time series (AR(1)), i.e. $\rho(\tau) = \rho(h(\mathbf{X}^{(t)}), h(\mathbf{X}^{(t+\tau)})) = \rho^{|\tau|}$, 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 7.7 (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:

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

[4] see e.g. Brockwell and Davis (1991, theorem 7.1.1) for details.

8. Simulated Annealing

8.1 A Monte-Carlo method for finding the mode of a distribution

So far we have studied various methods that allow for appoximating expectations $\mathbb{E}(h(\mathbf{X}))$ by ergodic averages $\frac{1}{T}\sum_{t=1}^{T} h(\mathbf{X}_i^{(t)})$. This section presents an algorithm for finding the (global) mode(s) of a distribution[1]. In section 8.2 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 8.1 (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 8.1 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 8.2 for an example). It is instructive to see informally why this is the case when considering a discrete random variable with probability density 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

$$p_{(\beta)}(x) = \frac{(p(x))^{\beta}}{\sum_{x\in E^*}(p(x))^{\beta} + \sum_{x\in E\setminus E^*}(p(x))^{\beta}} = \frac{(p(x)/m)^{\beta}}{\sum_{x\in E^*} 1 + \sum_{x\in E\setminus E^*}(p(x)/m)^{\beta}} \overset{\beta\to+\infty}{\longrightarrow} \begin{cases} 1/|E^*| & \text{if } x \in E^* \\ 0 & \text{if} x \notin E^* \end{cases}$$

[1] In this chapter we define the mode(s) of a distribution to be the set of global maxima of the density, i.e. $\{\boldsymbol{\xi} : \ f(\boldsymbol{\xi}) \geq f(\mathbf{x}) \ \forall \mathbf{x}\}$

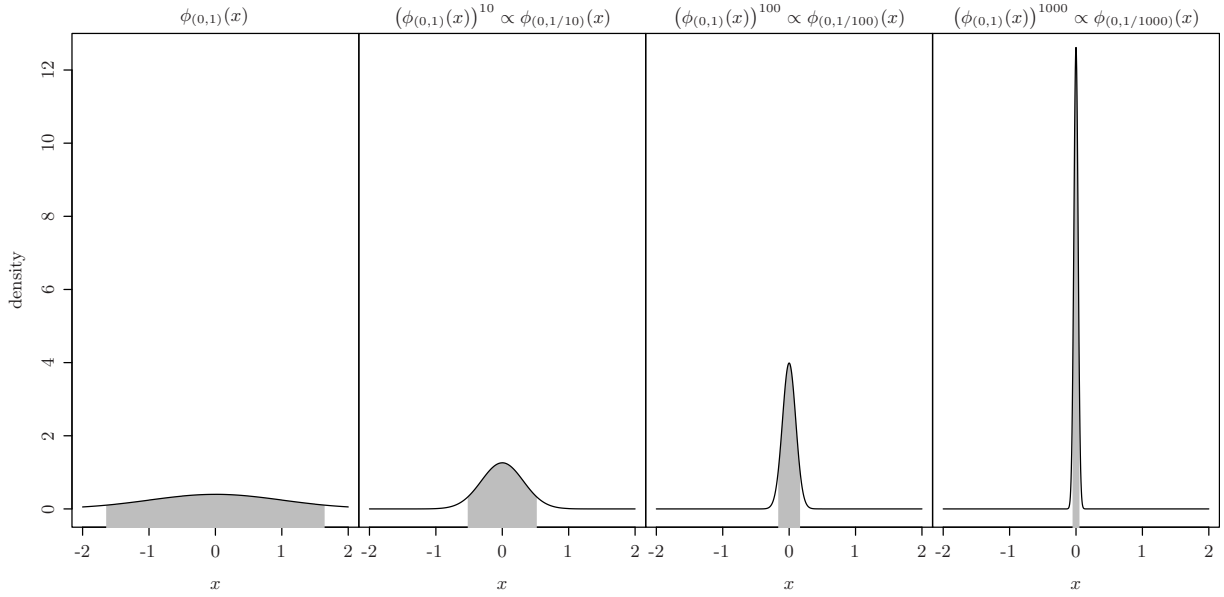


Fig. 8.1. Density of the $\mathsf{N}(0,1)$ raised to increasing powers. The areas shaded in grey represent 90% of the probability mass.

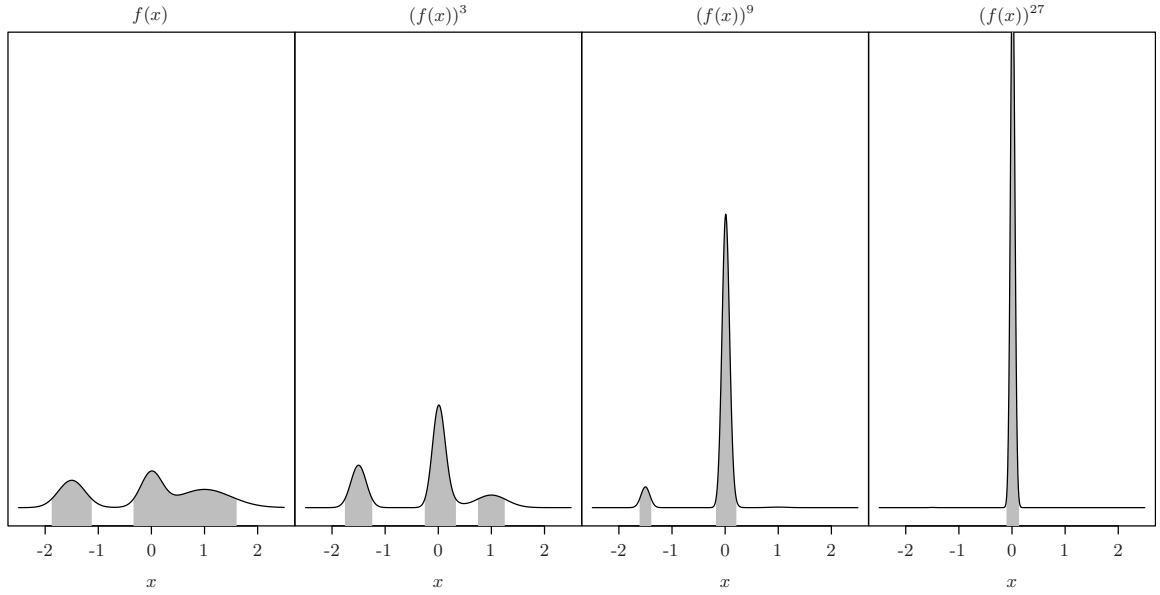


Fig. 8.2. An arbitrary multimodal density raised to increasing powers. The areas shaded in grey reach from the 5% to the 95% quantiles.

In the continuous case the distribution is not uniform on the nodes (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 8.2. 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 8.3 visualises 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 $\beta \to +\infty$. ◁

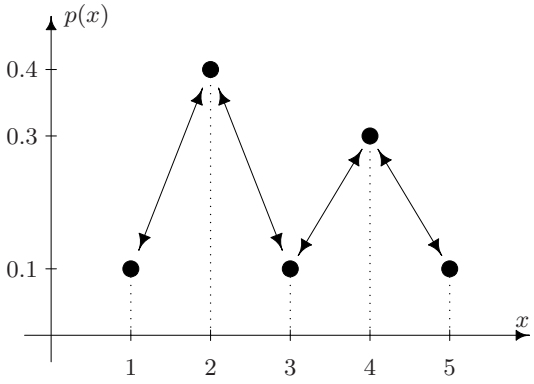


Fig. 8.3. Illustration of example 8.2

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[2].

[2] 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[3] (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.

8.2 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.[4]. 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 8.1 (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 and logarithmic tempering with a suitably large β_0 is used.

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

[3] The term *annealing* comes from metallurgy and refers to the technique of letting metal cool down slowly in order to produce a tougher metal. Taking up this analogy, $1/\beta$ is typically referred to as temperature, β as inverse temperature.

[4] In this framework, finding the mode of a density f corresponds to finding the minimum of $-\log(f(x))$

$$\mathbb{P}(X^{(t)} = x) = \sum_{\xi} \underbrace{\mathbb{P}(X^{(t)} = x | X^{(t-1)} = \xi)}_{\geq \exp(-\beta_{t-1}\Delta H)/|E|} \mathbb{P}(X^{(t-1)} = \xi) \geq \exp(-\beta_{t-1}\Delta H)/|E|$$

Using the logarithmic tempering schedule we obtain $\mathbb{P}(X^{(t)} = x) \geq t/|E|$ and thus the expected number of visits to state x is

$$\sum_{t=0}^{\infty} \mathbb{P}(X^{(t)} = x) \geq \sum_{t=0}^{\infty} t/|E| = +\infty.$$

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 8.3. Assume we want to find the maximum of the function

$$H(x) = \left((x-1)^2 - 1\right)^2 + 3 \cdot s(11.56 \cdot x^2), \text{ with } s(x) = \begin{cases} |x| \mod 2 & \text{for } 2k \leq |x| \leq 2k+1 \\ 2 - |x| \mod 2 & \text{for } 2k+1 \leq |x| \leq 2(k+1) \end{cases}$$

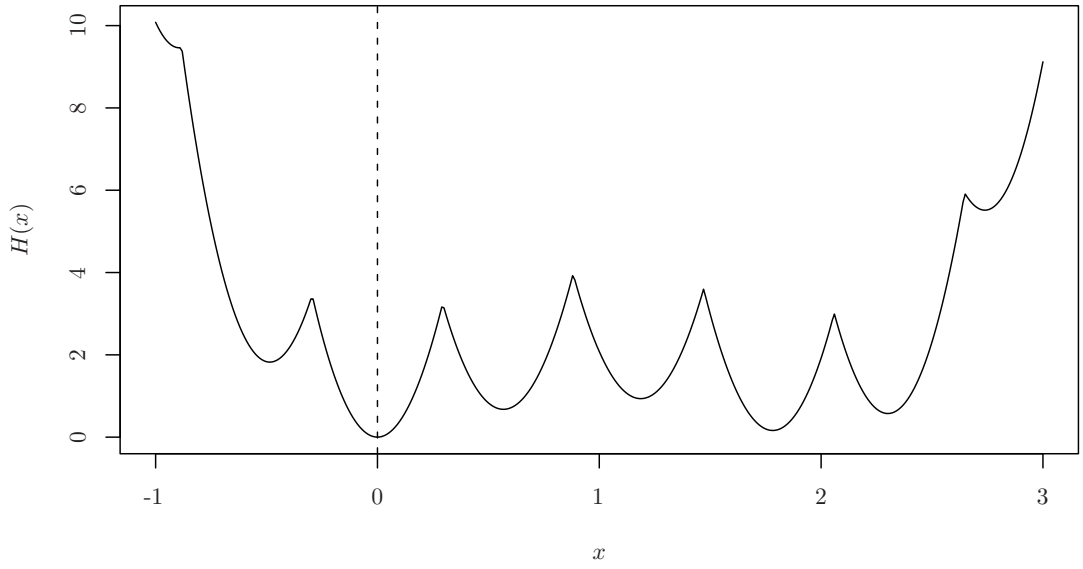
for $k \in \mathbb{N}_0$. Figure 8.4 (a) shows $H(x)$ for $x \in [-1, 3]$. The global minimum of $H(x)$ is at $x = 0$. We simulated annealing with a geometric tempering with $\beta_0 = 1$ and $\beta_t = 1.001\beta_{t-1}$ and a random walk Metropolis algorithm with $\varepsilon \sim \mathsf{Cauchy}(0, \sqrt{0.1})$. Figure 8.4 (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 practise, it would unrealistic to expect simulated annealing to converge to a *global* minimum, however in most cases it will find a "good" *local* minimum.

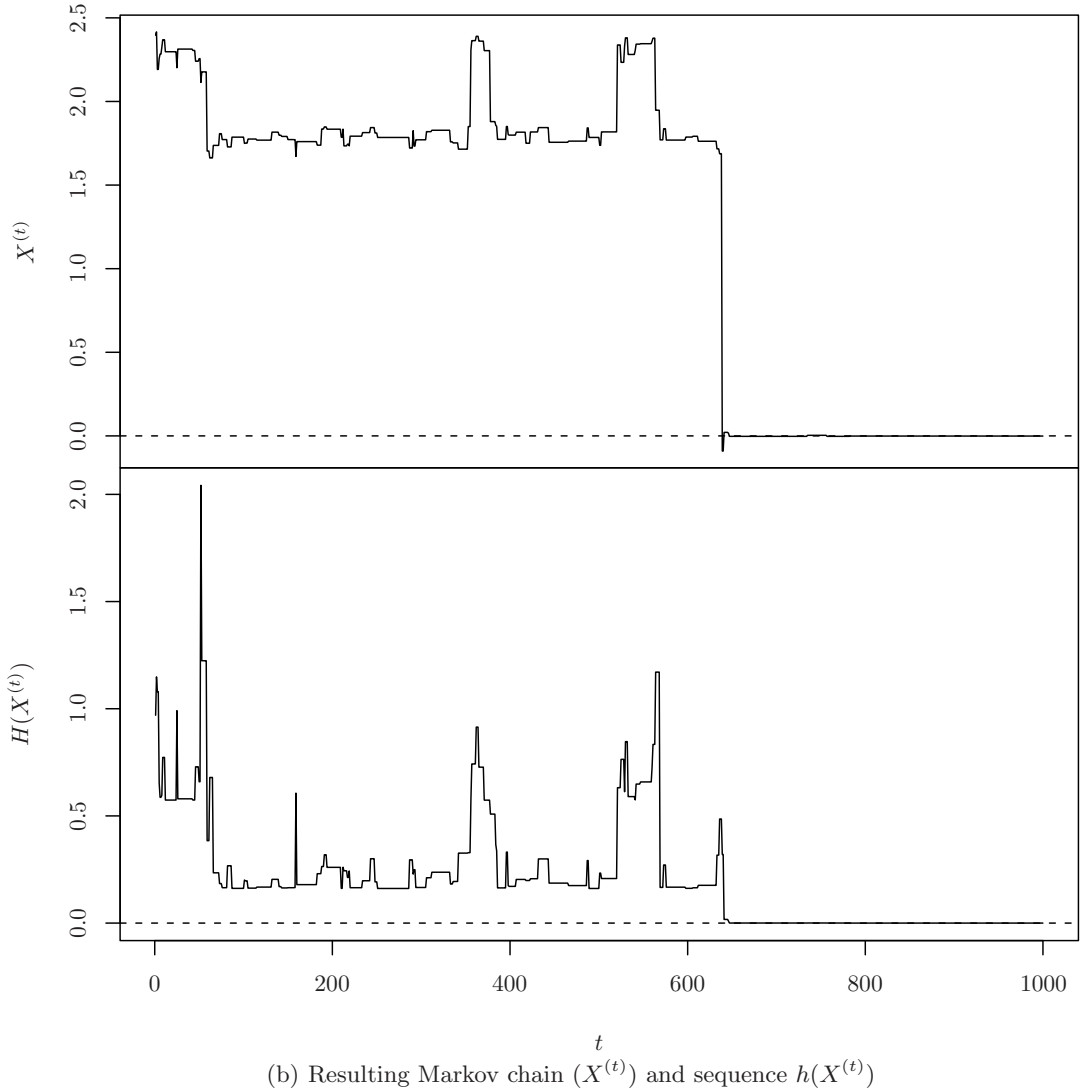
8.3 Using annealing strategies for imporving the convergence of MCMC algorithms

As we have seen in the previous chapter, 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 $f_{(\beta)}(x) = (f(x))^{(\beta)}$ 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. $\beta = 1$). One way to accommodate this is to consider an *ensemble* of distributions $(f_{(\beta_1)}(\cdot), \ldots, f_{(\beta_r)}(\cdot))$ with $\beta_1 \leq \ldots \leq \beta_r = 1$, and draw Markov chains from each member of the ensemble. The key idea is to let these chains "interact" by swapping values between neighbouring



(a) Objective function



(b) Resulting Markov chain $(X^{(t)})$ and sequence $h(X^{(t)})$

Fig. 8.4. Objective function $H(x)$ from example 8.3 and first 1,000 iterations of the Markov chain yielded by simulated annealing.

members, which can be formalised as a Metropolis Hastings algorithm on the augmented distribution $f(x_1, \ldots, x_r) = \prod_{\rho=1}^{r} f_{(\beta_\rho)}(x_\rho)$. The distribution of interest is the marginal distribution of X_r, which has $f(\cdot)$ as distribution. This setups allows members with small β to "help" members with large β to cross barriers of low probability, and thus can considerably improve the mixing properties. Chapter 10 of (Liu, 2001) gives a more detailed overview over such approaches.

9. Hidden Markov Models

Hidden Markov Models (HMMs) are a broad class of models which have found wide applicability in fields as diverse as bioinformatics and engineering. We introduce them briefly here as a broad class of Monte Carlo algorithms have been developed to deal with inference in models of this sort. HMMs will be covered in detail in the *Graphical Models* lecture course; this introduction is far from complete and is intended only to provide the background essential to understand those Monte Carlo techniques. An enormous amount of research has gone into inference in models of this sort, and is ongoing today. Indeed, an extensive monograph summarising the field was published recently (Cappé et al., 2005).

We shall take HMMs to be a class of models in which an underlying Markov process, (X_t) (usually termed either the *state* or *signal* process) forms the object of inference, whilst a related process, (Y_n) (usually known as the observation process), is observed and provides us with information about the process of interest. In order for such a pair of processes to form an HMM, they must have the *conditional independence properties* illustrated in figure 9.1. That is, the state process is a Markov chain so the distribution of X_n is, conditional upon X_{n-1} independent of all previous values of X and the distribution of each element of the observation process depends only upon the value of the state process at that time (more formally, $\mathbb{P}(Y_n \in A | X_{1:n}, Y_{1:n-1}) \equiv \mathbb{P}(Y_n \in A | X_n)$).

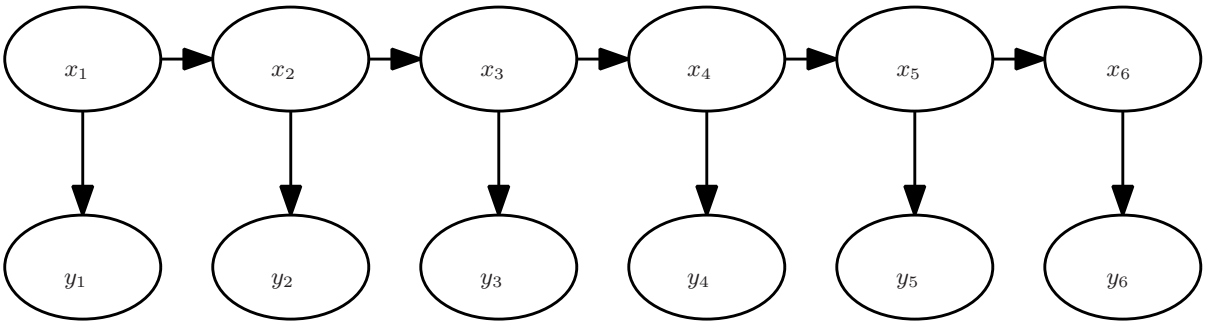


Fig. 9.1. The conditional independence structure of the first few states and observations in a Hidden Markov Model.

The term State-Space Model (SSM) is more popular in some application areas. Sometimes a distinction is made between the two, with HMMs being confined to the class of models in which the hidden states take values in a discrete set and SSMs including the case in which the hidden process takes values in

a continuous space (that is, whether the underlying Markov chain lies in a discrete or continuous state space); we will not make this distinction here.

It seems clear that a great many different problems in statistics, signal processing and related areas can be cast into a form based upon the HMM. Essentially, whenever one has a problem in which one wishes to perform inference as observations arrive (typically in real time) this is the first class of models which one considers using. It is a tribute to the flexibility and descriptive power of such models that it is relatively unusual for a more complex model to be required in order to capture the key properties of imperfectly-observed discrete time processes.

Before considering the particular inferential problems which occur most frequently when dealing with such models, it is worth noticing the particular feature of these models which distinguishes estimation and inference tasks from those considered more generally in the statistics literature, and which makes the application of standard techniques such as MCMC inappropriate much of the time. That feature is the particular temporal character of the models, and those things which are typically inferred in them. The model is designed to capture the essence of systems which move from one state to another, generating an observation after each move. Typically, one receives the observations generated by these systems in real time and wishes to estimate the state or other quantities in real time: and this imposes the particular requirement that the computational cost of the estimates remains fixed and is not a function of the size of the data set (otherwise increasing computational power will be required every time an observation is received). It is for this reason that the HMM is considered here, and a collection of popular Monte Carlo techniques for performing inference in systems of this type will be introduced in chapter 10.

In order to illustrate the broad applicability of this framework, and to provide some intuition into the nature of the model, it is interesting to consider some examples.

9.1 Examples

9.1.1 Tracking

Arguably the canonical example of the HMM is *tracking*. The states in the HMM comprise a vector of coordinates (which may include information such as velocity and acceleration as well as physical position, or additional information such as which type of object it is) for some object. The transitions of the Markov chain correspond to a dynamic model for that object, and the distribution of the observations conditional upon the state vector correspond to the measurement model encompassing systematic and random errors. As a definite example, consider something known as the the two dimensional (almost) constant velocity model. The state vector contains the position and velocity of an object in two dimension:

$$X_t = \begin{bmatrix} s_x \\ s_y \\ u_x \\ u_y \end{bmatrix} \qquad X_{t+1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X_t + W_t$$

where Δt is the measurement interval and W_t is an independent random variable which allows for variation in velocity. An additive error model is typically assumed in this type of application, so, $Y_t = X_t + V_t$ where V_t is an independent random variable with a distribution appropriate to describe measurement noise.

9.1.2 Statistical Signal Processing

Much of modern signal processing is model based, and estimates of signal parameters are obtained using statistical methods rather than more traditional signal processing techniques. One example might be attempting to infer an audio signal given a model for the time sequence X_t corresponding to audio amplitude and a measurement model which allows for a number of forms of noise. There are innumerable occurrences of models of this type.

9.2 State Estimation: Optimal Filtering, Prediction and Smoothing

Perhaps the most common inference problem which occurs with HMMs is the estimation of the current state value (or the sequence of states up to the present time) based upon the sequence of observations observed so far.

Bayesian approaches are typically employed as they provide a natural and flexible approach to the problem. In such an approach, one attempts to obtain the conditional distribution of the state variables given the observations. It is convenient to assume that both the state transition (i.e. the Markov kernel) and the conditional distributions of the measurements (the likelihood) admit densities with respect to a suitable version of Lebesgue measure. Writing the density of X_{t+1} conditional upon knowledge that $X_t = x_t$ as $f_{t+1}(\cdot|x_t)$ and that of the likelihood as $g_{t+1}(y_{t+1}|\cdot)$, and interpreting $p(x_{1:1}|y_{1:0})$ as some prior distribution $p(x_1)$, we may write the density of the distribution of interest via Bayes rule in the following form:

$$\begin{aligned} p(x_{1:t}|y_{1:t}) &\propto p(x_{1:t}|y_{1:t-1})g_t(y_t|x_t) \\ &= p(x_{1:t-1}|y_{1:t-1})f_t(x_t|x_{t-1})g_t(y_t|x_t) \end{aligned}$$

This problem is known as filtering and $p(x_{1:t}|y_{1:t})$ is known as the *smoothing* distribution. In some literature this distribution is known as the *filtering* distribution, here that term is reserved for the final time marginal $p(x_t|y_{1:t})$. It is usual to decompose this recursive solution into two parts. The first is termed *prediction* and corresponds to the estimation of the distribution of the first n states given only $n-1$ observations. The second is usually termed the *update step* (or sometimes, the data update step) and it involves correcting the predicted distribution to take into account the next observation:

$$\begin{aligned} p(x_{1:t}|y_{1:t-1}) &= p(x_{1:t-1}|y_{1:t-1})f_t(x_t|x_{t-1}) && \text{Prediction} \\ p(x_{1:t}|y_{1:t}) &= \frac{p(x_{1:t}|y_{1:t-1})g_t(y_t|x_t)}{\int p(x_{1:t}|y_{1:t-1})g_t(y_t|x_t)dx_{1:t}} && \text{Update.} \end{aligned}$$

Certain other distributions are also of interest. These can be divided into smoothing distributions, in which one wishes to estimate the distribution of some sequence of states conditional on knowledge of the observation sequence up to some stage in the future, and prediction distributions, in which one wishes to estimate the distribution of some group of future states conditional on knowledge of the observations up to the present.

Formally, smoothing can be considered to be the estimation of $p(x_{l:k}|y_{1:t})$ when $l \leq k \leq n$ and such estimates can all be obtained from the principle smoothing distribution:

$$p(x_{l:k}|y_{1:t}) = \int p(x_{1:t}|y_{1:t})dx_{1:l-1}dx_{k+1:t}.$$

Similarly, prediction can be viewed as the estimation of $p(x_{l:k}|y_{1:t})$ when $n \leq k$ and $l \leq k$. Noticing that when $k \geq n$,

$$p(x_{1:k}|y_{1:t}) = p(x_{1:t}|y_{1:t}) \prod_{j=t+1}^{k} f_j(x_j|x_{j-1})$$

it is analytically straightforward to obtain any prediction density by marginalisation (that is integration over the variables which are not of interest) of these densities.

Whilst this appears to be a solution to the problem of estimating the distributions of the state variables, and hence of estimating those parameters using either the posterior mean or the maximum a posteriori estimator, in practice the problem is far from being solved. The integrals which appear in the update step in particular is generally intractable.

In the linear Gaussian case (in which the distribution of $X_t|X_{t-1}$ and $Y_t|X_t$ can both be treated as Gaussian distributions centred at a point obtained by a linear transformation of the conditioning variable) it is possible to perform the integral analytically and hence to obtain closed form expressions for the distribution of interest. This leads to the widely used *Kalman filter* which is applicable to an important but restricted collection of problems. Various approaches to obtain approximate solutions by extensions of this approach in various ways (propagating mixtures of Gaussians or using local linearisation techniques, for example) are common in the engineering literature. In small discrete state spaces the integral is replaced with a summation and it becomes formally possible to exactly evaluate the distributions of interest. Indeed, a number of algorithms for performing inference efficiently in discrete HMMs do exist, but these tend to become impractical when the state space is large.

The most common approach to dealing with this estimation problem in general models is via a Monte Carlo technique which is often termed the *particle filter* but which is referred to in this course by the more general term of *sequential Monte Carlo*. These techniques will be introduced in the next chapter. The approach has a simple motivation. We wish to calculate, recursively in time, a sequence of distributions each of which can be obtained from the previous distribution if it is possible to carry out particular integrals with respect to that distribution. In order to approximate the recursion, it is possible to approximate each distribution with a collection of weighted samples and, using this collection to calculate the necessary integrals, to propagate this collection of samples through time in such a way that it approximates each of the distributions of interest one after another.

9.3 Static Parameter Estimation

Thus far, it has been assumed that the system dynamics and the measurement system are both exactly characterised. That is, it has been assumed that given x_t, the distributions of X_{t+1} and Y_t are known. In practice, in many situations this will not be the case. As in any inference problem, it is necessary to have some idea of the structure of the system in which inference is being performed, and the usual approach to incorporating some degree of uncertainty is the introduction of some unknown parameters. If the system dynamics or measurement distributions depend upon some collection of unknown parameters θ, then the problem of state estimation, conditional upon a particular value of those parameters is precisely that introduced in section 9.2.

If θ is not known then Bayesian inference is centred around the approximation of the sequence of distributions:

$$p(x_{1:t}, \theta|y_{1:t}) \propto p(\theta)p(x_{1:t}|\theta)p(y_{1:t}|x_{1:t}, \theta).$$

In some instances, the latent state sequence is not of statistical interest and one is really interested in approximation of the marginal distributions:

$$p(\theta|y_{1:t}) = \int p(x_{1:t}, \theta|y_{1:t}) dx_{1:t},$$

however, this integral is generally intractable and it is typically necessary to obtain Monte Carlo estimates of $p(x_{1:t}, \theta|y_{1:t})$ and to use the approximation to $p(\theta|y_{1:t})$ provided by these samples.

Estimation of static parameters appears a relatively simple problem, and one which does not much increase the dimensionality of the space of interest. Indeed, in scenarios in which one is presented with a fixed quantity of data MCMC algorithms are able to provide approximation to the distribution of interest this is the case – numerous algorithms for estimating parameters in latent variable models by various mechanisms exist and can be shown to work well. The difficulty is that inference problems of practical interest generally involve sequential estimation of the value of the parameters based upon the collection of observations available at that time. In order to do this, it is necessary to develop an efficient mechanism for updating the estimate of $p(\theta, x_{1:t-1}|y_{1:t})$ to provide an estimate of $p(\theta, x_{1:t}|y_{1:t})$.

10. Sequential Monte Carlo

This chapter is concerned with techniques for iteratively obtaining samples from sequences of distributions by employing importance sampling, and resampling, techniques. The principle application of these techniques is the approximate solution of the filtering, prediction and smoothing problems in HMMs (see chapter 9).

10.1 Importance Sampling Revisited

Recall from section 2.3, that importance sampling is a technique for approximating integrals under one probability distribution using a collection of samples from another, instrumental distribution. This was presented using the importance sampling identity that, given a distribution of interest π, and some sampling distribution μ, over some space E, and any integrable function $h : E \to \mathbb{R}$

$$\mathbb{E}_\pi(h(X)) = \int \pi(x)h(x)\,dx = \int \mu(x) \underbrace{\frac{\pi(x)}{\mu(x)}}_{=:w(x)} h(x)\,dx = \int \mu(x)w(x)h(x)\,dx = \mathbb{E}_\mu(w(X)\cdot h(X)), \quad (10.1)$$

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

The strong law of large numbers was then employed to verify that, given a collection of iid samples, $\{X_i\}_{i=1}^N$, distributed according to μ, the empirical average of $w \cdot h$ evaluated at the sampled values X_i converges, with probability 1, to the integral of h under the target distribution, π.

10.1.1 Empirical Distributions

It is convenient to introduce the concept of random distributions in order to consider some of these things in further detail and to motivate much of the material which follows. If we have a collection of points $\{x_i\}_{i=1}^N$, in E, then we may associated the following distribution over E with those points:

$$\eta^N(x) = \frac{1}{N}\sum_{i=1}^N \delta_{x_i}(x),$$

where, for any point $x \in E$, δ_x is the singular distribution which places all of its mass at that point. A rigorous treatment requires measure-theoretic arguments, but is is possible – albeit not entirely rigorous – to define this distribution in terms of its integrals: for any bounded measurable function:, $h : E \to \mathbb{R}$,

$$\int h(y)\delta_x(y) = h(x).$$

Given a collection of points and associated positive, real-valued weights, $\{x_i, w_i\}_{i=1}^N$, we can also define another empirical distribution:

$$\tilde{\eta}^N(x) = \frac{\sum_{i=1}^N w_i \delta_{x_i}(x)}{\sum_{i=1}^N w_i},$$

where the normalising term may be omitted if the collection of weights is, itself, properly normalised.

When those collections of points are actually random samples, and the weights are themselves random variables (typically deterministic functions of the points with which they are associated), one obtains a random distribution (or random measure). These are complex structures, a full technical analysis of which falls somewhat outside the scope of this course, but we will make some limited use of them here.

Within this framework, one could consider, for example, the importance sampling identity which was introduced above in terms of the empirical measure $\mu^N(x)$ associated with the collection of samples from μ; we can approximate integrals with respect to μ by integrals with respect to the associated empirical measure:

$$\int h(x)\mu(x)dx \approx \int h(x)\mu^N(x)dx.$$

This approximation is justified by asymptotic results which hold as the number of samples tends to infinity, particularly the law of large numbers and related results, including variants of the Glivenko-Cantelli theorem.

If we consider propagating this approximation through (10.1), then we find that:

$$\begin{aligned}
\mathbb{E}_\pi(h(X)) = \int \pi(x)h(x)\,dx \quad &= \int \mu(x)\frac{\pi(x)}{\mu(x)}h(x)\,dx = \int \mu(x)w(x)h(x)\,dx \quad \approx \mathbb{E}_{\mu^N}(w(X)\cdot h(X)) \\
\mathbb{E}_\pi(h(X)) = \int \pi(x)h(x)\,dx \quad &= \int \mu(x)\frac{\pi(x)}{\mu(x)}h(x)\,dx \approx \int \mu^N(x)w(x)h(x)\,dx \quad = \mathbb{E}_{\mu^N}(w(X)\cdot h(X)) \\
\mathbb{E}_\pi(h(X)) = \int \pi(x)h(x)\,dx \quad &\approx \int \underbrace{\mu^N(x)\frac{\pi(x)}{\mu(x)}}_{=:\pi^N(x)} h(x)\,dx = \int \mu^N(x)w(x)h(x)\,dx \quad = \mathbb{E}_{\mu^N}(w(X)\cdot h(X)).
\end{aligned}$$

Perhaps unsurprisingly, the approximation amounts, precisely, to approximating the target distribution, π, with the random distribution

$$\pi^N(x) = \sum_{i=1}^N W_i \delta_{X_i}(x),$$

where $W_i := w(X_i) = \pi(X_i)/\mu(X_i)$ are importance weights. Note that we have assumed that the importance weights are properly normalised here; the self-normalised version of importance sampling can be represented in essentially the same way.

This illustrates a particularly useful property of the empirical distribution approach: it is often possible to make precise the nature of the approximation which is employed at one step of an algorithm and then to simply employ the empirical distribution in place of the true distribution which it approximates and in so doing to obtain the corresponding approximation at the next step of the algorithm. This property will become clearer later, when some examples have been seen.

10.1.2 HMMs

The remainder of this chapter is concerned with inference in HMMs, carried out by propagating forward an approximation to the distributions of interest using techniques based upon importance sampling. As with many Monte Carlo methods, it is often simpler to interpret the methods presented here as techniques

for approximating *distributions* of interest, rather than particular integrals. That is, we shall view these approaches as algorithms which approximate the filtering distributions $p(x_n|y_{1:n})$ directly, rather than approaches to approximate $\int h(x_n)p(x_n|y_{1:n})dx_n$. These algorithms will have a common character: they will each involve propagating a weighted empirical distribution forward through time according to some sampling scheme in such a way that at each time it provides an approximation to the distributions of interest.

Although many of the techniques which follow are traditionally presented as being related but distinct methods, it should be understood that it is typically straightforward to combine elements of the various different approaches to obtain a single cohesive algorithm which works well in a particular setting. Indeed, it is only in relatively simple settings that any one of the basic techniques presented below work well, and it is the combination of elements of each of these approaches which is essential to obtain algorithms which perform well in interesting problems.

It will be convenient to refer to the smoothing and filtering distributions of interest using the notation:

$$\begin{aligned}
\pi_t(x_{1:t}) &:= p(x_{1:t}|y_{1:t-1}) \\
\hat{\pi}_t(x_{1:t}) &:= p(x_{1:t}|y_{1:t}) \\
\pi_t(x_t) &:= p(x_t|y_{1:t-1}) &&= \int \pi_t(x_{1:t})dx_{1:t-1} \\
\hat{\pi}_t(x_t) &:= p(x_t|y_{1:t}) &&= \int \hat{\pi}_t(x_{1:t})dx_{1:t-1}.
\end{aligned}$$

10.2 Sequential Importance Sampling

The basis of sequential importance sampling techniques, is the following idea which allows importance samples to be obtained from a sequence of distributions defined on increasing spaces to be obtained iteratively using a single collection of samples, which are termed *particles*. Given a sequence of spaces, $E_1, E_2, \ldots$ and a sequence of distributions π_1 on E_1, π_2 on $E_1 \times E_2$, ... each of which is to be approximated via importance sampling, it is useful to make use of the following. Consider only the first two distributions in the first instance. Given a function $h_2 : E_1 \times E_2 \to \mathbb{R}$, beginning with the standard importance sampling identity:

$$\begin{aligned}
\int h(x_1, x_2)\pi_2(x_1, x_2)dx_1dx_2 &= \int h(x_1, x_2)\frac{\pi_2(x_1, x_2)}{\mu(x_1, x_2)}\mu(x_1, x_2)dx_1dx_2 \\
&= \int h(x_1, x_2)\frac{\pi_2(x_1, x_2)}{\mu(x_2|x_1)\mu(x_1)}\mu(x_1)\mu(x_2|x_1)dx_1dx_2 \\
&= \int h(x_1, x_2)\frac{\pi_2(x_1, x_2)}{\mu(x_2|x_1)\pi_1(x_1)}\frac{\pi_1(x_1)}{\mu(x_1)}\mu(x_1)\mu(x_2|x_1)dx_1dx_2
\end{aligned}$$

The last equality illustrates the key point: by decomposing the importance weight into two components in this manner, we are able to perform our calculations sequentially. We first draw a collection of samples $\{X_1^{(i)}\}_{i=1}^N$ according to a distribution μ and then set their importance weights $W_1^{(i)} \propto \pi_1(X_1^{(i)})/\mu(X_1^{(i)})$ and thus obtain a weighted collection of particles which target the distribution π_1. In order to obtain a weighted collection of particles which target π_2, we simply extend each of these particles according to the conditional distribution implied by μ:

$$X_2^{(i)} \sim \mu(\cdot|X_1^{(i)}),$$

and then set the importance weights taking advantage of the above decomposition:

$$W_2^{(i)} \propto W_1^{(i)} \widetilde{W}_2^{(i)} \quad \widetilde{W}_2^{(i)} = \frac{\pi_2((X_1^{(i)}, X_2^{(i)}))}{\pi_1(X_1^{(i)})\mu(X_2^{(i)}|X_1^{(i)})}.$$

It is usual to refer to the $\widetilde{W}_2^{(i)}$ as *incremental weights* in this context. Thus we obtain a weighted sample which targets π_1, $\{X_1^{(i)}, W_1\}_{i=1}^N\}$ and subsequently one which targets π_2, with the minimum amount of additional sampling, $\{(X_1^{(i)}, X_2^{(i)}), W_2^{(i)}\}_{i=1}^N$ in which $\pi_1(X_1) \times \mu(X_2|X_1)$ is essentially used as an importance distribution for approximating π_2.

This procedure can be applied iteratively, leading to a sequence of weighted random samples which can be used to approximate a sequence of distributions on increasing state spaces. This scenario is precisely that faced in the smoothing scenario and is closely related to the filtering problem. In what follows, techniques for sequentially approximating the filtering and smoothing distributions will be presented.

10.2.1 The Natural SIS Filter

The first approach which one might consider is to use only samples from the prior distribution and the state transition densities together with importance sampling techniques in order to approximate the filtering, smoothing and one-step-ahead prediction distributions of interest. This approach is particularly natural and easy to understand, although we shall subsequently see that it has a number of shortcomings and more sophisticated techniques are generally required in practice.

As with all sequential importance sampling algorithms, the principle underlying this filter is that the empirical measure associated with a collection of weights and samples which approximates the filtering distribution at time t may be propagated forward in a particularly simple manner in order to obtain approximations to the predictive and filtering distributions at time $t+1$.

Given a collection of samples and associated weights, which we shall term *particles* at time t, $\{X_{1:t}^{(i)}, \hat{W}_t^{(i)}\}$ the associated empirical measure,

$$\hat{\pi}_{NSIS,t}^N(x_{1:t}) = \sum_{i=1}^N \hat{W}_i \delta_{X_{1:t}^{(i)}}(x_{1:t}),$$

provides an approximation to $\hat{\pi}_t$, an approximation to π_{t+1} can be obtained. We know that,

$$\pi_{t+1}(x_{1:t+1}) = \hat{\pi}_t(x_{1:t}) f_{t+1}(x_{t+1}|x_t),$$

and so one would like to use the approximation

$$\hat{\pi}_{NSIS,t}^N(x_{1:t}) f_{t+1}(x_{t+1}|x_t) = \sum_{i=1}^N \delta_{X_{1:t}^{(i)}}(x_{1:t}) f_{t+1}\left(x_{t+1}|x_t\right).$$

However, this would present us with a distribution which it is difficult to propagate forward further, and with respect to which it is difficult to calculate integrals. Instead, one might consider sampling from this distribution. As we will see later, this is the approach which is employed in some algorithms. In the present algorithm, a little more subtlety is employed: we have a mixture representation and so could use stratified sampling techniques to reduce the variance. Taking this to its logical conclusion, the simplest approach is to retain the mixture weights and sample a single value from each mixture component:

$$X_{t+1}^{(i)} \sim f_{t+1}\left(\cdot|X_t^{(i)}\right),$$

leading to the empirical measure,

$$\pi^N_{NSIS,t+1}(x_{1:t+1}) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X^{(i)}_{1:t+1}}(x_{1:t+1}).$$

We can obtain an approximation to $\hat{\pi}_{t+1}$ by substituting $\pi^N_{NSIS,t+1}$ directly into the recursion:

$$\begin{aligned}
\hat{\pi}_{t+1}(x_{1:t+1}) &= \frac{\pi_{t+1}(x_{1:t+1}) g_{t+1}(y_{t+1}|x_{t+1})}{\int \pi_{t+1}(x'_{1:t+1}) g_{t+1}(y_{t+1}|x'_{t+1}) dx'_{1:t+1}} \\
\Rightarrow \hat{\pi}^N_{NSIS,t+1}(x_{1:t+1}) &= \frac{\pi^N_{NSIS,t+1}(x_{t+1}) g_{t+1}(y_{t+1}|x_{t+1})}{\int \pi^N_{NSIS,t+1}(x'_{t+1}) g_{t+1}(y_{t+1}|x'_{t+1}) dx'_{t+1}} \\
&= \frac{\sum_{i=1}^{N} W_t^{(i)} g_{t+1}(y_{t+1}|X^{(i)}_{t+1}) \delta_{X^{(i)}_{t+1}}(x)}{\sum_{i=1}^{N} W_t^{(i)} g_{t+1}(y_{t+1}|X^{(i)}_{t+1})} \\
&= \sum_{i=1}^{N} W^{(i)}_{t+1} \delta^{(i)}_{X_{t+1}}(x),
\end{aligned}$$

where the updated importance weights are:

$$W^{(i)}_{t+1} = \frac{W_t^{(i)} g_{t+1}(y_{t+1}|X^{(i)}_{t+1})}{\sum_{j=1}^{N} W_t^{(j)} g_{t+1}(y_{t+1}|X^{(j)}_{t+1})}.$$

Note, that in practice these weights can be updated efficiently by setting $\hat{W}^{(i)}_{t+1} = W_t^{(i)} g_{t+1}(y_{t+1}|X^{(i)}_{t+1})$ and then setting $W^{(i)}_{t+1} = \hat{W}^{(i)}_{t+1} / \sum_{j=1}^{N} \hat{W}^{(j)}_{t+1}$.

If one can obtain a collection of weighted particles which target π_1 – which can typically be achieved by sampling from π_1 and setting all of the weights equal to $1/N$ – then this filter provides, recursively, weighted collections of particles from the smoothing distribution. As a byproduct, the filtering distribution, $\hat{\pi}_t(X_t)$, and the one-step-ahead predictive distribution, $\pi_t(X_t)$ can also be approximated using the same particle set:

$$\begin{aligned}
\pi^N_{NSIS,t}(x_t) &= \sum_{i=1}^{N} W_{t-1} \delta_{X_t^{(i)}}(x_t) \\
\hat{\pi}^N_{NSIS,t}(x_t) &= \sum_{i=1}^{N} W_t \delta_{X_t^{(i)}}(x_t).
\end{aligned}$$

Algorithm 1 provides a formal algorithm specification.

Algorithm 1 A Natural SIS Filter

1: Set $t = 1$.
2: For $i = 1 : N$, sample $X_1^{(i)} \sim \pi_1(\cdot)$.
3: For $i = 1 : N$, set $W_1^{(i)} \propto g_1(y_1|X_1^{(i)})$. Normalise such that $\sum_{i=1}^{N} W_1^{(i)} = 1$.
4: $t \leftarrow t + 1$
5: For $i = 1 : N$, sample $X_t^{(i)} \sim f_t(\cdot|X^{(i)}_{t=1})$.
6: For $i = 1 : N$, set $W_t^{(i)} \propto W^{(i)}_{t-1} g_t(y_t|X_t^{(i)})$. Normalise such that $\sum_{i=1}^{N} W_t^{(i)} = 1$.
7: The smoothing, filtering and predictive distributions at time t may be approximated with

$$\hat{\pi}^N_{NSIS,t} = \sum_{i=1}^{N} W_t \delta_{X^{(i)}_{1:t}}, \quad \hat{\pi}^N_{NSIS,t} = \sum_{i=1}^{N} W_t \delta_{X_t^{(i)}} \quad \text{and} \quad \pi^N_{NSIS,t} = \sum_{i=1}^{N} W_{t-1} \delta_{X_t^{(i)}}, \quad \text{respectively.}$$

8: Go to step 4.

By way of illustration, we consider applying the filter to a particularly simple HMM, in which $X_t \in \mathbb{R}$, and:

$$X_1 \sim \mathsf{N}(0, 1)$$
$$X_t | X_{t-1} = x_{t-1} \sim \mathsf{N}(0.9x_{t-1} + 0.7, 0.25)$$
$$Y_t | X_t = x_t \sim \mathsf{N}(x_t, 1).$$

This system is so simple that it can be solved analytically (in the sense that the distributions of interest are all Gaussian and a recursive expression for the parameters of those Gaussians may be derived straightforwardly), however, it provides a simple illustration of a number of important phenomena when the filter is applied to it.

24 states and their associated observations were generated by sampling directly from the model. Figure 10.1 shows the true states, the observations and the mean an 90% confidence interval for the state estimate obtained by using the filtering distribution obtained at each time in order to estimate the state at that time. While figure 10.2 shows the true states, the observations and the mean and 90% confidence interval associated with each state using the approximate smoothing distribution at time $t = 24$.

These two figures appear to illustrate the behaviour that would be expect, and to show good performance. In both instances, these graphs suggest that the empirical measures appear to provide a reasonable description of the distribution of the state sequence. It is also apparent that, by incorporating information from future observations as well as the past, the smoother is able to reduce the influence of occasional outlying observations.

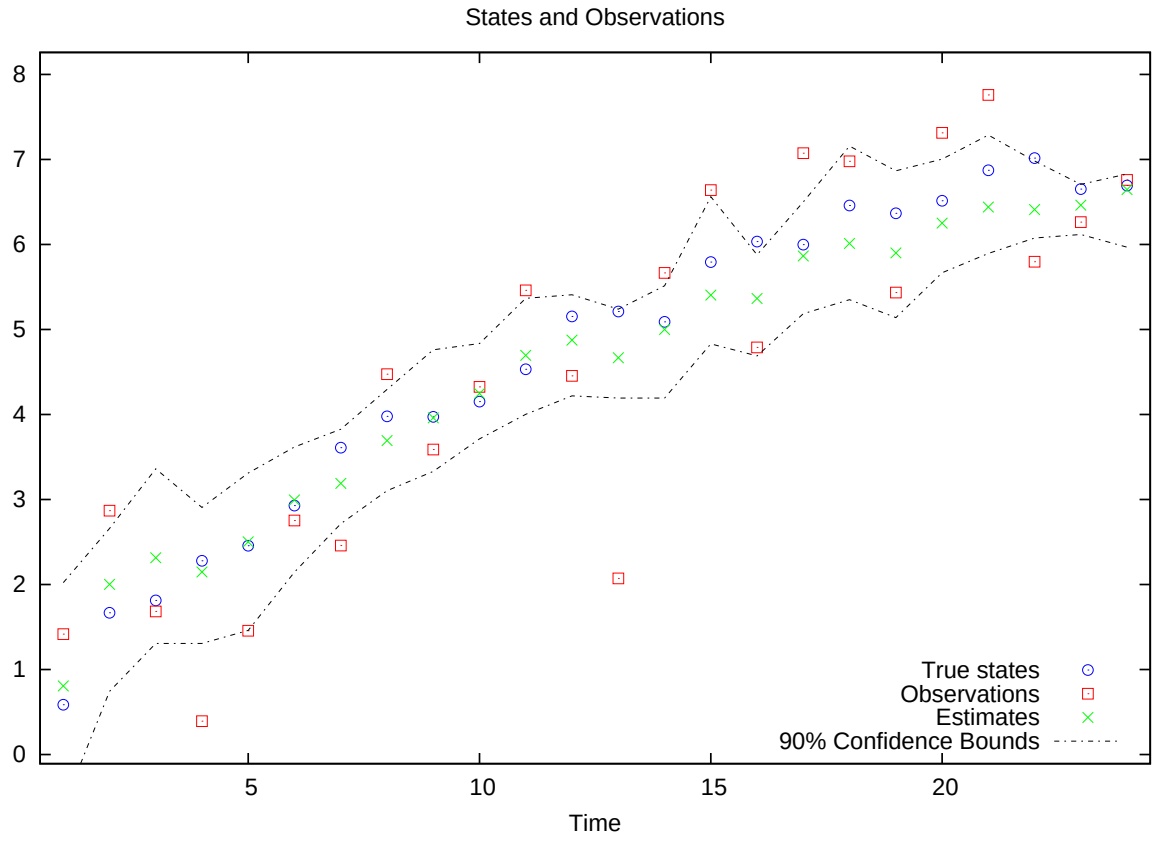


Fig. 10.1. Toy example: the true state and observation sequence, together with natural SIS filtering estimates.

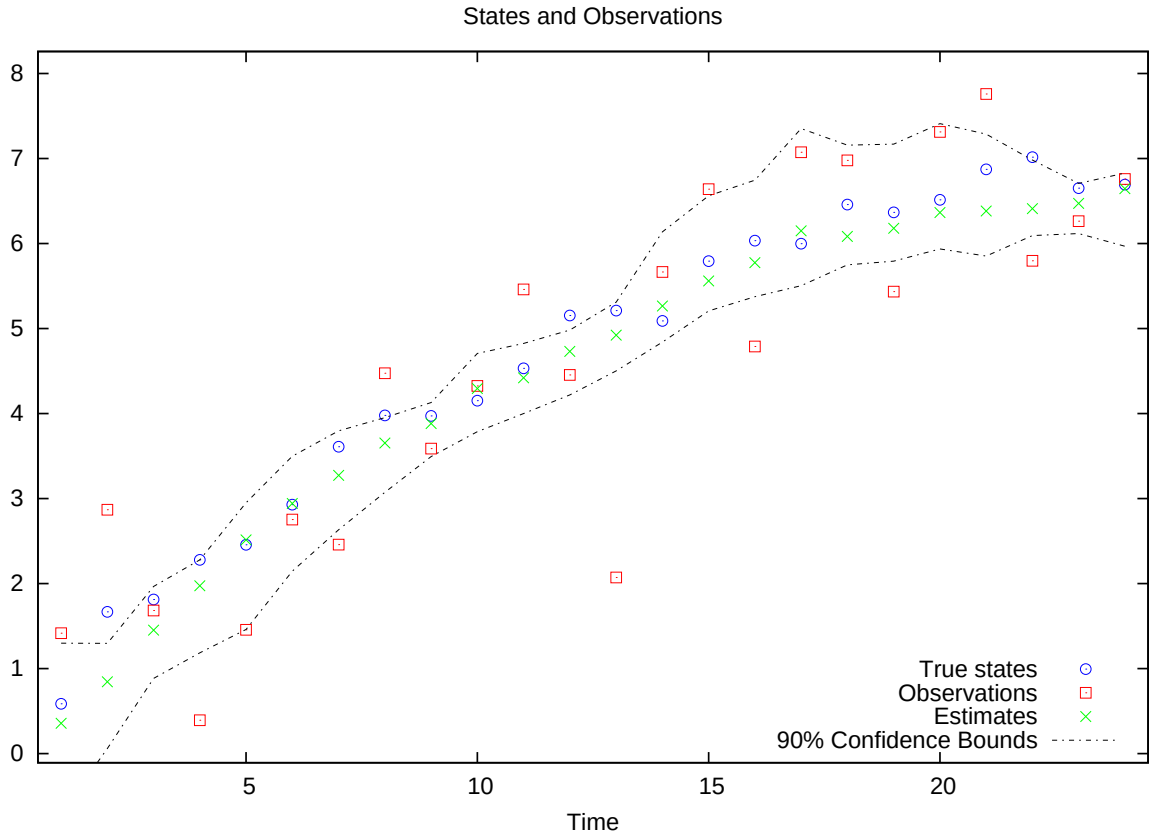


Fig. 10.2. Toy example: the true state and observation sequence, together with natural SIS smoothing estimates.

However, it is always important to exercise caution when considering simulation output. In order to illustrate how the filter is actually behaving, it is useful to look at a graphical representation of the approximate filtering and prediction distributions produced by the algorithm. Figure 10.3 shows the location and weights of the particles at for values of t. Note that this type of plot, which is commonly used to illustrate the empirical measure associated with a weighted collection of samples, does not have quite the same interpretation as a standard density plot. One must consider the number of particles in a region as well as their weights in order to associate a probability with that region. The first few plots show the behaviour that would be expected: the system is initialised with a collection of unweighted samples which are then weighted to produce an updated, weighted collection. For the first few time-steps everything continues as it should. The bottom half of the figure illustrates the empirical measures obtained during the twelfth and twenty-fourth iterations of the algorithm, and here there is some cause for concern. It appears that, as time passes, the number of particles whose weight is significantly above zero is falling with each iteration, by the twenty-fourth iteration there are relatively few samples contributing any significant mass to the empirical measure.

In order to investigate these phenomenon, it is useful to consider a very similar model in which the observations are more informative, and there is a greater degree of variability in the system trajectory:

$$X_1 \sim \mathsf{N}(0, 1)$$
$$X_t | X_{t-1} = x_{t-1} \sim \mathsf{N}(0.9x_{t-1} + 0.7, 1)$$
$$Y_t | X_t = x_t \sim \mathsf{N}(x_t, 0.1).$$

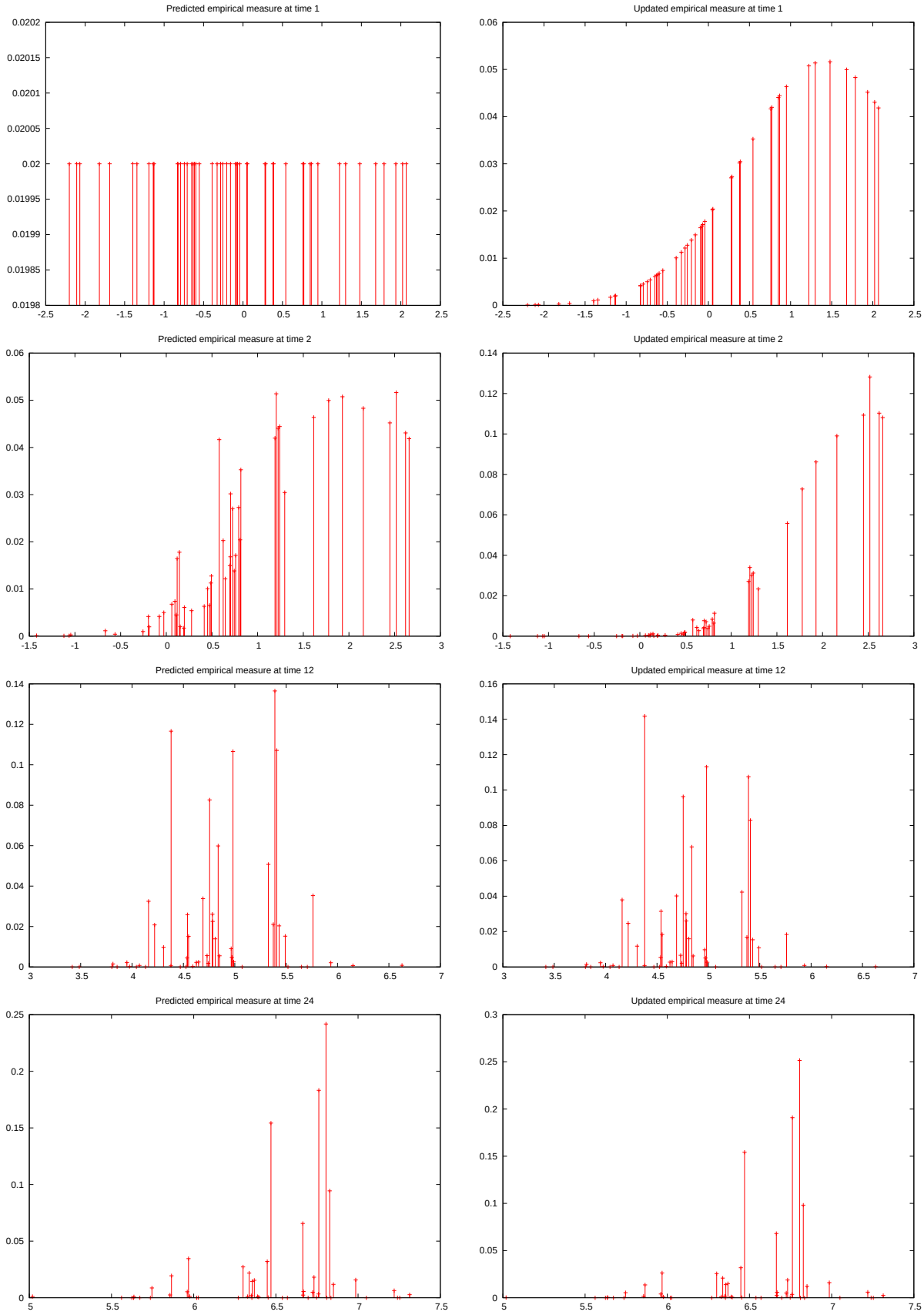


Fig. 10.3. Some predictive and filtering measures obtained via the natural SIS filter.

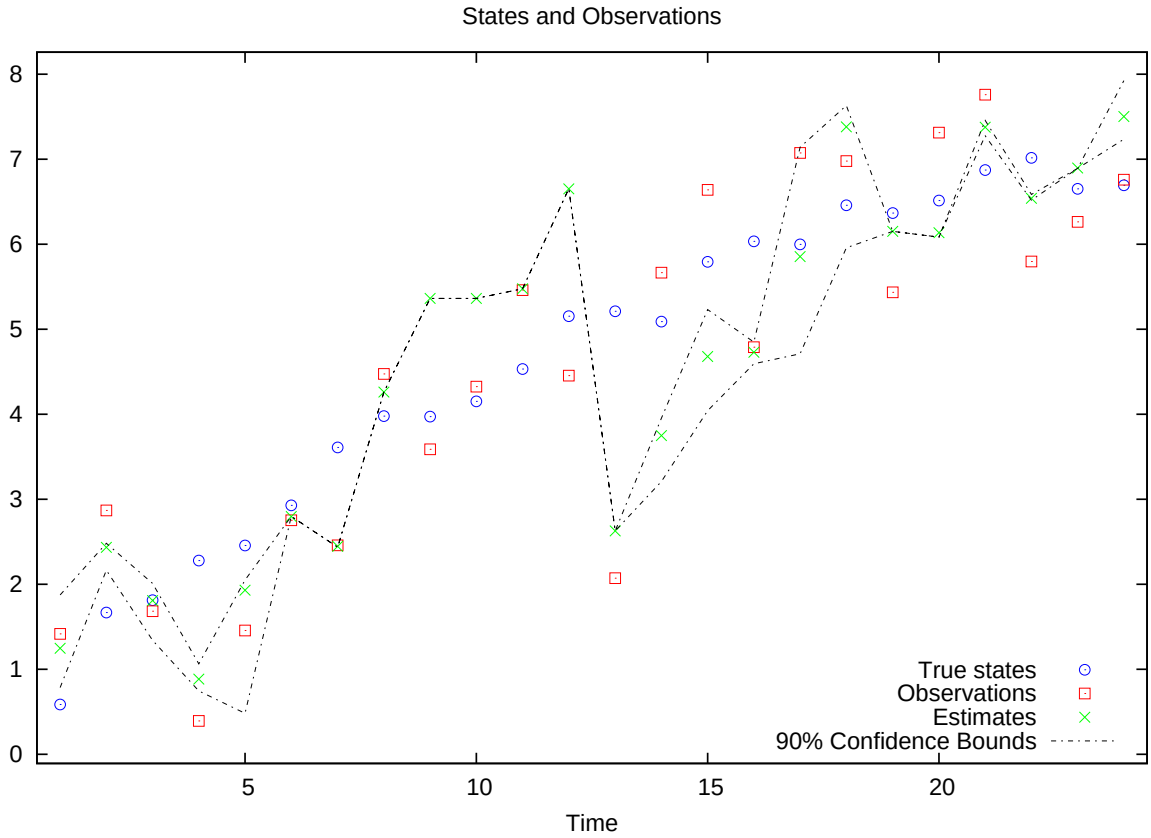


Fig. 10.4. Toy example: the true state and observation sequence, together with natural SIS filtering estimates.

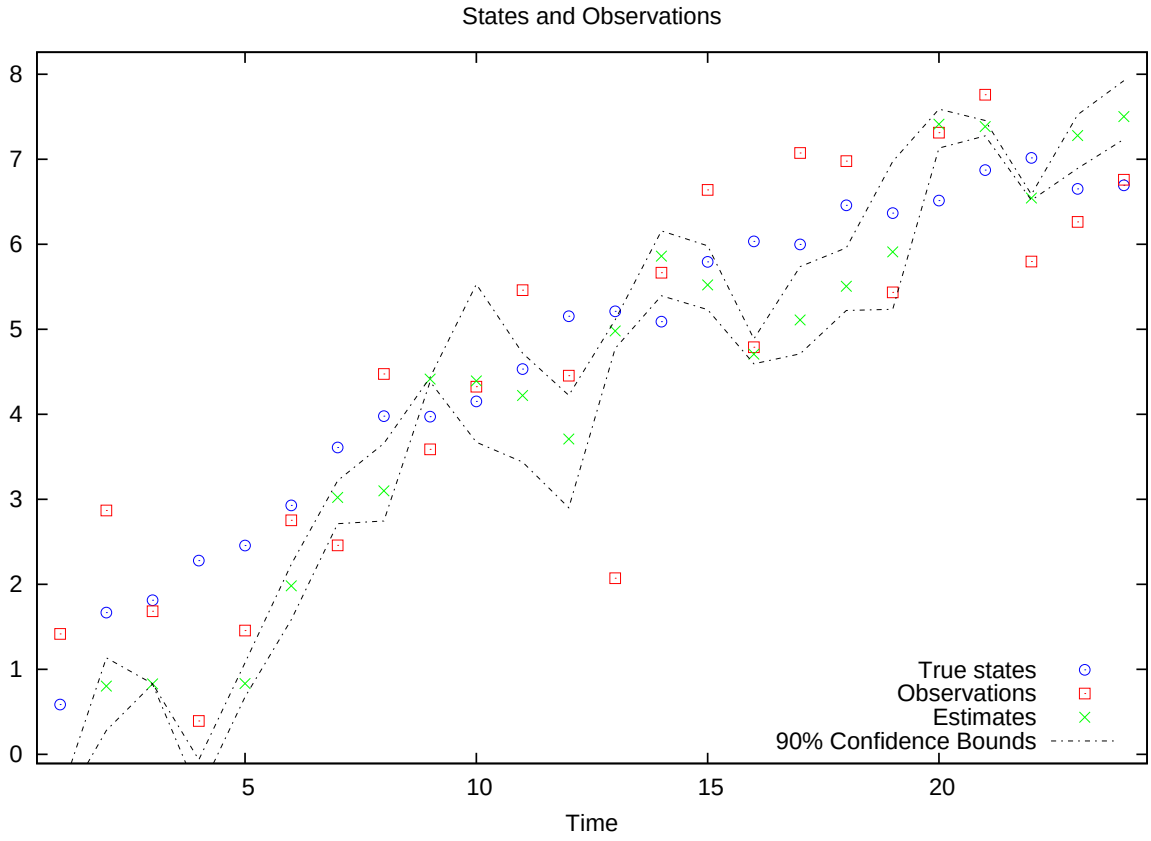


Fig. 10.5. Toy example: the true state and observation sequence, together with natural SIS smoothing estimates.

Figure 10.4 shows the filtering estimate of the trajectory of this system, while figure 10.5 illustrates the smoothed estimate. A selection of the associated empirical distributions are shown in figure 10.6. In the case of this example, it is clear that there is a problem.

10.2.2 Weight Degeneracy

The problems which are observed with the natural SIS filter are an example of a very general phenomenon known as *weight degeneracy*. Direct importance sampling on a very large space is rarely efficient as the importance weights exhibit very high variance. This is precisely the phenomenon which was observed in the case of the SIS filter presented in the previous section.

In order to understand this phenomenon, it is useful to introduce the following technical lemma which relates the variance of a random variable to its conditional variance and expectation (for any conditioning variable). Throughout this section, any expectation or variance with a subscript corresponding to a random variable should be interpreted as the expectation or variance with respect to that random variable.

Lemma 10.1 (Law of Total Variance). *Given two random variables, A and B, on the same probability space, such that* $\mathrm{Var}(A) < \infty$*, then the following decomposition exists:*

$$\mathrm{Var}(A) = \mathbb{E}_B\left[\mathrm{Var}_A(A|B)\right] + \mathrm{Var}_B\left(\mathbb{E}_A[A|B]\right).$$

Proof. By definition, and the law of total probability, we have:

$$\begin{aligned}\mathrm{Var}(A) &= \mathbb{E}\left[A^2\right] - \mathbb{E}[A]^2 \\ &= \mathbb{E}_B\left[\mathbb{E}_A\left[A^2|B\right]\right] - \mathbb{E}_B\left[\mathbb{E}_A[A|B]\right]^2.\end{aligned}$$

Considering the definition of conditional variance, and then variance, it is clear that:

$$\begin{aligned}\mathrm{Var}(A) &= \mathbb{E}_B\left[\mathrm{Var}_A(A|B) + \mathbb{E}_A[A|B]^2\right] - \mathbb{E}_B\left[\mathbb{E}_A[A|B]\right]^2 \\ &= \mathbb{E}_B\left[\mathrm{Var}_A(A|B)\right] + \mathbb{E}_B\left[\mathbb{E}_A[A|B]^2\right] - \mathbb{E}_B\left[\mathbb{E}_A[A|B]\right]^2 \\ &= \mathbb{E}_B\left[\mathrm{Var}_A(A|B)\right] + \mathrm{Var}_B\left(\mathbb{E}_A[A|B]\right).\end{aligned}$$

□

For simplicity, consider only the properly normalised importance sampling approach in which the normalising constant is known and used, rather than using the sum of the weights to renormalise the empirical measure. Consider a target distribution on $E = E_1 \times E_2$, and some sampling distribution μ on the same space. We may decompose the importance weight function as:

$$\begin{aligned}W_2(x_1, x_2) &= \frac{\pi(x_1, x_2)}{\mu(x_1, x_2)} \\ &= \underbrace{\frac{\pi(x_1)}{\mu(x_1)}}_{=:W_1(x_1)} \underbrace{\frac{\pi(x_2|x_1)}{\mu(x_2|x_1)}}_{=:\widetilde{W}_2(x_2|x_1)}\end{aligned}$$

We are now in a position to show that the variance of the importance weights over E is greater than that which would be obtained by considering only the marginal distribution over E_1. The following variance decomposition, together with lemma 10.1 provide the following equality:

$$\begin{aligned}\mathrm{Var}(W_2) &= \mathrm{Var}\left(W_1(X_1)\widetilde{W}_2(X_2|X_1)\right) \\ &= \mathrm{Var}_{X_1}\left(\mathbb{E}_{X_2}\left[W_1(X_1)\widetilde{W}_2(X_2|X_1)|X_1\right]\right) + \mathbb{E}_{X_1}\left[\mathrm{Var}_{X_2}\left(W_1(X_1)\widetilde{W}_2(X_2)|X_1\right)\right].\end{aligned}$$

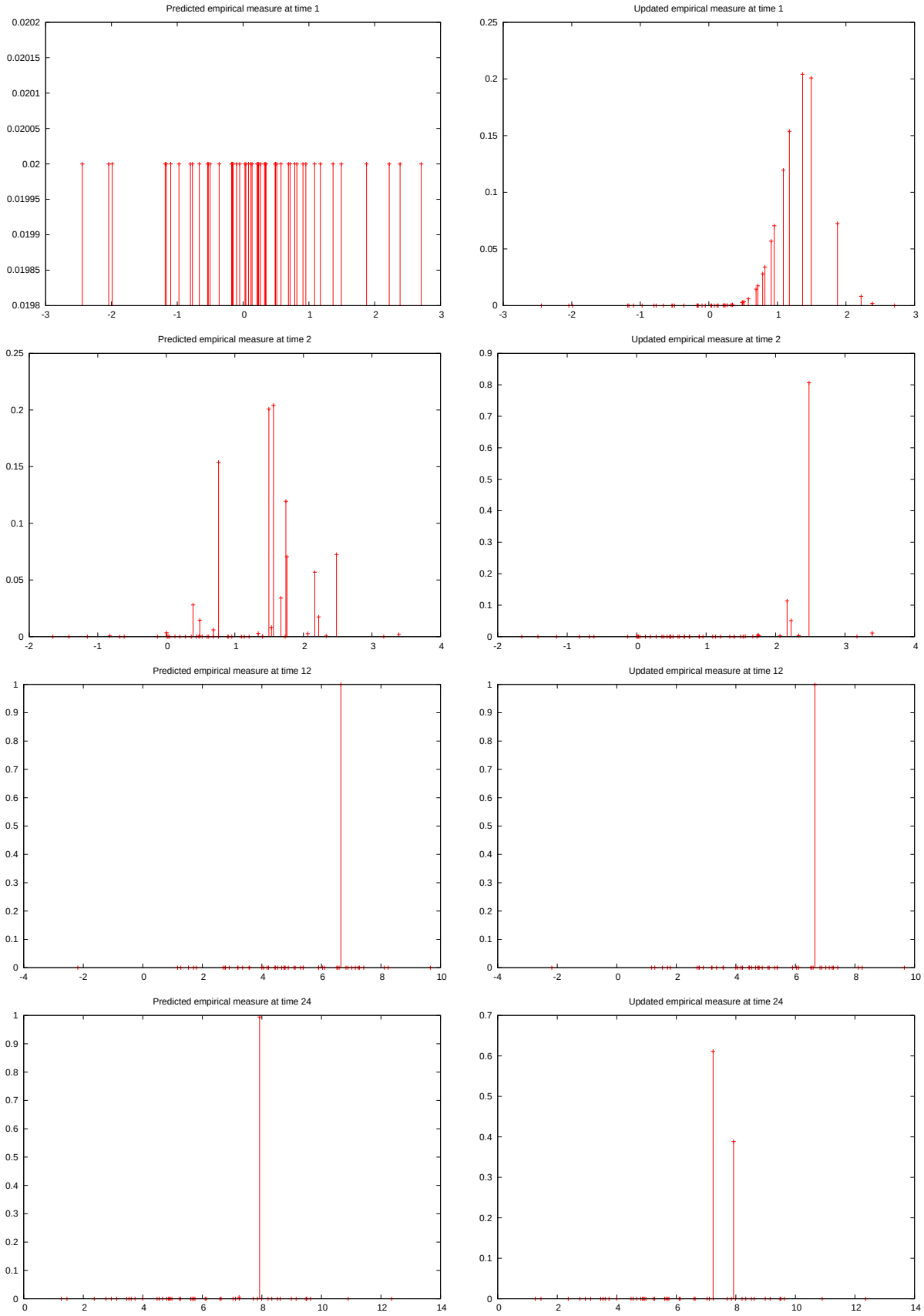


Fig. 10.6. Some predictive and filtering measures obtained via the natural SIS filter.

Noting that $W_1(X_1)$ is a deterministic function of X_1, we obtain:

$$\begin{aligned}\mathrm{Var}\left(W_2\right) &= \mathrm{Var}_{X_1}\left(W_1(X_1)\underbrace{\mathbb{E}_{X_2}\left[\widetilde{W}_2(X_2|X_1)|X_1\right]}_{1}\right) + \mathbb{E}_{X_1}\left[W_1(X_1)^2\mathrm{Var}_{X_2}\left(\widetilde{W}_2(X_2)|X_1\right)\right] \\ &= \mathrm{Var}_{X_1}\left(W_1(X_1)\right) + \mathbb{E}_{X_1}\left[W_1(X_1)^2\mathrm{Var}_{X_2}\left(\widetilde{W}_2(X_2)|X_1\right)\right] \geq \mathrm{Var}\left(W_1(X_1)\right).\end{aligned}$$

The final inequality holds as $W_1(X_1)^2$ and $\mathrm{Var}_{X_2}\left(\widetilde{W}_2(X_2)|X_1\right)$ non-negative functions. Indeed, equality is achieved only if the weight function is constant over the support of X_2, a situation which is somewhat uncommon.

This tells us, then, that there is a problem with the sequential importance sampling approach proposed above: the variance of the importance weights will increase with every iteration[1] and consequently, the quality of the estimators which the empirical measure provides will decrease as time progresses.

10.2.3 A More General Approach

The first approach to solving the problem of weight degeneracy or, at least, reducing its severity is to directly reduce the variance of the incremental importance weights. In order to attempt to do this, it is first necessary to consider a more general formulation of sequential importance sampling than the "natural" approach considered thus far.

Whilst there is a pleasing simplicity in iteratively approximating first the predictive and then the filtering/smoothing distributions, there is no need to do this in practice. It is often sufficient to obtain a sequence of measures which approximate the filtering/smoothing distributions – indeed, from these one can also form an estimate of the predictive distribution if one is required. With this perspective, the sequential importance sampling approach is this: given a weighted collection of samples $\{W_t^{(i)}, X_{1:t}^{(i)}\}_{i=1}^N$, which target the distribution $\hat{\pi}_t$, how can we obtain a collection of particles which target $\hat{\pi}_{t+1}$?

The answer to this question becomes apparent very quickly if the problem is looked at from the right angle. We have a collection of samples $\{X_{1:t}^{(i)}\}_{i=1}^N$ from some distribution $\mu_t^{(i)}$, and a collection of importance weights, $W_t^{(i)} = \hat{\pi}_t(X_{1:t})/\mu_t(X_{1:t})$. If we extend these samples, by sampling $X_{t+1}^{(i)} \sim q_{t+1}(\cdot|X_t)$, and wish to weight the collection $\{X_{1:t+1}^{(i)}\}_{i=1}^N$ such that it targets $\hat{\pi}_{t+1}$, we have:

$$W_{t+1}^{(i)} = \frac{\hat{\pi}_{t+1}(X_{1:t+1}^{(i)})}{\mu_t(X_{1:t}^{(i)})q_{t+1}(X_{t+1}^{(i)}|X_t^{(i)})},$$

and this leads immediately to the iterative decomposition

$$\begin{aligned}W_{t+1}^{(i)} =& \frac{\hat{\pi}_t(X_{1:t}^{(i)})}{\mu_t(X_{1:t}^{(i)})}\frac{\hat{\pi}_{t+1}(X_{1:t+1}^{(i)})}{\hat{\pi}_t(X_{1:t}^{(i)})q_{t+1}(X_{t+1}^{(i)}|X_t^{(i)})} \\ =& W_t^{(i)}\frac{\hat{\pi}_{t+1}(X_{1:t+1}^{(i)})}{\hat{\pi}_t(X_{1:t}^{(i)})q_{t+1}(X_{t+1}^{(i)}|X_t^{(i)})} \\ =& W_t^{(i)}\frac{\hat{\pi}_{t+1}(X_t^{(i)})}{\hat{\pi}_t(X_t^{(i)})}\frac{f_{t+1}(X_{t+1}^{(i)}|X_t^{(i)})g_{t+1}(y_{t+1}|X_{t+1}^{(i)})}{q_{t+1}(X_{t+1}^{(i)}|X_t^{(i)})}\end{aligned}$$

[1] Actually, this is not quite true. We have not considered the effect of renormalising importance weights here. It is evident that, at least asymptotically (in the number of particles) such normalisation has no effect on the outcome. A detailed analysis lends little additional intuition.

It is immediately clear that using a proposal distribution $q_{t+1}(x_{t+1}|x_{t-1}) \propto f_{t+1}(x_{t+1}|x_t)g_{t+1}(y_{t+1}|x_t)$ will minimise the variance of the importance weights. Thus, this is clearly the proposal distribution which minimises the variance of the incremental importance weights and hence, in that sense, is optimal.

All that this result tells us is that better performance will be expected if we take observations into account via the proposal distribution rather than solely via importance weighting. This is consistent with what we know about standard importance sampling: the best results are obtained when the proposal distribution matches the target distribution as closely as possible.

Initially, this result may seem rather formal: in general, it will not be possible to sample from this optimal proposal distribution. However, knowledge of its form may be used to guide the design of tractable distributions from which samples can easily be obtained and it is often possible to obtain good approximations of this distribution in quite complex realistic problems. Furthermore, this more general method has another substantial advantage over the natural approach considered above: it is only necessary to be able to evaluate the density of the transition probabilities pointwise up to a normalising constant whilst in the natural case it was necessary to be able to sample from that density. When using an approximation to the optimal importance variance, of course, the variance of the incremental importance weights is non-zero and the variance of the weights still increases over time. Unsurprisingly, using a better importance distribution reduces the variance but does not eliminate it. In the context of filtering, this means that using a close to optimal proposal distribution increases the time-scale over which the distributions of interest remain well characterised: eventually, degeneracy still occurs.

Algorithm 2 A General SIS Filter

1: Set $t = 1$.
2: For $i = 1 : N$, sample $X_1^{(i)} \sim q_1(\cdot)$.
3: For $i = 1 : N$, set $W_1^{(i)} \propto \pi_1(X_1^{(i)})g_1(y_1|X_1^{(i)})/q_1(X_1^{(i)})$. Normalise such that $\sum_{i=1}^{N} W_1^{(i)} = 1$.
4: $t \leftarrow t + 1$
5: For $i = 1 : N$, sample $X_t^{(i)} \sim q_t(\cdot|X_{t-1}^{(i)})$.
6: For $i = 1 : N$, set $W_t^{(i)} \propto W_{t-1}^{(i)} f_t(X_t^{(i)}|X_{t-1}^{(i)})g_t(y_t|X_t^{(i)})/q_t(X_t^{(i)}|X_{t-1}^{(i)})$. Normalise such that $\sum_{i=1}^{N} W_t^{(i)} = 1$.
7: The smoothing and filtering distributions at time t may be approximated with

$$\hat{\pi}_{SIS,t}^N = \sum_{i=1}^{N} W_t \delta_{X_{1:t}^{(i)}}, \qquad \text{and } \hat{\pi}_{SIS,t}^N = \sum_{i=1}^{N} W_t \delta_{X_t^{(i)}}, \qquad \text{respectively.}$$

8: Go to step 4.

10.3 Sequential Importance Resampling

Having established that there is a fundamental problem with SIS, the question of how to deal with that problem arises. The difficulty is that the variance of the importance weights increases over time; eventually, the associated particle system and empirical measure provide an inadequate description of the distributions of interest.

The variance in the importance weights is something which accumulates over a number of iterations: this suggests that that what is required is a mechanism for resetting the importance weights regularly to prevent this accumulation of variance. It is not immediately apparent that there is a mechanism by which this can be accomplished given that we are unable to sample directly from the distributions of interest

in any degree of generality. However, when concerned with approximating only the filtering distributions, (Gordon et al., 1993) developed the following approach. Under suitable regularity conditions, the law of large numbers tells us that for importance sampling:

$$\lim_{N\to\infty} \int \varphi(x_t)\hat{\pi}^N_{SIS,t}(x_t)dx_t \to \int \varphi(x_t)\hat{\pi}_t(x_t)dx_t,$$

for any suitably regular test function φ and so, any consistent estimator of the left hand side will provide a consistent estimate of the integral of interest.

We wish to obtain an unweighted collection of particles which approximate the distribution $\hat{\pi}^N_{SIS,t}(x_t)$ in some sense: this is exactly what we do in simple Monte Carlo. In order to approximate the integral on the left hand side, we can apply a crude Monte Carlo approximation, sampling, for $i = 1$ to N', $\tilde{X}^{(i)}_t \sim \hat{\pi}^{N'}_{SIS,t}$ and then using the simple Monte Carlo estimate of the integral. It is then straightforward to employ the law of large numbers to verify that:

$$\lim_{N'\to\infty} \frac{1}{N'} \sum_{i=1}^{N'} \varphi(\tilde{X}^{(i)}_t) \to \int \varphi(x_t)\hat{\pi}^N_{SIS,t}(x_t)dx_t$$

For simplicity, we assume that we sample N times from the empirical distribution in order to obtain our new distribution: $N' = N$ – this simplifies the computational implementation as well as analysis. Note that it is *not* possible to simply increase the number of particle samples at every iteration, although it initially seems advisable to do so to compensate for the additional Monte Carlo error introduced by the additional sampling step. Whilst it would, indeed, reduce the variance it would also lead to an exponentially increasing computational cost with the number of particles used rapidly exceeding the number that available computing facilities are able to deal with. Instead, it is generally recommended that one should use as many particles as computational resources permit from the outset.

Algorithm 3 A General SIR Filter

1: Set $t = 1$.

2: For $i = 1 : N$, sample $\tilde{X}^{(i)}_{1,1} \sim q_1(\cdot)$.

3: For $i = 1 : N$, set $W^{(i)}_1 \propto \pi_1(X^{(i)}_{1,1})g_1(y_1|\tilde{X}^{(i)}_{1,1})/q_1(\tilde{X}^{(i)}_1)$. Normalise such that $\sum_{i=1}^{N} W^{(i)}_1 = 1$.

4: Resample: for $i = 1 : N$, sample

$$X^{(i)}_{1,1,} \sim \frac{\sum_{i=1}^{N} W^{(i)}_1 \delta_{\tilde{X}^{(i)}_{1,1}}}{\sum_{j=1}^{N} W^{(j)}_1}.$$

5: $t \leftarrow t + 1$

6: For $i = 1 : N$, set $\tilde{X}^{(i)}_{t,1:t-1} = X^{(i)}_{t-1,1:t-1}$ and sample $\tilde{X}^{(i)}_{t,t} \sim q_t(\cdot|\tilde{X}^{(i)}_{t,t-1})$.

7: For $i = 1 : N$, set $W^{(i)}_t \propto f_t(\tilde{X}^{(i)}_{t,t}|\tilde{X}^{(i)}_{t,t-1})g_t(y_t|\tilde{X}^{(i)}_{t,t})/q_t(\tilde{X}^{(i)}_{t,t}|\tilde{X}^{(i)}_{t,t-1})$. Normalise such that $\sum_{i=1}^{N} W^{(i)}_t = 1$.

8: Resample: for $i = 1 : N$, sample

$$X^{(i)}_{t,1:t,} \sim \frac{\sum_{i=1}^{N} W^{(i)}_1 \delta_{\tilde{X}^{(i)}_{t,1:t}}}{\sum_{j=1}^{N} W^{(j)}_1}.$$

9: The smoothing and filtering distributions at time t may be approximated with

$$\hat{\pi}^N_{SIR,t}(x_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X^{(i)}_{t,1:t}}(x_{1:t}), \qquad \text{and } \hat{\pi}^N_{SIR,t}(x_t) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X^{(i)}_{t,t}}(x_t), \qquad \text{respectively.}$$

10: Go to step 5.

As algorithm 3 demonstrates, this technique can formally be generalised (in the obvious way) to provide an approximation to the smoothing distributions; however, as will be made clear shortly, there are difficulties with this approach and there is a good reason for concentrating on the filtering distribution. The generalisation is simply to resample the entire path followed by each particle through time, rather than only its terminal state. Note that doing this makes it necessary to introduce an additional subscript to account for the fact that the location of the j^{th} coordinate of the i^{th} particle changes through time (as we resample the full trajectory associated with each particle). Hence $X_{t,j}^{(i)}$ is used to refer to this quantity at time t, and $X_{t,1:t}^{(i)}$ refers to the full trajectory of the i^{th} particle at time t.

10.3.1 Sample Impoverishment

The motivation behind the resampling step introduced in the bootstrap filter and its generalisation was that it provides a method for eliminating particles with very low weights whilst replicating those with large weights. Doing this allows all of the particles to contribute significantly to the approximation of the distributions of interest. When considering the filtering distributions it is clear, loosely speaking, that resampling at the previous time step does lead to a better level of sample diversity as those particles which it was already extremely improbable at time $t-1$ are likely to have been eliminated and those which remain have a better chance of representing the situation at time t accurately.

There is, however, a difficulty. The resampling step does homogenize the particle weights but it can only do this by replicating particle values: we typically obtain several particles with particular values after each resampling step. This is not of itself a problem, providing that we retain a reasonable number of distinct particles and, in a reasonably well designed algorithm, this can be true for the filtering distribution. Considering the smoothing distribution makes it clear that there is a problem: the algorithm only ever reduces the number of distinct values taken at any time in the past. As the sampling mechanism only appends new values to the particle trajectories and the resampling mechanism eliminates some trajectories with every iteration, we ultimately end up with a degeneracy problem. The beginning of every particle trajectory will ultimately become the same. The reduction of the number of distinct samples within a collection by this mechanism, particularly when the reduction is pronounced is termed *sample impoverishment*.

Resampling is a more than cosmetic activity and it does more than eliminating direct evidence of sample degeneracy. However, it works by maintaining a good degree of sample diversity at the end of the particle trajectories at any given time: it does nothing to improve diversity in the past and the accumulation of degeneracy consequently remains a problem if one is interested in the full history.

Resampling at any given time simply increases the Monte Carlo variance of the estimator *at that time*. However, it can reduce the variance of estimators at later times and it is for that reason that it is widely used. For simplicity, in the algorithms presented in this course the estimator at time t is often given immediately after the resampling step. However, lower variance estimates would be obtained by employing the weighted sample available immediately before resampling and this is what should be done in practice except in situations in which it is essential to have an unweighted sample.

10.3.2 Effective Sample Sizes

It is clear that resampling must increase the immediate Monte Carlo variance (it is used because it will hopefully reduce the variance in the future and hence the variance cost of resampling may be offset by the reduction in variance obtained by having a more diverse collection of samples available). Consequently, it would be preferable not to resample during every iteration of the algorithm.

There competing requirements which must somehow be balanced: we must resample often enough to prevent sample degeneracy but we wish to do so sufficiently infrequently that sample impoverishment does not become a serious problem, at least for the filtering estimator. It would be useful to have a method for quantifying the quality of the current sample or, perhaps, to determine whether the current collection of weights is acceptable.

It is, of course, difficult to quantify the quality of a weighted sample which targets a distribution which is not known analytically. This is the problem which must be addressed when dealing with iterative algorithms of the sort described within this chapter. One figure of merit which could be used if it were available would be the ratio of the variance of an estimator obtained from this sample to that of an estimator obtained using crude Monte Carlo to sample from the target distribution itself (were that possible). Noting that the variance of the crude Monte Carlo estimator is proportional to the inverse of the sample size, N times this quantity corresponds to the *effective sample size* in the sense that this is the number of iid samples from the target distribution which would be required to obtain an estimator with the same variance.

It is useful to consider an abstract scenario. Let π denote a target distribution and μ an instrumental distribution. Assume that N samples from each are available, $\{X_i\}_{i=1}^N$ are iid samples from μ, whilst $\{Y_i\}_{i=1}^N$ are distributed according to π. Letting $W(x) \propto \pi(x)/\mu(x)$, and allowing φ to be a real-valued function whose integral we wish to estimate, we have two natural estimators:

$$\hat{h}_{CMC} = \frac{1}{N}\sum_{i=1}^{N}\varphi(Y_i)$$

$$\hat{h}_{IS} = \frac{\sum_{i=1}^{N} W(X_i)\varphi(X_i)}{\sum_{j=1}^{N} W(X_i)}.$$

The effective sample size mentioned above may be written as:

$$N_{ESS} = N\frac{\mathrm{Var}\left(\hat{h}_{CMC}\right)}{\mathrm{Var}\left(\hat{h}_{IS}\right)}.$$

Whilst this quantity might provide a reasonable measure of the quality of the weighted sample, it remains impossible to calculate it in any degree of generality. In order to make further progress it is necessary to make use of some approximations. It is shown in (Kong et al., 1994) that the quantity of interest may be approximated by $\mathrm{Var}\left(\hat{h}_{IS}\right)/\mathrm{Var}\left(\hat{h}_{CMC}\right) \approx 1 + \mathrm{Var}_\mu\left(\bar{W}\right)$, where $\bar{W}$ is the normalised version of the importance weights, $\bar{W}(x) = \pi(x)/\mu(x) \equiv W(x)/\int W(y)\mu(y)dy$. This approximation may be obtained via the delta method (which may be viewed as a combination of a suitable suitable central limit theorem and a Taylor expansion).

Considering this approximation:

$$\begin{aligned}1 + \mathrm{Var}_\mu\left(\bar{W}\right) &= 1 + \mathbb{E}_\mu\left[\bar{W}^2\right] - \mathbb{E}_\mu\left[\bar{W}\right]^2 \\ &= 1 + \mathbb{E}_\mu\left[\bar{W}^2\right] - 1 = \mathbb{E}_\mu\left[\bar{W}^2\right],\end{aligned}$$

which allows us to write $N_{ESS} \approx 1/\mathbb{E}_\mu\left[\bar{W}^2\right]$. It isn't typically possible to evaluate even this approximation, but looking at the expression we see that we need only evaluate an integral under the sampling distribution. This is precisely what is normally done in Monte Carlo simulation, and the approach which is usually used is to make use of the sample approximation of the expectation in place of the truth, so, we set:

$$\hat{N}_{ESS} := N \left(\frac{1}{N} \sum_{i=1}^{N} W(X_i)^2 \Big/ \left[\frac{1}{N} \sum_{j=1}^{N} W(X_j) \right]^2 \right)^{-1}$$
$$= \left[\sum_{j=1}^{N} W(X_j) \right]^2 \Big/ \sum_{i=1}^{N} W(X_i)^2.$$

This quantity can readily be calculated to provide an estimate of the quality of the sample at any iteration in a particle filtering algorithm. It is common practice to choose a threshold value for the effective sample size and to resample after those iterations in which the effective sample size falls below that threshold. Doing this provides a heuristic method for minimising the frequency of resampling subject to the constraint that it is necessary to retain a sample in which weight degeneracy is not too severe.

Whilst good results are obtained by using this approach when the proposal distribution is adequate, caution must be exercised in interpreting this approximation as an effective sample size. The use of the collection of samples itself to evaluate the integral which is used to assess its quality poses an additional problem. If the sample if particularly poor then it might provide a poor estimate of that integral and consequently, suggest that the sample is rather better than it actually is. In order to understand the most common pathology it is useful to consider a simple importance sampling scenario which generalises directly to the sequential cases considered within this chapter.

Consider a situation in which the proposal distribution has little mass in the mode of the target distribution and in which both distributions are reasonably flat in the mode of the proposal. It is quite possible that all of the importance weights are similar and the estimated ESS is large but, in fact, it is an extremely poor representation of the distribution of interest (in this setting, such a pathology will be revealed by using a large enough sample size; unfortunately, one does not know *a priori* how large that sample must be, and it is typically impossible to employ such a strategy in a sequential setting). For definiteness, let $\pi = 0.9\mathsf{N}(0,1) + 0.1\mathsf{N}(4, 0.1^2)$ and let $\mu = \mathsf{N}(0,1)$.

The proposal and target densities are illustrated in figure 10.7. Whilst the supports of these two distributions are the same and the tails of the proposal are as heavy as those of the target, it is immediately apparent that the two distributions are poorly matched: there is small probability of proposing samples with values greater than 3 and yet more than 90% of the mass of the target lies in this region. Consequently, the variance is finite but potentially extremely large. However, drawing collections of samples of various sizes from μ and calculating their ESS once they have been weighted to target π reveals a problem. Table

N	100	500	1,000	10,000
$\mathbf{N_{ESS}}$	100	500	999.68	1.8969
$\mathbf{N_{ESS}/N}$	1.000	1.000	0.9997	0.0002

Table 10.1. Effective sample sizes for the ESS example for various sample sizes.

10.1 shows the effective sample sizes estimated from samples of various different sizes. It is clear that when the estimator is a good one, N_{ESS}/N should converge to a constant. What we observe is that this ratio is close to unity for the smaller values of N which are considered (and notice that for a one dimensional problem these numbers do not seem that small). However, when a larger sample is used the effective sample size drops suddenly to a very low value: one which clearly indicates that the sample is no more use than one or two iid samples from the target distribution (in the sense quantified by the ESS, of course). On consideration it is clear that the problem is that unless the sample is fairly large, no particles will be

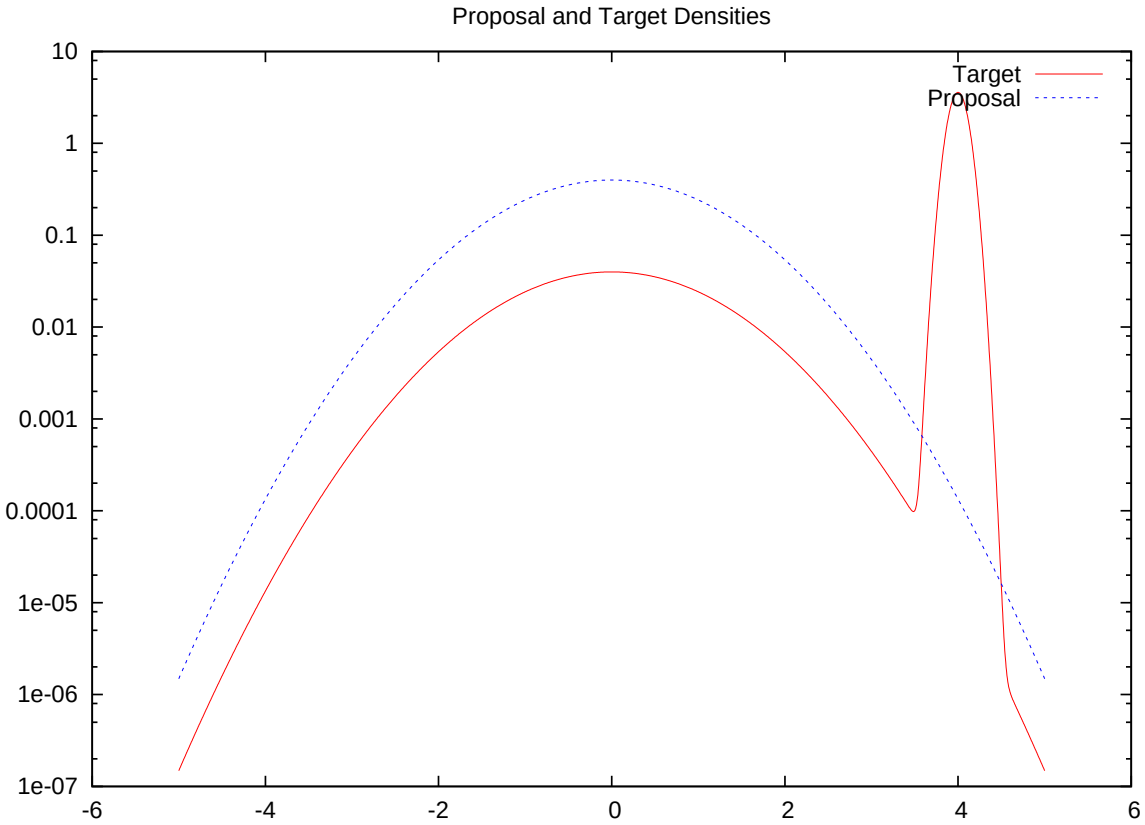


Fig. 10.7. Proposal and target densities for the ESS example scenario.

sampled in the region in which the importance weights are very large: outside this region those weights are approximately constant. In the example here it is clear that a different proposal distribution would be required, it would require an enormous number of particles to reduce the estimator variance to reasonable levels with the present choice.

Whilst this example is somewhat contrived, it demonstrates a phenomenon which is seen in practice in scenarios in which it is much more difficult to diagnose. In general, it is not possible to plot the target distribution at all and the true normalising constant associated with the importance weights is unknown.In summary, the standard approximation to the ESS used in importance sampling provides a useful descriptive statistic and one which can be used to guide the design of resampling strategies but it is not infallible and care should be taken before drawing any conclusions from its value. As always, there is no substitute for careful consideration of the problem at hand.

10.3.3 Approaches to Resampling

Whilst the presentation above suggests that only one approach to resampling exists, a more sophisticated analysis reveals that this isn't the case. Indeed, there are approaches which are clearly superior in terms of the Monte Carlo variance associated with them.

Resampling has been justified as another sampling step: approximating the weighted empirical distribution with an unweighted collection of samples from that distribution. Whilst this is intuitively appealing, an alternative view of resampling makes it clear that there are lower variance approaches which introduce no additional bias. We require that the expected value of the integral of a test function under the empirical distribution associated with the resampled particles matches that under the weighted empirical measure before resampling for all bounded integrable test functions. Assume that we have a collection of samples

$\{W_i, X_i\}_{i=1}^N$ and allow $\{\tilde{X}_i\}$ to denote the collection of particles after resampling. We want to ensure that:

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\varphi(\tilde{X}_i)\middle| X_{1:N}\right] = \sum_{i=1}^{N} W_i\varphi(X_i).$$

It is useful to consider that the $\tilde{X}_i$ must all be equal to one of the X_i as we a drawing from a discrete distribution with mass on only those values. Consequently, we could, alternatively consider the number of replicates of each of the original particles which are present in the resampled set. Let $M_i = |\{j : \tilde{X}_j = X_i\}|$ be the number of replicates of X_i which are present in the resampled set. Then, we have:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\varphi(\tilde{X}_i)\middle| X_{1:N}\right] &= \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} M_i\varphi(X_i)\middle| X_{1:N}\right] \\ &= \sum_{i=1}^{N}\mathbb{E}\left[\frac{M_i}{N}\varphi(X_i)\middle| X_{1:N}\right] \\ &= \sum_{i=1}^{N}\mathbb{E}\left[\frac{M_i}{N}\middle| X_{1:N}\right]\varphi(X_i).\end{aligned}$$

It is clear that any scheme in which $\mathbb{E}\left[\frac{M_i}{N}\middle| X_{1:N}\right] = W_i$ will introduce no additional bias. This is unsurprising: any scheme in which the expected number of replicates of a particle after resampling is precisely the weight associated with that sample will be unbiased. Clearly, many such schemes will have extremely large variance – for example, selecting a single index with probability proportional to W_i and setting all of the particles to X_i. However, there are a number of techniques by which such unbiased sampling can be accomplished with small variance.

A brief review of possible strategies is provided by (Robert and Casella, 2004, Section 14.3.5), but three of the most commonly used approaches are summarised here.

Multinomial Resampling. The approach which has been described previously, in which each $\tilde{X}_i$ is drawn independently from the empirical distribution associated with the collection $\{W_i, X_i\}_{i=1}^N$ is equivalent to drawing the vector of replicate counts M from a multinomial distribution with N trials and parameter vector $W_{1:N}$. That is: $M \sim \mathsf{M}(\cdot|N, W)$, where the multinomial distribution is a generalisation of the binomial distribution to trials which have more than two possible outcomes. It takes mass on the points $\{M \in \mathbb{R}^N : \sum_{i=1}^N M_i = N\}$ and has the probability mass function:

$$\mathsf{M}(M|N, W) = \begin{cases} \frac{N!}{\prod_{i=1}^N M_i!}\prod_{i=1}^N W_i^{M_i} & \text{if } \sum_{i=1}^N M_i = N, \\ 0 & \text{otherwise.}\end{cases}$$

For this reason the simple scheme described above is usually termed *multinomial resampling.*

Residual Resampling. The reason that resampling is not carried out deterministically is that in general NW_i is not an integer and so it is not possible to replicate particles deterministically in a manner which is unbiased. However, it is possible to remove the integer component of NW_i for each weight and then to assign the remainder of the mass by multinomial resampling. This is the basis of a scheme known as *residual resampling.* The approach is as follows:

1. Set $\tilde{M}_i = \lfloor NW_i \rfloor$ where $\lfloor NW_i \rfloor := \sup\{z \in \mathbb{Z} : z < NW_i\}$.
   Set $\tilde{W}_i = W_i - \lfloor NW_i \rfloor / N$.
2. Sample $M' \sim \mathsf{M}(\cdot|N - \sum_{i=1}^N \lfloor NW_i \rfloor, \tilde{W})$.
3. Set $M_i = \tilde{M}_i + M'_i$.

The intuition is that by deterministically replicating those particles that we expect at least one of in the replicated set and introducing randomness only to allow us to deal with the non-integer (residual) components of NW_i we retain the unbiased behaviour of multinomial resampling whilst substantially reducing the degree of randomness introduced by the resampling procedure. It is straightforward to verify that the variance of this approach is, indeed, less than that of multinomial resampling.

Stratified and Systematic Resampling. The motivation for residual resampling was that deterministically replicating particles and hence reducing the variability of the resampled particle set whilst retain the lack of bias must necessarily provide an improvement. Another approach is motivated by the stratified sampling technique. In order to sample from mixture distributions with known mixture weights, the variance is reduced if one draws a deterministic number of samples from each component, with the number proportional to the weight of that component[2]. Taking this to its logical conclusion, one notices that one can partition any single dimensional distribution in a suitable manner by taking it cumulative distribution function, dividing it into a number of segments, each with equal associated probability mass, and then drawing one sample from each segment (which may be done, at least in the case of discrete distributions by drawing a uniform random variable from the range of each of the CDF segments and applying inversion sampling).

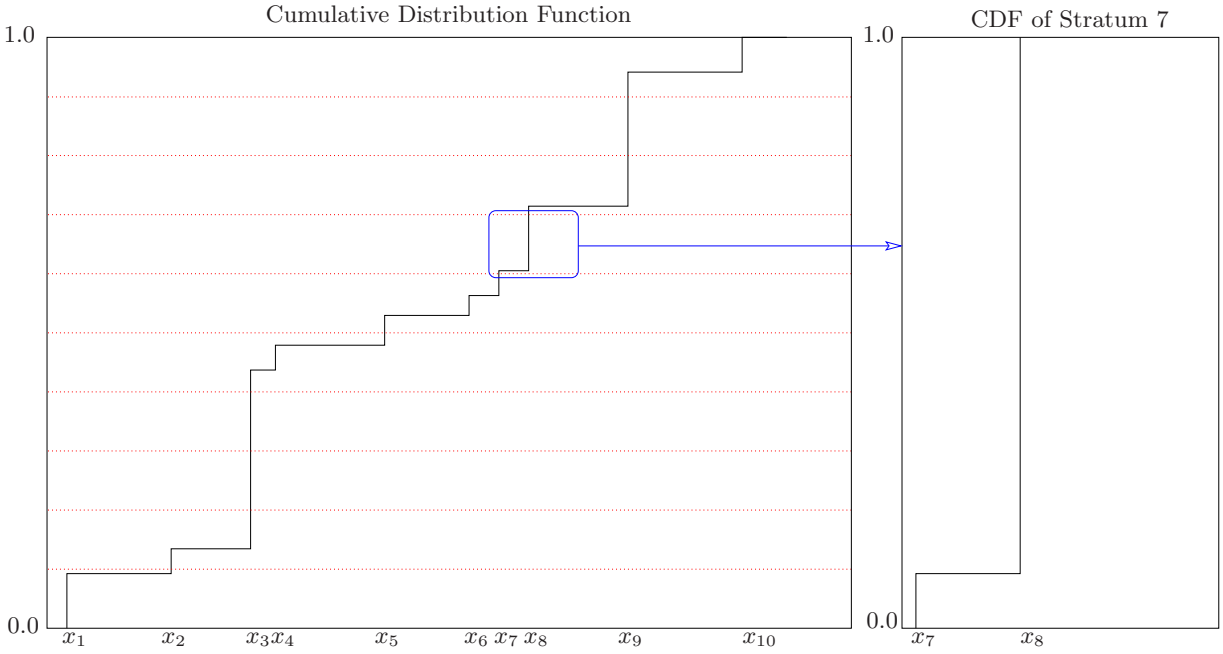


Fig. 10.8. The cumulative distribution associated with a collection of ten particles divided into ten strata, together with the distribution function associated with the seventh stratum.

Figure 10.8 illustrates this technique for a simple scenario in which ten particles are present. As usual, the cumulative distribution function associated with a discrete distribution (the empirical distribution associated with the weighted particles in this case) consists of a number of points of discontinuity connected by regions in which the CDF is constant. These discontinuities have magnitudes corresponding to the sample weights and locations corresponding to the values of those particles. If the sample values are not one dimensional it becomes easier to consider sampling the particle indices rather than the values themselves and then using those indices to deduce the values of the resampled particles; this adds little

[2] Again, a little additional randomness may be required to prevent bias if the product of the number of samples required and the various mixture weights is not an integer.

complexity. Having obtained the CDF, one divides it into N equally-spaced segments vertically (such that each encloses equal probability mass). Each segment of the CDF may then be rescaled to produce a new CDF describing the distribution conditional upon it being drawn from that stratum – the figure illustrates the case of the seventh stratum from the bottom in a particular case. Inversion sampling once from each stratum then provides a new collection of N samples.

It can be seen that this approach has some features in common with residual resampling. For example, a particle with a large weight will be replicated with probability one (in the case of stratified resampling, if a particle has weight W_i such that $NW_i \geq n+1$ then it is guaranteed to produce at least n replicates in the resampled population. For example, in figure 10.8, the particle with value x_3 has a sufficiently large weight that it will certainly be selected from stratum 3 and 4, and it has positive probability of being drawn in the two adjacent strata as well.

In fact, it is possible to further reduce the randomness, by drawing a single uniform random number and using that to select the value sample from *all* of the segments of the CDF. This technique is widely used in practice although it is relatively difficult to analyse. In order to implement this particular version of resampling, typically termed *systematic resampling*, one simply draws a random number, U, uniformly in the interval $[0, 1/N]$ and then, allowing F_N to denote the CDF of the empirical distribution of the particle set before resampling, set, for each i in $1, \ldots, N$:

$$\tilde{X}_i := F_N^{-1}(i/N - U).$$

In terms of figure 10.8, this can be understood as drawing a random variable, U, uniformly from an interval describing the height of a single stratum and then taking as the new sample those values obtained from the CDF at a point U below the top of any stratum. It is straightforward to establish that this technique is unbiased, but the correlated samples which it produces complicate more subtle analysis. The systematic sampling approach ensures that the expected number of replicates of a particle with weight W_i is NW_i and, furthermore, the actual number is within 1 of NW_i.

10.4 Resample-Move Algorithms

An attempt at reducing the sample impoverishment was proposed by (Gilks and Berzuini, 2001a). Their approach is based upon the following premise which (Robert and Casella, 2004) describe as *generalised importance sampling*. Given a target distribution π, an instrumental distribution μ and a π-invariant Markov kernel, K, the following generalisation of the importance sampling identity is trivially true:

$$\int \mu(x)K(x,y)\frac{\pi(x)}{\mu(x)}h(y)dxdy = \int \mu(x)K(x,y)\frac{\pi(x)}{\mu(x)}dxh(y)dy \quad = \int \pi(x)K(x,y)dxh(y)dy = \int \pi(y)h(y)dy.$$

In conjunction with the law of large numbers this tells us, essentially, that given a weighted collection of particles $\{W^{(i)}, X^{(i)}\}_{i=1}^N$ from μ weighted to target π, if we sample $Y^{(i)} \sim K(X^{(i)}, \cdot)$ then the weighted collection of particles $\{W^{(i)}, Y^{(i)}\}_{i=1}^N$ also targets π.

Perhaps the simplest way to verify that this is true is simply to interpret the approach as importance sampling on an enlarged space using $\mu(x)K(x,y)$ as the proposal distribution for a target $\pi(x)K(x,y)$ and then estimating a function $h'(x,y) = h(y)$. It is then clear that this is precisely standard importance sampling. As with many areas of mathematics the terminology can be a little misleading: correctly interpreted, "generalised importance sampling" is simply a particular case of importance sampling.

Whilst this result might seem rather uninteresting, it has a particularly useful property. Whenever we have a weighted sample which targets π, we are free to *move* the samples by applying any π-invariant

Markov kernel in the manner described above and we retain a weighted sample targeting the distribution of interest. (Gilks and Berzuini, 2001a) proposed to take advantage of this to help solve the sample impoverishment problem. They proposed an approach to filtering which they termed *resample-move*, in which the particle set is moved according to a suitable Markov kernel after the resampling step. In the original paper, and the associated tutorial (Gilks and Berzuini, 2001b), these moves are proposed in the context of a standard SIR algorithm but, in fact, is is straightforward to incorporate them into any iterative algorithm which employs a collection of weighted samples from a particular distribution at each time step.

Algorithm 4 A Resample-Move Filter

1: Set $t = 1$.
2: For $i = 1 : N$, sample $\tilde{X}_{1,1}^{(i)} \sim q_1(\cdot)$.
3: For $i = 1 : N$, set $W_1^{(i)} \propto \pi_1(X_{1,1}^{(i)}) g_1(y_1|\tilde{X}_{1,1}^{(i)})/q_1(\tilde{X}_1^{(i)})$. Normalise such that $\sum_{i=1}^{N} W_1^{(i)} = 1$.
4: Resample: for $i = 1 : N$, sample

$$\hat{X}_{1,1,}^{(i)} \sim \frac{\sum_{i=1}^{N} W_1^{(i)} \delta_{\tilde{X}_{1,1}^{(i)}}}{\sum_{j=1}^{N} W_1^{(j)}}.$$

5: Move: for $i = 1 : N$, sample $X_{1,1}^{(i)} \sim K_1(\hat{X}_{1,1}^{(i)}, \cdot)$, where K_1 is π_1-invariant.
6: $t \leftarrow t + 1$
7: For $i = 1 : N$, set $\tilde{X}_{t,1:t-1}^{(i)} = X_{t-1,1:t-1}^{(i)}$ and sample $\tilde{X}_{t,t}^{(i)} \sim q_t(\cdot|\tilde{X}_{t,t-1}^{(i)})$.
8: For $i = 1 : N$, set $W_t^{(i)} \propto W_{t-1}^{(i)} f_t(\tilde{X}_{t,t}^{(i)}|\tilde{X}_{t,t-1}^{(i)}) g_t(y_t|\tilde{X}_{t,t}^{(i)})/q_t(\tilde{X}_{t,t}^{(i)}|\tilde{X}_{t,t-1}^{(i)})$. Normalise such that $\sum_{i=1}^{N} W_t^{(i)} = 1$.
9: Resample: for $i = 1 : N$, sample

$$\hat{X}_{t,1:t,}^{(i)} \sim \frac{\sum_{i=1}^{N} W_1^{(i)} \delta_{\tilde{X}_{t,1:t}^{(i)}}}{\sum_{j=1}^{N} W_1^{(j)}}.$$

10: Move: for $i = 1 : N$, sample $X_{t,1:t}^{(i)} \sim K_t(\hat{X}_{t,1:t}, \cdot)$, where K_t is π_t-invariant.
11: The smoothing and filtering distributions at time t may be approximated with

$$\hat{\pi}_{SIS,t}^{N}(x_{1:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{t,1:t}^{(i)}}(x_{1:t}), \qquad \text{and } \hat{\pi}_{SIS,t}^{N}(x_t) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{t,t}^{(i)}}(x_t), \qquad \text{respectively.}$$

12: Go to step 6.

Thus some progress has been made: a mechanism for injecting additional diversity into the sample has been developed. However, there is *still* a problem in the smoothing scenario. If we were to apply Markov Kernels of invariant distribution corresponding to the full path-space smoothing distribution at each time, then the space on which those Markov kernels were defined must increase with every iteration. This would have two effects: the most direct is that it would take an increasing length of time to apply the move with every iteration; the second is that it is extremely difficult to design fast-mixing Markov kernels on spaces of high dimension. If an attempt is made to do so in general it will be found that the degree of movement provided by application of the Markov kernel decreases in some sense as the size of the space increases. For example, if a sequence of Gibbs sampler moves are used, then each coordinate will move only very slightly whilst a Metropolis Hastings approach will either proposal only very small perturbations or will have a very high rate of rejection.

In practice, one tends to employ Markov kernels which only alter the terminal value of the particles (although there is no difficulty with updating the last few values by this mechanism, this has an increased computational cost and does not *directly* alter the future trajectories). The insertion of additional MCMC

moves (as the application of a Markov kernel of the correct invariant distribution is often termed) in a filtering context, then, is a convenient way to increase the diversity of the terminal values of the sample paths.

It is not a coincidence that the various techniques to improve sample diversity all provide improved approximations of the filtering distribution but do little to improve the distribution over the path-space distributions involved in smoothing. In fact, theoretical analysis of the various particle filters reveal that their stability arises from that of the system which they are approximating. Under reasonable regularity conditions, the filtering equations can be shown to forget their past: a badly initialised filter will eventually converge to the correct distribution. As the dynamics of the particle system mirror those of the exact system which it is approximating, this property is also held by the particle system. It prevents the accumulation of errors and allows convergence results to be obtained which are *time uniform* in the sense that the same quality of convergence (in a variety of senses) can be obtained at all times with a constant number of particles if one is considering only the filtering distributions; similar results have not been obtained in the more general smoothing case and there are good technical reasons to suppose that they will not be forthcoming. Technical results are provided by (Del Moral, 2004, Chapter 7) but these lie far outside the scope of this course. It is, however, important to be aware of this phenomenon when implementing these algorithms in practice.

10.5 Auxiliary Particle Filters

The auxiliary particle filter (APF) is a technique which was originally proposed by (Pitt and Shephard, 1999, 2001), in the form presented in section 10.5.2, based upon auxiliary variables. It has recently been recognised that it can, in fact, be interpreted as no more than an SIR algorithm which targets a slightly different sequence of distributions, and then uses importance sampling to correct for the discrepancy; this intuitive explanation, due to (Johansen and Doucet, 2007) is presented in section 10.5.3.

10.5.1 Motivations

Before becoming involved in the details, it is necessary to look at one particular problem from which SMC methods can suffer: and what one would like to do in order to assuage this problem. That difficulty is that one resamples the particle set at the conclusion of one iteration of the algorithm before moving them and then weighting them, taking into account the most recent observation. In practice, this observation can provide significant information about the likely state at the previous time (whilst x_t is independent of y_{t+1} conditional upon the knowledge of x_{t+1} we do not, in fact, know x_{t+1}). Whilst deferring resampling isn't really an option as it would simply mean leaving resampling until the following iteration and leaving the same problem for that iteration, it would be nice if one could pre-weight the particles prior to resampling to reflect how their compatibility with the next observation. This is essentially the idea behind the APF.

The relationship, in the filtering and smoothing context, between x_n and y_{n+1}, assuming that x_{n+1} is unknown (we wish to establish how well x_n is able to account for the next observation *before* sampling the next state) is:

$$p(y_{n+1}|x_{1:n}, y_{1:n}) = p(x_n|y_{1:n}) \int f_{n+1}(x_{n+1}|x_n) g_{n+1}(y_{n+1}|x_{n+1}) dx_{n+1}$$

which is simply the integral of the joint distribution of $x_{n:n+1}$ and y_{n+1} given $y_{1:n}$. Defining $g_{n+1}(y_{n+1}|x_n) = \int f_{n+1}(x_{n+1}|x_n) g_{n+1}(y_{n+1}|x_{n+1}) dx_{n+1}$, it would be desirable to use a term of this sort to determine how

well a particle matches the next observation in advance of resampling (before correcting with the discrepancy that this introduces into the distribution after resampling by importance weighting).

10.5.2 Standard Formulation

The usual view of the APF is that it is a technique which employs some approximation $\hat{g}_{n+1}(y_{n+1}|x_n)$ to the *predictive likelihood*, $g_n(y_{n+1}|x_n)$ in an additional pre-weighting step and employs an auxiliary variable technique to make use of these weights. The original proposal for the APF had a number of steps for each iteration:

1. Apply an auxiliary weighting proportional to $\hat{g}_{n+1}(y_{n+1}|x_n)$.
2. Using the auxiliary weights propose moves from a mixture distribution.
3. Weight moves to account for the auxiliary weighting introduced previously, the observation and the proposal distribution.
4. Resample according to the standard weights.

Although the first three of these steps differ from a standard SIR algorithm the third differs only in the nature of the importance weighting and the first two are essentially the key innovation associated with the auxiliary particle filter.

The idea is that, given a sample of particles which target $\hat{\pi}_t$, and knowledge of the next observation, y_{t+1}, we can make use of an approximation of the conditional likelihood to attach a weight, $\lambda^{(i)}$ each particle. This weight has the interpretation that it describes how consistent each element of the sample is with the next observation. Having determined these weightings, one proposes the values of each particle at time $t+1$ (in the filtering case which was considered originally) independently from the mixture distribution

$$\sum_{j=1}^{N} \frac{\lambda^{(j)}}{\sum_{k=1}^{N} \lambda^{(k)}} q_{n+1}(\cdot|X_t^j).$$

The samples are then weighted such that they target $\hat{\pi}_{t+1}$ and then resampling is carried out as usual. This procedure is carried out iteratively as in any other particle filter, with the detailed algorithmic description being provided by algorithm 5.

10.5.3 Interpretation as SIR

However, over time it has become apparent that there is actually little point in resampling immediately prior to the auxiliary weighting step. Indeed, the use of an auxiliary variable in the manner described in the previous section is exactly equivalent to first resampling the particles (via a multinomial scheme) according to the auxiliary weights and then making standard proposals with that for each particle being made from the previous value of that particle. Thus resampling, weighting and resampling again will introduce additional Monte Carlo variance for which there is no benefit. In fact, it makes more sense to resample after the auxiliary weighting step and not at the end of the algorithm iteration; the propagation of weights from the previous iterations means that resampling is then carried out on the basis of both the importance weights and the auxiliary weights.

Having noticed this, it should be clear that there is actually little more to the APF than there is to SIR algorithms and, indeed, it is possible to interpret the auxiliary particle filter as no more than an SIR algorithm which targets a slightly different sequence of distributions to those of interest and then to use

Algorithm 5 Traditional View of the APF

1: Set $t = 1$.
2: For $i = 1 : N$, sample $\tilde{X}_{1,1}^{(i)} \sim q_1(\cdot)$.
3: For $i = 1 : N$, set $W_1^{(i)} \propto \pi_1(X_{1,1}^{(i)}) g_1(y_1|\tilde{X}_{1,1}^{(i)})/q_1(\tilde{X}_1^{(i)})$. Normalise such that $\sum_{i=1}^{N} W_1^{(i)} = 1$.
4: Resample: for $i = 1 : N$, sample

$$X_{1,1,}^{(i)} \sim \frac{\sum_{i=1}^{N} W_1^{(i)} \delta_{\tilde{X}_{1,1}^{(i)}}}{\sum_{j=1}^{N} W_1^{(j)}}.$$

5: $t \leftarrow t + 1$
6: Calculate auxiliary weights: set $\lambda_t^{(i)} \propto \hat{g}_t(y_t|X_{t-1,t-1}^{(i)})$ and normalise such that $\sum_{i=1}^{N} \lambda_t^{(i)} = 1$.
7: Sample $\alpha_t^{(i)}$ such that $\mathbb{P}\left(\alpha_t^{(i)} = j\right) = \lambda_t^{(j)}$ (i.e. sample from the discrete distribution with parameter λ_t).
8: For $i = 1 : N$, set $\tilde{X}_{t,1:t-1}^{(i)} = X_{t-1,1:t-1}^{(\alpha_t^{(i)})}$ and sample $\tilde{X}_{t,t}^{(i)} \sim q_t(\cdot|\tilde{X}_{t,t-1}^{(i)})$.
9: For $i = 1 : N$, set

$$W_t^{(i)} \propto \frac{f_t(\tilde{X}_{t,t}^{(i)}|\tilde{X}_{t,t-1}^{(i)}) g_t(y_t|\tilde{X}_{t,t}^{(i)})}{\lambda_t^{(\alpha_t^{(i)})} q_t(\tilde{X}_{t,t}^{(i)}|\tilde{X}_{t,t-1}^{(i)})}.$$

Normalise such that $\sum_{i=1}^{N} W_t^{(i)} = 1$.
10: Resample: for $i = 1 : N$, sample

$$X_{t,1:t,}^{(i)} \sim \frac{\sum_{i=1}^{N} W_1^{(i)} \delta_{\tilde{X}_{t,1:t}^{(i)}}}{\sum_{j=1}^{N} W_1^{(j)}}.$$

11: The smoothing and filtering distributions at time t may be approximated with

$$\hat{\pi}_{SIS,t}^{N}(x_{1:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{t,1:t}^{(i)}}(x_{1:t}), \qquad \text{and } \hat{\pi}_{SIS,t}^{N}(x_t) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{t,t}^{(i)}}(x_t), \qquad \text{respectively.}$$

12: Go to step 5.

Algorithm 6 The APF as SIR

1: Set $t = 1$.
2: For $i = 1 : N$, sample $X_{1,1}^{(i)} \sim q_1(\cdot)$.
3: For $i = 1 : N$, set $W_1^{(i)} \propto \pi_1(X_{1,1}^{(i)}) g_1(y_1|X_{1,1}^{(i)})/q_1(X_1^{(i)})$.
4: $t \leftarrow t + 1$
5: Set $\tilde{W}_t^{(i)} \propto W_{t-1}^{(i)} \lambda_t^{(i)}$ where $\lambda_t^{(i)} \propto \hat{g}_t(y_t|X_{t-1,t-1}^{(i)})$ and normalise such that $\sum_{i=1}^{N} \tilde{W}_t^{(i)} = 1$.
6: Resample: for $i = 1 : N$, sample

$$X_{t,1:t-1}^{(i)} \sim \frac{\sum_{i=1}^{N} \tilde{W}_t^{(i)} \delta_{X_{t-1,1:t-1}^{(i)}}}{\sum_{j=1}^{N} \tilde{W}_t^{(j)}}.$$

7: For $i = 1 : N$, sample $X_{t,t}^{(i)} \sim q_t(\cdot|X_{t,t-1}^{(i)})$.
8: For $i = 1 : N$, set $W_t^{(i)} \propto f_t(X_{t,t}^{(i)}|X_{t,t-1}^{(i)}) g_t(y_t|X_{t,t}^{(i)})/q_t(\tilde{X}_{t,t}^{(i)}|\tilde{X}_{t,t-1}^{(i)}) \hat{g}_t(y_t|X_{t,t-1}^{(i)})$. Normalise such that $\sum_{i=1}^{N} W_t^{(i)} = 1$.
9: The smoothing and filtering distributions at time t may be approximated with

$$\hat{\pi}_{SIS,t}^{N}(x_{1:t}) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{t,1:t}^{(i)}}(x_{1:t}), \qquad \text{and } \hat{\pi}_{SIS,t}^{N}(x_t) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{t,t}^{(i)}}(x_t), \qquad \text{respectively.}$$

10: Go to step 5.

these as an importance sampling proposal distribution in order to estimate quantities of interest (Johansen and Doucet, 2007). Although the precise formulation shown in algorithm 6 appears to differ in some minor details from algorithm 5 it corresponds exactly to the formulation of the APF which is most widely used in practice and which can be found throughout the literature. The significant point is that this type of filtering algorithm has an interpretation as a simple SIR algorithm with an importance sampling step and can be analysed within the same framework as the other algorithms presented within this chapter.

10.5.4 A Note on Asymptotic Variances

It is often useful in estimation scenarios based upon sampling to compare the asymptotic variance of those estimators which are available. In the context of sequential Monte Carlo methods a substantial amount of calculation is required to obtain even asymptotic variance expressions and these can be difficult to interpret. In the case of the standard SIS algorithm one is performing standard importance sampling and so the asymptotic variance can be obtained directly by standard methods, see (Geweke, 1989), for example. In the case of SIR and the APF a little more subtlety is required. Relatively convenient variance decompositions for these algorithms may be found in (Johansen and Doucet, 2007), which makes use of expressions applicable to the SIR case obtained by (Del Moral, 2004; Chopin, 2004).

10.6 Static Parameter Estimation

The difficulty with static parameter estimation becomes clear when considered in light of the information acquired in studying the filtering and smoothing problems: although we are able to obtain a good characterisation of the filtering distribution, the smoothing distribution is much more difficult to deal with. Typically, degeneracy occurs towards the beginning of the trajectory and the estimated distribution of the earlier coordinates of the trajectory are extremely poor. In order to estimate the distribution of a static parameter within a HMM, it is generally necessary to also estimate the distribution of the latent state sequence. The posterior distribution of the static parameter is heavily dependent upon the distribution of the full sequence of latent variables and the aforementioned degeneracy problems lead to severe difficulties when attempting to perform online parameter estimation within this framework.

A heuristic approach which has been advocated in some parts of the literature is to modify the model in such a way that the static parameter is replaced with a slowly time-varying parameter. This introduces a degree of forgetting and it is possible to produce a model in which the time-scale over which degeneracy becomes a problem is less than that over which the parameter's dynamics allow it to forget the past. However, this is clearly rather unsatisfactory and it is far from clear how inference based upon this model relates to that of interest.

A more principled approach was suggested by (Chopin, 2002). His approach was based upon something which he termed *artificial dynamics*. Essentially, the static parameter is updated according to a Markov kernel of the correct invariant distribution after each iteration of the algorithm, allowing it to change. This overcomes one problem: specifically that otherwise no new values of the parameter would ever be introduced into the algorithm, but does not eliminate all of the problems which arise from the ultimate degeneracy of the path-space particle distribution.

As of the time of writing, no entirely satisfactory solution to the problem of online estimation of static parameters in a HMM via their posterior mode has been proposed, although a number of techniques which provide some improvement have been developed in recent years.

10.7 Extensions, Recent Developments and Further Reading

Much recent work has focused upon the problem of sampling from *arbitrary* sequences of distributions using the techniques developed in the SMC setting presented here. This work is interesting, but cannot be included in the present course due to time constraints; the interested reader is referred to (Del Moral et al., 2006a,b) which present a sophisticated method which include other approaches which have been proposed recently (Neal, 1998) as particular cases.

An introductory, application-oriented text on sequential Monte Carlo methods is provided by the collection (Doucet et al., 2001). Short, but clear introductions are provided by (Doucet et al., 2000) and (Robert and Casella, 2004, chapter 14).

A simple self-contained and reasonably direct theoretical analysis providing basic convergence results for standard SMC methods of the sort presented in this chapter can be found in (Crisan and Doucet, 2002). Direct proofs of central limit theorems are given by (Chopin, 2004; Künsch, 2005). A comprehensive mathematical treatment of a class of equations which can be used to analyse sequential Monte Carlo methods with some degree of sophistication is provided by (Del Moral, 2004) – this is necessarily somewhat technical, and requires a considerably higher level of technical sophistication than this course; nonetheless, the treatment provided is lucid, self-contained and comprehensive incorporating convergence results, central limit theorems and numerous stronger results.

Bibliography

Badger, L. (1994) Lazzarini's lucky approximation of π. *Mathematics Magazine*, **67**, 83–91.

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.

Cappé, O., Moulines, E. and Rydén, T. (2005) *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer.

Chopin, N. (2002) A sequential particle filter method for static models. *Biometrika*, **89**, 539–551.

— (2004) Central limit theorem for sequential Monte Carlo methods and its applications to Bayesian inference. *Annals of Statistics*, **32**, 2385–2411.

Crisan, D. and Doucet, A. (2002) A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, **50**, 736–746.

Del Moral, P. (2004) *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Probability and Its Applications. New York: Springer.

Del Moral, P., Doucet, A. and Jasra, A. (2006a) Sequential Monte Carlo methods for Bayesian Computation. In *Bayesian Statistics 8*. Oxford University Press.

— (2006b) Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society B*, **63**, 411–436.

Doucet, A., de Freitas, N. and Gordon, N. (eds.) (2001) *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. New York: Springer.

Doucet, A., Godsill, S. and Andrieu, C. (2000) On sequential simulation-based methods for Bayesian filtering. *Statistics and Computing*, **10**, 197–208.

Doucet, A., Godsill, S. J. and Robert, C. P. (2002) Marginal maximum a posteriori estimation using Markov chain Monte Carlo. *Statistics and Computing*, **12**, 77–84.

Fahrmeir, L. and Tutz, G. (2001) *Multivariate Statistical Modelling Based on Generalised Linear Models*. New York: Springer, 2 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.

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. and Berzuini, C. (2001a) Following a moving target – Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society B*, **63**, 127–146.

— (2001b) RESAMPLE-MOVE filtering with Cross-Model jumps. In Doucet et al. (2001), 117–138.

Gilks, W. R., Richardson, S. and Spieghalter, D. J. (eds.) (1996) *Markov Chain Monte Carlo In Practice*. Chapman and Hall, first edn.

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. (2008) Markov chains. In *Encyclopaedia of Computer Science and Engineering*. Wiley and Sons. In preparation.

Johansen, A. M. and Doucet, A. (2007) Auxiliary variable sequential Monte Carlo methods. *Tech. Rep. 07:09*, University of Bristol, Department of Mathematics – Statistics Group, University Walk, Bristol, BS8 1TW, UK.

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.

Kong, A., Liu, J. S. and Wong, W. H. (1994) Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, **89**, 278–288.

Künsch, H. R. (2005) Recursive Monte Carlo filters: Algorithms and theoretical analysis. *Annals of Statistics*, **33**, 1983–2021.

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.

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.

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/`.

Neal, R. M. (1998) Annealed importance sampling. *Technical Report 9805*, University of Toronto, Department of Statistics. URL `ftp://ftp.cs.toronto.edu/pub/radford/ais.ps`.

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.

Pitt, M. K. and Shephard, N. (1999) Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, **94**, 590–599.

— (2001) Auxiliary variable based particle filters. In Doucet et al. (2001), chap. 13, 273–293.

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.

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.