Monte Carlo:
Simulation Methods for Statistical Inference

Sinan Yıldırım

December 22, 2017

Contents

List of Figures

List of Tables

List of Abbreviations

a.s.	almost surely
cdf	cumulative distribution function
HMM	hidden Markov model
i.i.d.	independently and identically distributed
IS	Importance sampling
MAP	maximum a posteriori
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
MLE	maximum likelihood estimate (or estimation)
MMSE	minimum mean square error
MSE	mean square error
pdf	Probability density function
pmf	Probability mass function
SIS	sequential importance sampling
SISR	sequential importance sampling resampling
SMC	sequential Monte Carlo

Chapter 1

Introduction

Summary: *This chapter provides a motivation for Monte Carlo methods. We will basically discuss averaging, which is the core of Monte Carlo integration. Then we will discuss theoretical and practical justifications of Monte Carlo. The chapter ends with a toy example, Buffon's needle experiment.*

1.1 Sample averages

Suppose we are given $N \geq 1$ *random samples* $X^{(1)}, \ldots, X^{(N)}$, each taking values from a set $\mathcal{X} \subset \mathbb{R}^{d_x}$ for some $d_x \geq 1$. The samples are *independent and identically distributed* (i.i.d.) according to some distribution P for a random variable X. We summarise this sentence as

$$X^{(1)}, \ldots, X^{(N)} \overset{\text{i.i.d.}}{\sim} P.$$

Also, the distribution P is unknown.

Mean value of the distribution: First, we are asked to provide an estimate of the mean value i.e. the expectation of X with respect to P using the sample set $X^{(1:N)}$. If P has a probability density function $p(x)$, this expectation can be written as[1]

$$\mathbb{E}_P(X) = \int_{\mathcal{X}} xp(x)dx. \tag{1.1}$$

A reasonable estimate of this quantity would be

$$\mathbb{E}_P(X) \approx \frac{1}{N} \sum_{i=1}^{N} X^{(i)}. \tag{1.2}$$

Expectation of a general function: Next, we are asked to provide an estimate of the expectation of a certain function $\varphi : \mathcal{X} \to \mathbb{R}$ with respect to P, that is[2]

$$P(\varphi) := \mathbb{E}_P(\varphi(X)) = \int_{\mathcal{X}} \varphi(x)p(x)dx. \tag{1.3}$$

[1] We assume continuity of X to avoid repetitions for continuous and discrete variables. For discrete distributions change the integral to the sum $\sum_i x_i p(x_i)$.

[2] Again, for discrete variables this would be $\sum_i \varphi(x_i)p(x_i)$.

(Here, the notation $P(\varphi)$ is introduced for simplicity.) This time, we replace our estimator with one that has the values of the φ evaluated at the samples $\varphi(X^{(i)})$, $i = 1, \dots, N$ instead of $X^{(i)}$ themselves.

$$\mathbb{E}_P(\varphi(X)) \approx \frac{1}{N} \sum_{i=1}^{N} \varphi(X^{(i)}). \tag{1.4}$$

It is easy to see that the second problem is just a simple generalisation of the first: Put $\varphi(X) = X$ and you will come back to the first problem, which was to estimate the expected value of X. The function φ can correspond to another moment of interest, for example $\varphi(x) = x^2$, or a specific function of interest, for example $\varphi(x) = \log x$.

Probability of a set: Another special case of φ is seen when we are interested in the probability of a certain set $A \subseteq \mathcal{X}$. How do we write this probability

$$P(A) := \mathbb{P}(X \in A)$$

as an expectation of a function with respect to P? For this, consider the *indicator function* $\mathbb{I}_A : \mathcal{X} \to \{0, 1\}$ such that

$$\mathbb{I}_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \tag{1.5}$$

Now, let us consider $\varphi = \mathbb{I}_A$ and write the expectation of this function with respect to P:

$$\begin{aligned} \mathbb{E}_P(\mathbb{I}_A(X)) &= \int_{\mathcal{X}} \mathbb{I}_A(x) p(x) dx \\ &= \int_A p(x) dx \\ &= \mathbb{P}(X \in A). \end{aligned} \tag{1.6}$$

where the second line follows from the fact that the integrand becomes $p(x)$ for $x \in A$ and 0 for $x \notin A$. Therefore, we know what to do when $X^{(1)}, \dots, X^{(N)} \overset{\text{i.i.d.}}{\sim} P$ are given and we want to estimate $\mathbb{P}(X \in A) = \mathbb{E}_P(\mathbb{I}_A(X))$: we simply apply equation (1.4) for the function $\mathbb{I}_A(X)$:

$$\mathbb{P}(X \in A) \approx \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_A(X^{(i)}). \tag{1.7}$$

Notice that this will output a value that is guaranteed to be in $[0, 1]$ (since the sum can be at least 0 and at most N), so it is a valid probability estimate.

In the following, we will talk in general about the expectation (1.3) and its estimate (1.4), after hopefully having convinced you that the other expectations and their estimates are just special cases.

1.2 Monte Carlo: Generating your own samples

See the equation (1.4); this is what we would do to estimate a certain quantity that is to do with the distribution P. However, notice that you do not need to know anything explicit about P in order to make that calculation. All you need are the samples $X^{(1:N)}$ from P. Perhaps we would be able to calculate the exact value of (1.3) if P were known. However, we said P was unknown, and we proposed to use the estimate (1.4) instead.

Now consider a new scenario: This time you *do* know P, at least up to a certain extent; but you are not given any samples from P. The following are what you can and cannot do about P:

- You *can* generate (draw) i.i.d. samples from P, as many as you want.
- You *cannot* compute (one or more of) the integral in (1.3), or you can only compute it in a very very long time - so long that you do not want to! Another way of saying this is that the integral is *intractable*.

What would you do to estimate (1.3) in that case? Of course, by generating your own samples $X^{(1)}, \dots, X^{(N)}$ from P so that the problem reduces to the one in the previous section.[3] This simple idea is the core of Monte Carlo methods. Once you generate samples from P, you do not need to deal with it in order to implement the estimate (1.4).

The term Monte Carlo was coined in the 1940s, see Metropolis and Ulam (1949) for a first use of the term, and Metropolis (1987); Eckhardt (1987) for a historical review.

1.2.1 Justification of Monte Carlo

Let $P^N_{\text{MC}}(\varphi)$ denote the Monte Carlo estimate of $P(\varphi)$ in (1.3) that is given in (1.4) using n samples, i.e.

$$P^N_{\text{MC}}(\varphi) = \frac{1}{N} \sum_{i=1}^{N} \varphi(X^{(i)}). \tag{1.8}$$

[3] An engineer is asked how to make tea using an empty kettle, a tea bag, a cup, and tap water. The engineer explained that first she would pour water in the kettle, then boil it, pour it in a cup, put the tea bag in the cup, and wait until the tea brews. Next, a mathematician is asked how to make tea using a kettle full of water, a tea bag, a cup, and tap water. The mathematician proposed to empty the kettle first so that they are back to the first problem.

It is easy to show that $P_{\mathrm{MC}}^N(\varphi)$ is an unbiased estimator of $P(\varphi)$ for any $N \geq 1$:

$$\begin{aligned}\mathbb{E}\left(P_{\mathrm{MC}}^N(\varphi)\right) &= \mathbb{E}\left(\frac{1}{N}\sum_{i=1}^N \varphi(X^{(i)})\right). \\ &= \frac{1}{N}\sum_{i=1}^N \mathbb{E}_P(\varphi(X^{(i)})) \\ &= \frac{1}{N} N\mathbb{E}_P(\varphi(X)) \\ &= \mathbb{E}_P(\varphi(X)) = P(\varphi).\end{aligned}$$

However, unbiasedness itself is not enough.[4] Fortunately, we have results on the convergence and decreasing variance as N increases.

Law of large numbers: If $|P(\varphi)| < \infty$, the *law of large numbers* (e.g. Shiryaev (1995), p. 391) ensures almost sure (a.s.) convergence of $P_{\mathrm{MC}}^N(\varphi)$ to $P(\varphi)$ as the number of i.i.d. samples tends to infinity,

$$P_{\mathrm{MC}}^N(\varphi) \overset{a.s.}{\to} P(\varphi), \quad \text{as } N \to \infty.$$

Central limit theorem: The variance of $P_{\mathrm{MC}}^N(\varphi)$ is given by

$$\mathbb{V}\left[P_{\mathrm{MC}}^N(\varphi)\right] = \frac{1}{N^2}\sum_{i=1}^N \mathbb{V}_P\left[\varphi(X^{(i)})\right] = \frac{1}{N}\mathbb{V}_P\left[\varphi(X)\right].$$

which indicates the improvement in the accuracy with increasing N, provided that $\mathbb{V}_P\left[\varphi(X)\right]$ is finite. Also, if $\mathbb{V}_P\left[\varphi(X)\right]$ is finite, the distribution of the estimator is well behaved in the limit, which is ensured by the *central limit theorem* (e.g. Shiryaev (1995), p. 335)

$$\sqrt{N}\left[P_{MC}^N(\varphi) - P(\varphi)\right] \overset{d}{\to} \mathcal{N}\left(0, \mathbb{V}_P\left[\varphi(X)\right]\right) \quad \text{as } N \to \infty.$$

Advantage over deterministic integration: There are deterministic numerical integration techniques available to approximate $P(\varphi)$; however these methods encounter the problem called the *curse of dimensionality* since the amount of computation grows exponentially with the dimension of X, d_x (Press, 2007). Therefore, they are far from being practical and reliable unless they work in low dimensional problems. Monte Carlo integration is a powerful alternative to deterministic methods for integration problems. Compared to deterministic numerical integration algorithms, the performance of Monte Carlo does not depend on the dimension d_x (check the variance of the Monte Carlo estimate above, which does not depend on d_x). This makes the method particularly useful for high dimensional integrations (Newman and Barkema, 1999).

[4] Even $\varphi(X^{(1)})$ is an unbiased estimate of $P(\varphi)$ but it is 'somewhat' inferior than taking the average over N samples.

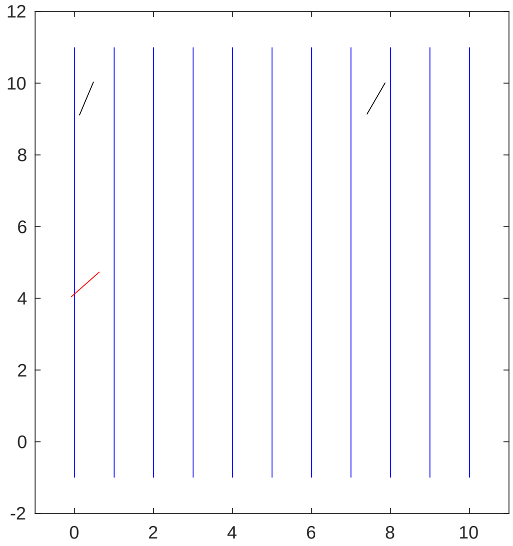


Figure 1.1: Buffon's needles: 3 throws

1.2.2 Toy example: Buffon's needle

This is an illustrative example for the use of Monte Carlo. In mathematics, Buffon's needle problem is a question first posed in the 18'th century by Georges-Louis Leclerc, Comte de Buffon. Suppose we have a floor made of parallel strips of wood, each the same width, and we drop a needle onto the floor. What is the probability that the needle will lie across a line between two strips?

Buffon's needle was the earliest problem in geometric probability to be solved; it can be solved using integral geometry. The solution, in the case where the needle length is not greater than the width of the strips, can be used to design a Monte Carlo method for approximating the number π, (although that was not the original motivation for de Buffon's question).

First, let us try to answer the initial question: What is the probability of the needle of length 1 (without loss of generality) crossing a line between the strips of width 1 if the location and the direction of the needle are independent and uniformly distributed? The probability can actually be calculated: Let d be the distance from middle of the needle to the nearest line and θ be the acute angle between the parallel lines and the needle (between 0 and $\pi/2$). A needle touches a line if and only if

$$\frac{d}{\sin\theta} < \frac{1}{2} \tag{1.9}$$

Try to verify this by observing the needles Figure 1.1. The variables d, θ are independent and uniformly distributed in $[0, 1/2]$ and $[0, \pi/2]$, respectively, so that their joint probability

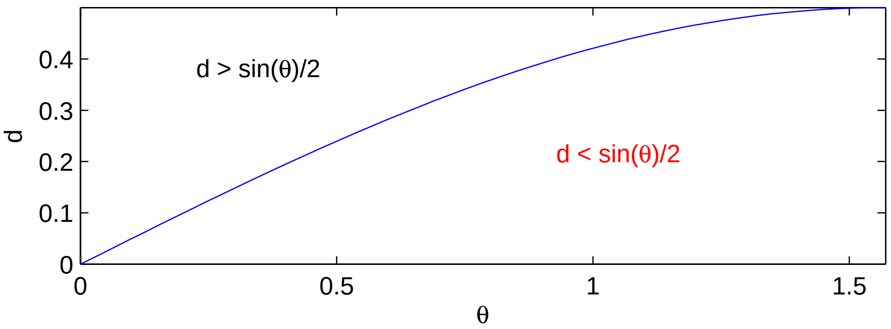


Figure 1.2: The set A that corresponds to the needle crossing a line

density can be written as

$$p(d,\theta) = \begin{cases} \frac{4}{\pi}, & (d,\theta) \in [0,1/2] \times [0,\pi/2] \\ 0, & \text{else} \end{cases}$$

Now, define the set $A = \{(d,\theta) : d/\sin\theta < 1/2\} = \{(d,\theta) : d < \sin\theta/2\}$. The set A corresponds to the area under the curve in Figure 1.2. Letting $X = (d,\theta)$, the required probability is

$$\begin{aligned} \mathbb{P}(X \in A) &= \int\int_A p(r,\theta) dr d\theta \\ &= \int_0^{\pi/2} d\theta \int_0^{\sin(\theta)/2} \frac{4}{\pi} dr \\ &= \frac{2}{\pi} \int_0^{\pi/2} d\theta \sin(\theta) \\ &= \frac{2}{\pi}[-\cos(\pi/2) + \cos(0)] \\ &= \frac{2}{\pi} \qquad (1.10) \end{aligned}$$

where the dummy variable r is used for d (just to avoid writing dd in the integral!).

Monte Carlo approximation: Suppose it is not our day and we cannot carry out calculation in (1.10). In order to find $\mathbb{P}(X \in A)$, we decide to run a Monte Carlo experiment instead. Let $X = (d,\theta)$ and observe $\mathbb{P}(X \in A) = \mathbb{E}(\mathbb{I}_A(X))$. The idea is to generate samples $X^{(i)} = (d^{(i)}, \theta^{(i)})$, $i = 1, \ldots, N$ where each sample is generated independently as

$$d^{(i)} \sim \text{Unif}(0, 1/2), \quad \theta^{(i)} \sim \text{Unif}(0, \pi/2), \qquad (1.11)$$

and estimate $\mathbb{P}(X \in A)$ as

$$\begin{aligned}\mathbb{P}(X \in A) &\approx \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_A(X^{(i)}) \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(d^{(i)} < \sin(\theta^{(i)})/2) \qquad (1.12)\end{aligned}$$

(where we have introduced another notation-wise use of the indicator function $\mathbb{I}$). In words, for each sample $X^{(i)} = (d^{(i)}, \theta^{(i)})$ we check whether $d^{(i)} < \sin(\theta^{(i)})/2$ (in other words, we check whether $X^{(i)} \in A$) and we divide the number of samples satisfying this condition by the total number samples n. Observe that this is the implementation of (1.7) for this problem. The samples $(d^{(i)}, \theta^{(i)})$ can be generated by throwing a needle on a table with parallel lines - or using a computer! Figure 1.3 shows the described Monte Carlo experiment performed with $N = 100$ throws. The d, θ values corresponding to these throws are shown in Figure 1.3. Note that (1.12) is

$$\mathbb{P}(X \in A) \approx \frac{\text{number of red dots}}{\text{total number of dots}}.$$

The *law of large numbers* says that as n tends to infinity the estimate above converges to the true value $\mathbb{P}(X \in A) = 2/\pi$. This fact can be 'felt' by observing the Figure 1.4 which show the results of the same Monte Carlo experiment with larger N values. The estimate of $\mathbb{P}(X \in A)$ improves with N.

Estimating π: We have already stated that $\mathbb{P}(X \in A)$ is known for this problem and in fact it is $2/\pi$. We have also described a Monte Carlo experiment to estimate this value. With a little modification, our estimate can be used to estimate π. Since we have

$$\pi = \frac{2}{\mathbb{P}(X \in A)},$$

we can approximate π by

$$\begin{aligned}\pi &\approx 2 \times \frac{N}{\sum_{i=1}^{N} \mathbb{I}(d^{(i)} < \sin(\theta^{(i)})/2)} \\ &= 2 \times \frac{\text{total number of dots}}{\text{number of red dots}}\end{aligned}$$

This is a pretty fancy way of estimating π with a needle and a table! Figure 1.5 shows the estimated value versus the number of samples n. Again, we see improvement in the estimate as N increases.

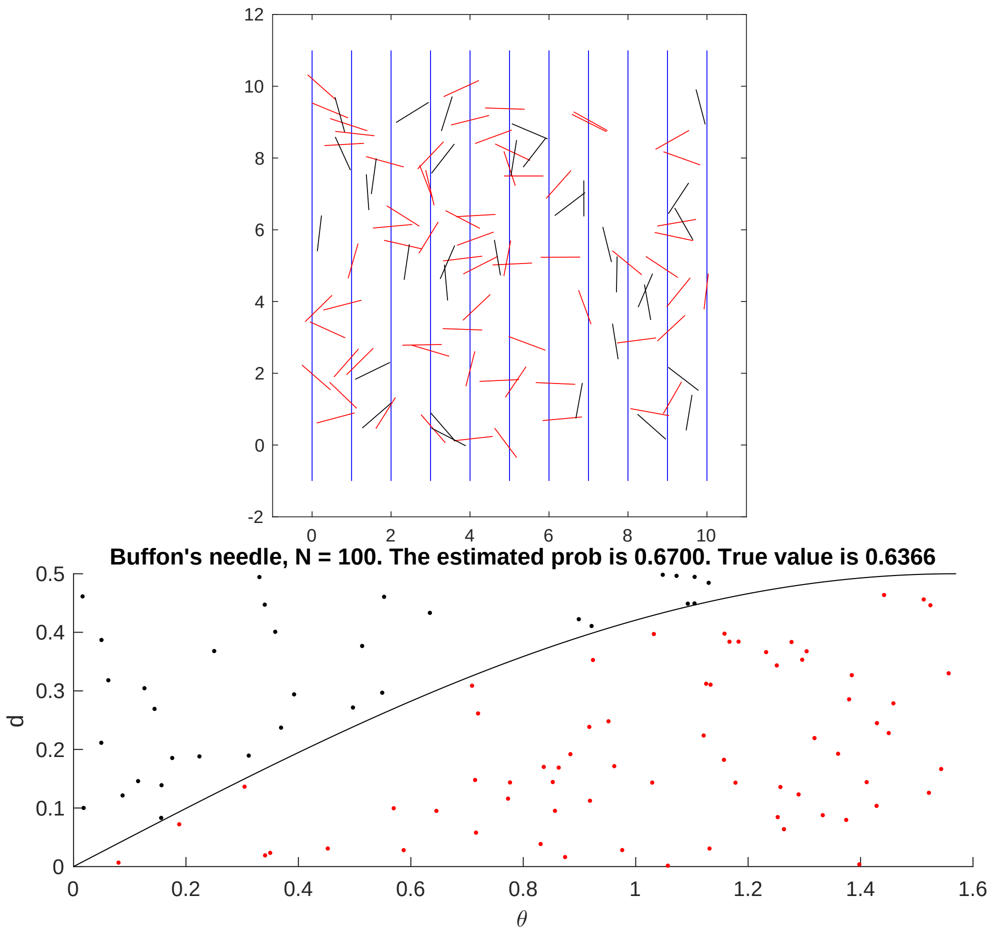


Figure 1.3: Top: Buffon's needle experiment with 100 independent throws. Bottom: (d, θ) values of the needle throws

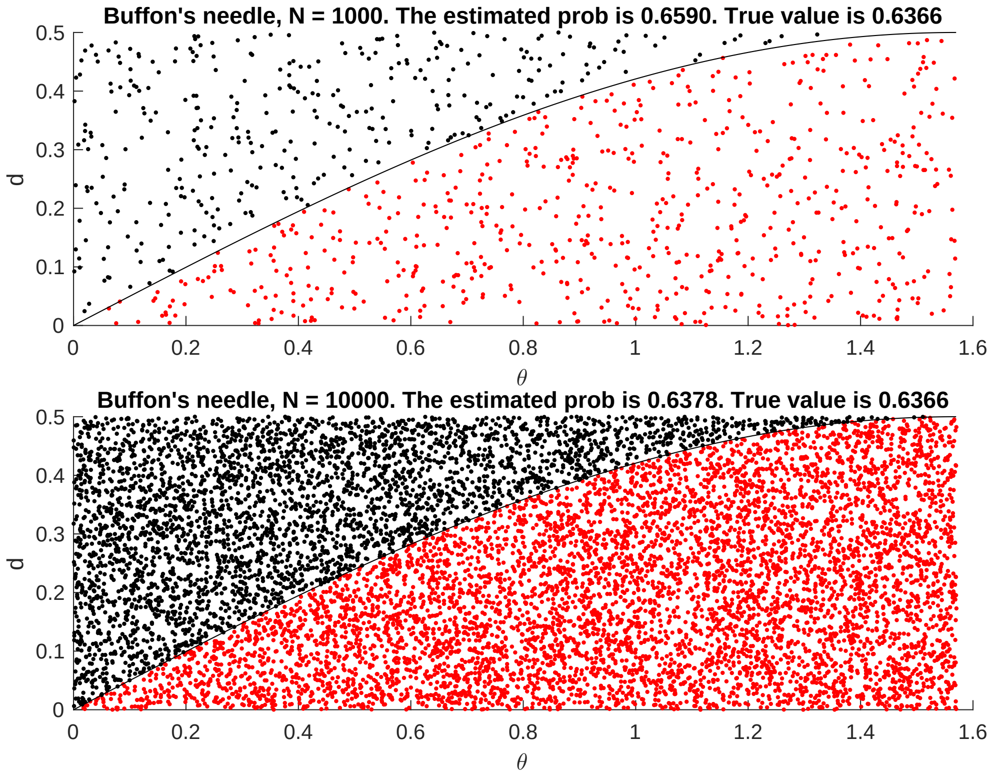


Figure 1.4: (d, θ) values of $N = 1000$ and $N = 10000$ independent needle throws

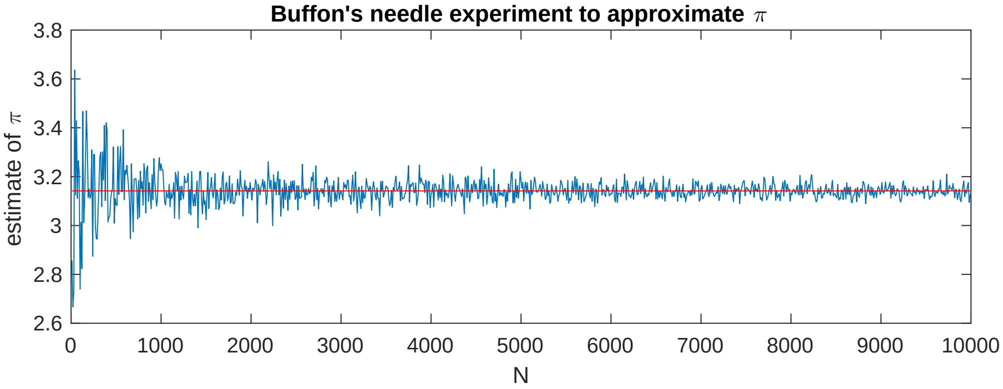


Figure 1.5: Approximating π with Buffon's needle experiment. The plot shows the value of the estimate versus the number of needle throws, N.

1.2.3 The need for more sophisticated methods

The distribution P in the toy example was the product of two uniform distributions for $X = (d, \theta)$. However, in many problems P is not always trivial to sample from. In the rest of the course, we will see some methods to generate exact samples from P (that is, samples that are exactly distributed according to P). These are the inverse transform method and the rejection sampling method.

However, the story does not end there: Being able to sample from the distribution of interest exactly is *rarely* the case when it comes to real applications in engineering and science. Especially in Bayesian statistics, where the distribution we want to sample from is the posterior distribution of some variable X given $Y = y$, which is of the form

$$\begin{aligned} p_{X|Y}(x|y) &= \frac{p_X(x)p_{Y|X}(y|x)}{\int p_X(x')p_{Y|X}(y|x')dx'} = \frac{p_{X,Y}(x,y)}{\int p_{X,Y}(x',y)dx'} \\ &\propto p_X(x)p_{Y|X}(y|x) \end{aligned}$$

it is either too costly or impossible to perform exact sampling. That explains the vast amount of literature on sophisticated Monte Carlo methods that aim to generate *approximate* samples. In the course, we will cover some of these methods. Among them, importance sampling and Markov chain Monte Carlo methods are worth mentioning as early as here.

Chapter 2

Exact Sampling Methods

Summary: *In order to obtain estimates as in (1.8), we need exact i.i.d. samples from P, that is samples that are exactly distributed from P. This chapter describes some exact sampling methods. These methods are the method of inversion, transformation, composition, and rejection sampling*

2.1 Pseudo-random number generation

"The generation of random numbers is too important to be left to chance" and truly random numbers are impossible to generate on a deterministic computer. Published tables or other mechanical methods such as throwing dice, flipping coins, shuffling cards or turning the roulette wheels are clearly not very practical for generating the random numbers that are needed for computer simulations. Other techniques rely on chaotic behaviour, such as the thermal noise in Zener diodes or other analog circuits as well as the atmospheric noise (see, e.g., `www.Random.org`) or running a hash function against a frame of a video stream. Still, the vast amount of random numbers are obtained from pseudo-random number generators. Apart from being very efficient, one additional advantage of these techniques is that the sequences are reproducible by setting a *seed*, this property is key for debugging a Monte Carlo code.

2.1.1 Pseudo-random number generators for Unif$(0, 1)$

Today, in most applications the task of random variable generation is performed on computers. In fact, a computer is mainly responsible for generating pseudo-random numbers that *look as if* they are independent and distributed uniformly from between 0 and 1, so goes the name "pseudo-random". That is, any sequence of pseudo-random numbers that are produced by the pseudo-random number generator should look like a sequence of i.i.d. uniformly distributed random numbers between 0 and 1, showing no correlation and spreading over the $(0, 1)$ interval uniformly.

There already exist highly sophisticated numerical methods to generate such pseudo-random numbers that pass certain tests for uniformity and independence. The most well known method for generating random numbers is based on a Linear Congruential Generator (LCG). The theory is well understood, and the method is easy to implement and fast. A

LCG is defined by the recurrence relation:

$$x_{n+1} = (ax_n + c)(\text{mod } M)$$

If the coefficients a and c are chosen carefully (e.g. relatively prime to M), x_n will be roughly uniformly distributed between 0 and $M - 1$ (and with normalisation by M they can be shrunk between 0 and 1). By "roughly uniformly" we mean that the sequence of numbers x_n will pass many reasonable tests for randomness. One such test suite are the so called DIEHARD tests, developed by George Marsaglia, that are a battery of statistical tests for measuring the quality of a random number generator.

A more recently proposed generator is the Mersenne Twister algorithm, by Matsumoto and Nishimura, 1997. It has several desirable features such as a long period and being very fast. Many public domain implementations of the algorithm exist and it is the preferred random number generator for statistical simulations and Monte Carlo computations.

2.2 Some exact sampling methods

In the sequel, we will assume that a computer can produce for us an independent variable

$$U \sim \text{Unif}(0, 1)$$

every time we ask it to do so. The crucial part is how to transform one or more copies of U such that the resulting number is distributed according to a particular distribution that we want to sample from. In a more general context, how can one exploit the ability of the computer to generate uniform random variables so that we can obtain random numbers from any desired distribution?

In the following we will see some exact sampling methods.

2.2.1 Method of inversion

Suppose $X \sim P$ taking values in $\mathcal{X} \subseteq \mathbb{R}$ with cdf F as defined above: $F(x) = \mathbb{P}(X \leq x)$, $x \in \mathbb{R}$. Recall that F takes values in $[0, 1]$. Define the *generalised inverse cdf* $G : (0, 1) \to \mathbb{R}$ as

$$G(u) := \inf\{x \in \mathcal{X} : F(x) \geq u\}. \tag{2.1}$$

Remark 2.1. *Define the set $S(u) = \{x \in \mathcal{X} : F(x) \geq u\}$. We can show that, by right-continuity of F, $S(u)$ actually attains its infimum, that is the minimum of $S(u)$ exists and hence* $\inf S(u) = \min S(u)$, *or* $S(u) = [G(u), \infty)$[1].

[1] Proof: If $x < G(u)$, $x \notin S(u)$ by definition. If $x > G(u)$, then there exists $x' < x$ with $x' \in S(u)$; since $F$is non-decreasing, $F(x) \geq F(x') \geq u$, so $x \in S(u)$. Finally, by the right-continuity of F, we have $F(G(u)) = \inf F(y) : y > G(u) \geq u$. Therefore $G(u) \in S(u)$ and $S(u) = [G(u), \infty)$

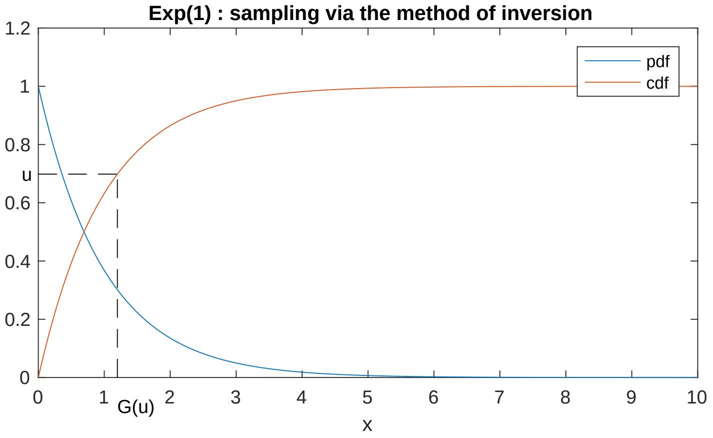


Figure 2.1: Method of inversion for the exponential distribution

If X is discrete taking values $x_1, x_2, \ldots$, this definition reduces to $G(u) = x_{i^*}$ where $i^* = \min\{i : F(x_i) \geq u\}$. In other words, $G(u) = x_{i^*}$ such that

$$F(x_{i^*-1}) < u \leq F(x_{i^*}). \tag{2.2}$$

If X is continuous with a pdf $p(x) > 0$ for all $x \in \mathcal{X}$, (i.e. F has no jumps and no flat parts in $\mathcal{X}$), then F is strictly monotonic in $\mathcal{X}$, its inverse $G = F^{-1}$ can be defined on $\mathcal{X}$, and we simply have $G(u) = F^{-1}(u)$.

The following Lemma enables the method of inversion.

Lemma 2.1. *If* $U \sim \mathrm{Unif}(0, 1)$, $G(U) \sim P$

Proof. Since $S(u) = [G(u), \infty)$ (see Remark 2.1), we have $x \geq G(u)$ if and only if $F(x) \geq u$. Hence, $\mathbb{P}(X \leq x) = \mathbb{P}(G(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x)$. □

Lemma 2.1 suggests we can sample $X \sim P$ by first sampling $U \in \mathrm{Unif}(0, 1)$ and then transforming $X = G(U)$. This approach is called the method of inversion. It was considered by Ulam prior to 1947 (Eckhardt, 1987) and some extensions to the method are provided by Robert and Casella (2004).

Corollary 2.1. *Suppose F is continuous. If $X \sim P$, then* $F(X) \sim \mathrm{Unif}(0, 1)$.

Proof. Since we have $S(u) = [G(u), \infty)$, $x \geq G(u)$ implies $F(x) \geq u$. Moreover, if $x < G(u)$ then $F(x) < u$ by definition of G. By continuity of F, we have $F(G(u)) = u$, so $F(x) \leq u$ if and only if $x \leq G(u)$. Hence $\mathbb{P}(F(X) \leq u) = \mathbb{P}(X \leq G(u)) = F(G(u)) = u$, and we conclude that the cdf of $F(X)$ is the cdf of $\mathrm{Unif}(0, 1)$. □

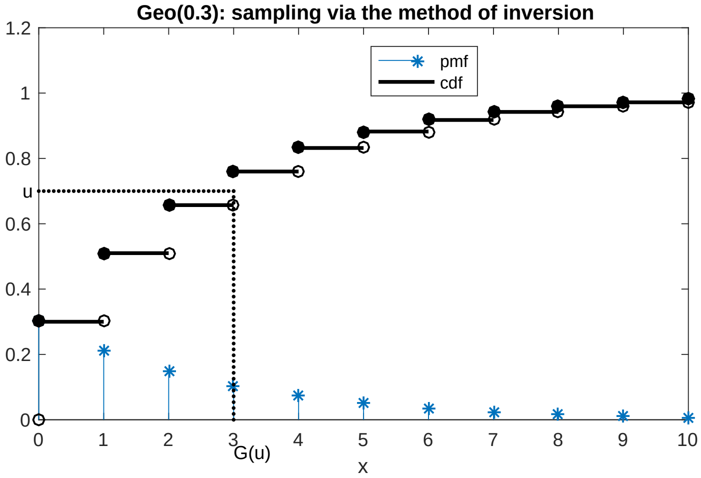


Figure 2.2: Method of inversion for the geometric distribution

Example 2.1. *Suppose we want to sample* $X \sim P = \text{Exp}(\lambda)$ *from the exponential distribution with rate parameter* $\lambda > 0$. *The pdf of* $\text{Exp}(\lambda)$ *is*

$$p(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & else \end{cases}.$$

The cdf is

$$u = F(x) = \begin{cases} \int_0^x \lambda e^{-\lambda t} dt = 1 - e^{-\lambda x}, & x \geq 0 \\ 0, & else \end{cases}.$$

Therefore, we have $x = -\log(1-u)/\lambda$. *So, we can generate* $U \sim \text{Unif}(0,1)$ *and transform* $X = -\log(1-U)/\lambda \sim \text{Exp}(\lambda)$. *See Figure 2.1 for an illustration.*

Example 2.2. *Suppose we want to sample* $X \sim P = \text{Geo}(\rho)$ *from the geometric distribution on* $\mathcal{X} = \mathbb{N}$ *with success rate parameter* $\rho \in (0,1)$ *and pmf*[2]

$$p(x) = (1-\rho)^x \rho, \quad x = 0, 1, 2 \dots.$$

Making use of $\sum_{i=0}^{x} \alpha^i = \frac{1-\alpha^{x+1}}{1-\alpha}$ *with* $\alpha = 1 - \rho$, *the cdf at the support points is given by*

$$F(x) = 1 - (1-\rho)^{x+1}.$$

Given $U = u$ *sampled from* $\text{Unif}(0,1)$, *the rule in (2.2) implies*

$$1 - (1-\rho)^x < u \leq 1 - (1-\rho)^{x+1}$$

[2] This distribution is used for the number of trials prior to the first success in a Bernoulli process with success rate ρ. Another convention is to take the support range as $1, 2, \dots$ rather than $0, 1, 2$ and interpret X as the number of trials until the successful trial, including the successful one. Then the pmf changes to $p(x) = (1-\rho)^{x-1}\rho, x \geq 1$

Solving the inequality for x we arrive at

$$\frac{\log(1-u)}{\log(1-\rho)} - 1 \leq x < \frac{\log(1-u)}{\log(1-\rho)}.$$

This is nothing but the round-up function written explicitly:

$$x = \left\lceil \frac{\log(1-u)}{\log(1-\rho)} - 1 \right\rceil .$$

See Figure 2.2 for an illustration.

2.2.2 Transformation (change of variables)

The method of inversion can be seen as a transformation from U to $X = G(U)$. In fact, one can use transformation in a more general sense than using G by considering a change of variables via a suitable function g.

Example 2.3. *If we want to sample from $X \sim \mathrm{Unif}(a,b)$, $a < b$, we can sample $U \sim \mathrm{Unif}(0,1)$ and use the transformation*

$$X = g(U) := (b-a)U + a. \tag{2.3}$$

Transformation can also be used for more complicated situations than in Example 2.3. Suppose we have an m-dimensional random variable $X \in \mathcal{X} \subseteq \mathbb{R}^m$ with pdf $p_X(x)$ and we apply a transform to X using an invertible function $g : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{Y} \subseteq \mathbb{R}^m$ to obtain

$$Y = (Y_1, \ldots, Y_m) = g(X_1, \ldots, X_m)$$

Since g is invertible, we have $X = g^{-1}(Y)$. What is the pdf of Y, $p_Y(y)$? This density can be found as follows: Define the Jacobian determinant (or simply Jacobian) of the inverse transformation g^{-1} as

$$J(y) = \det \frac{\partial g^{-1}(y)}{\partial y} \tag{2.4}$$

The usual practice to ease the notation is to introduce the short hand notation $(y_1, \ldots, y_m) = g(x_1, \ldots, x_m)$ and write $J(y)$ by making implicit reference to g as

$$J(y) = \det \frac{\partial x}{\partial y} = \det \frac{\partial(x_1, \ldots, x_m)}{\partial(y_1, \ldots, y_m)} = \det \begin{bmatrix} \partial x_1/\partial y_1 & \ldots & \partial x_1/\partial y_m \\ \vdots & \ddots & \vdots \\ \partial x_m/\partial y_1 & \ldots & \partial x_m/\partial y_m \end{bmatrix}$$

The Jacobian is useful for integration: If we make a change of variables from $x \rightarrow y$, we have to substitute $dx = |J(y)|dy$. When we apply this for the integral of any function

$\varphi : \mathcal{X} \to \mathbb{R}$ with respect to $p_X(x)$, we have

$$\begin{aligned}\int p_X(x)\varphi(x)dx &= \int p_X(g^{-1}(y))\varphi(g^{-1}(y))\left|J(y)\right|dy \\ &= \int p_X(g^{-1}(y))\left|J(y)\right|\varphi(g^{-1}(y))dy \\ &= \int p_Y(y)\varphi(g^{-1}(y))dy\end{aligned}$$

where

$$p_Y(y) := p_X(g^{-1}(y))\left|J(y)\right| \tag{2.5}$$

is the pdf of Y.

Change of variables can be useful when P is difficult to sample from using the method of inversion but $X \sim P$ can be performed by a certain transformation of random variables that are easier to generate, such as uniform random variables.

Example 2.4. *We describe the Box-Muller method for generating random variables from the standard normal (Gaussian) distribution $\mathcal{N}(0,1)$. The pdf for $\mathcal{N}(\mu,\sigma^2)$ is*

$$\phi(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

The method of inversion is not an easy option to sample from $\mathcal{N}(0,1)$ since the cdf of $\mathcal{N}(0,1)$ is not easy to invert. Instead we use transformation.

The Box-Muller method generates a pair of independent standard normal random variables $X_1, X_2 \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ as follows: First we generate

$$R \sim \text{Exp}(1/2), \quad \Theta \sim \text{Unif}(0, 2\pi).$$

and then apply the transformation

$$X_1 = \sqrt{R}\cos(\Theta), \quad X_2 = \sqrt{R}\sin(\Theta)$$

If we wanted to start off from uniform random numbers, we could consider generating $U_1, U_2 \overset{i.i.d.}{\sim} \text{Unif}(0,1)$ and setting $R = -2\log(U_1)$ and $\Theta = 2\pi U_2$ so that R, Θ are distributed as desired. In other words,

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

One way to see why this works is to use change of variables. Note that[3]

$$(R,\Theta) = (X_1^2 + X_2^2, \arctan(X_2/X_1))) \tag{2.6}$$

[3] To be precise, $\Theta = \arctan(X_2/X_1) + \pi\mathbb{I}(X_1 < 0)$ since $\Theta \in [0, 2\pi]$, but omitting the extra term $\pi\mathbb{I}(X_1 < 0)$ does not change the results.

Then the Jacobean at $(x_1, x_2) = (\sqrt{r}\cos\theta, \sqrt{r}\sin\theta)$ *is*

$$J(x_1, x_2) = \begin{vmatrix} \partial r/\partial x_1 & \partial r/\partial x_2 \\ \partial\theta/\partial x_1 & \partial\theta/\partial x_2 \end{vmatrix} = \begin{vmatrix} 2x_1 & 2x_2 \\ \frac{1}{1+(y_2/y_1)^2}\frac{-y_2}{y_1^2} & \frac{1}{1+(y_2/y_1)^2}\frac{1}{y_1} \end{vmatrix} = 2 \tag{2.7}$$

Therefore, we can apply (2.5) *to get*

$$\begin{aligned} p_{X_1,X_2}(x_1, x_2) &= p_R(r)p_\Theta(\theta)|J(x_1, x_2)| \\ &= p_R(x_1^2 + x_2^2)p_\Theta(\arctan(x_2/x_1))|J(x_1, x_2)| \\ &= \frac{1}{2}e^{-\frac{1}{2}(x_1^2+x_2^2)}\frac{1}{2\pi}2 \\ &= \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x_1^2}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x_2^2} \\ &= \phi(x_1; 0, 1)\phi(x_2; 0, 1) \end{aligned} \tag{2.8}$$

which is the product of pdf of $\mathcal{N}(0, 1)$ *evaluated at* x_1 *and* x_2*. Therefore, we conclude that* $X_1, X_2 \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$*.*

Multivariate normal distribution: Another important transformation that we should be familiar with is a linear transformation of a multivariate normal random variable. We denote the distribution of an $n \times 1$ multivariable normal random variable as $X \sim \mathcal{N}(\mu, \Sigma)$ where $\mu = \mathbb{E}(X)$ is an $n \times 1$ *mean vector* and

$$\Sigma = \mathrm{Cov}(X) = \mathbb{E}[(X - \mu)(X - \mu)^T]$$

is an $n \times n$ symmetric positive definite[4] *covariance matrix* The (i, j)'th element of Σ is

$$\sigma_{ij} = \mathrm{Cov}(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] = \mathbb{E}(X_iX_j) - \mu_i\mu_j$$

The pdf of this distribution is (using the same letter as for the pdf of the univariate normal distribution)

$$\phi(x; \mu, \Sigma) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu)^T\Sigma^{-1}(x - \mu)\right\} \tag{2.9}$$

where $|\cdot|$ stands for determinant.

Suppose $X = (X_1, \dots, X_n)^T \sim \mathcal{N}(\mu, \Sigma)$ and we have the transformation

$$Y = AX + \eta$$

where A is an $m \times n$ matrix with $m \leq n$ with rank m[5], and η is an $m \times 1$ vector. We know for a fact that a linear transformation of X has to be normally distributed as well. Also,

[4] In fact, positive semi-definite covariance matrices are also allowed, however the distribution is called degenerate and it does not have a pdf.

[5] We constraint A to full row rank matrices since otherwise the resulting covariance matrix for $A\Sigma A^T$ is no longer positive definite and Y is degenerate.

the normal distribution is completely characterised by its mean and covariance. Therefore, we can work out the mean and the variance of Y in order to identify its distribution.

$$\begin{aligned}\mathbb{E}(Y) &= \mathbb{E}(AX+\eta)\\ &= A\mathbb{E}(X)+\eta\\ &= A\mu+\eta\\ \mathrm{Cov}(Y) &= \mathbb{E}([Y-\mathbb{E}(Y)][Y-\mathbb{E}(Y)]^T)\\ &= \mathbb{E}([AX+\eta-(A\mu+\eta)][AX+\eta-(A\mu+\eta)]^T)\\ &= \mathbb{E}(A(X-\mu)(X-\mu)^TA^T) = A\mathrm{Cov}(X)A^T\\ &= A\Sigma A^T\end{aligned}$$

Therefore, $Y \sim \mathcal{N}(A\mu+\eta, A\Sigma A^T)$.

Example 2.5. *The above derivation suggests a way to generate an n-dimensional multivariate sample $X \sim \mathcal{N}(\mu,\Sigma)$. We can first generate i.i.d. normal random variables $R_1,\dots,R_n \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ so that $R=(R_1,\dots,R_n)\sim\mathcal{N}(0_n,I_n)$ where 0_n is the $n\times 1$ vector of zeros and I_n is the identity matrix of size n. Then, we decompose $\Sigma = AA^T$ using the Cholesky decomposition. Finally, we let $X = AR+\mu$. Observe that the mean of X is $A0_n+\mu=\mu$ and covariance matrix of X is $AI_nA^T = AA^T = \Sigma$, so we are done.*

2.2.3 Composition

Let a random variable $Z \sim \Pi$ taking values from the set $\mathcal{Z}$ and Π has a pdf or pmf shown as $\pi(z)$. Suppose also that given z, $X|z \sim P_z$ where each P_z admits either a pmf or a pdf shown as $p_z(x)$. Then the marginal distribution P is a *mixture distribution* and in the presence of pdf's or pmf's, we have

$$p(x) = \begin{cases}\int p_z(x)\pi(z)dz, & \text{if } \pi(z) \text{ is a pdf}\\ \sum_z p_z(x)\pi(z), & \text{if } \pi(z) \text{ is a pmf}\end{cases} \tag{2.10}$$

Whether $p(x)$ is pmf or a pdf depends on whether $p_z(x)$ is pmf or pdf. The integral/sum may be hard to evaluate, and the mixture distribution may be hard to sample directly. But if we can easily sample from Π and from each P_z, then we can just

1. sample $Z \sim \Pi$,
2. sample $X \sim P_Z$, and
3. ignore Z and return X.

The random number we produce in this way will be an exact sample from P, i.e. $X \sim P$. This is the method of *composition*. Ignoring Z is also called *marginalisation*, by which we overcome the difficulty of dealing with the tough integral/sum in (2.10).

Example 2.6. *The density of a mixture of Gaussian distribution with K components with means and variance values $(\mu_1, \sigma_1^2), \dots, (\mu_K, \sigma_K^2)$, and probability weights $w_1, \dots, w_K$ for its components (such that $w_1 + \cdots + w_K = 1$) is given by*

$$p(x) = \sum_{k=1}^{K} w_k \phi(x; \mu_k, \sigma_k^2).$$

To sample from $p(x)$, we first sample the component number k with probability w_k (for example using the method of inversion), and given k, we sample $X \sim \mathcal{N}(\mu_k, \sigma_k^2)$

Example 2.7. *A sales company decides to reveal the demand D for a product over a month. However, for privacy reasons, it shares this average by adding some noise to D, which results in the shared value X. It is given that the distribution of the revealed demand X has the pdf*

$$p(x) = \sum_{d} \left[\frac{e^{-\lambda} \lambda^d}{d!} \right] \left[\frac{1}{2b} \exp\left(-\frac{|x-d|}{b} \right) \right]$$

We want to perform a Monte Carlo simulation for this data sharing process. How do we sample $X \sim P$?

Although $p(x)$ looks hard, observe that the first term in the sum is the pmf of $\mathcal{PO}(\lambda)$ evaluated at d (can be viewed as the demand) and the second term in the sum is the pdf of $\text{Laplace}(d, b)$ *evaluated at x (can be viewed as the noisy demand)*[6]*. Therefore, generation of X is possible by the method of composition as*

1. *Sample $D \sim \mathcal{PO}(\lambda)$,*

2. *Sample $X \sim \text{Laplace}(D, b)$ (equivalent to $V \sim \text{Laplace}(0, b)$ and $X = D + V$.).*

3. *Ignore D and return X.*

It is an exercise for you to figure out how one can sample from the Poisson and Laplace distributions.

2.2.4 Rejection sampling

Another common method of obtaining i.i.d. samples from P with density $p(x)$ is *rejection sampling*. This method was first mentioned in a 1947 letter by Von Neumann (Eckhardt, 1987), it was also presented a few years later in von Neumann (1951). The method is available when there exists an instrumental distribution Q with density $q(x)$ such that

- $q(x) > 0$ whenever $p(x) > 0$, and

- There exists $M > 0$ such that $p(x) \le M q(x)$ for all $x \in \mathcal{X}$.

Algorithm 2.1: Rejection sampling

```
1 Generate $X' \sim Q$ and $U \sim \text{Unif}(0,1)$.
2 If $U \leq \frac{p(X')}{Mq(X')}$, accept $X = X'$; else go to 1.
```

The rejection sampling method for obtaining one sample from P can be implemented with any $q(x)$ and $M > 0$ that satisfy the conditions above as in Algorithm 2.1.

How quickly do we obtain a sample with this method? Noting that the pdf of X' is $p_{X'}(x) = q(x)$, the acceptance probability can be derived as

$$\begin{aligned} \mathbb{P}(\text{Accept}) &= \int \mathbb{P}(\text{Accept}|X' = x) p_{X'}(x) dx \\ &= \int \frac{p(x)}{Mq(x)} q(x) dx \\ &= \frac{1}{M} \int p(x) dx \\ &= \frac{1}{M}, \end{aligned} \tag{2.11}$$

which is also the long term proportion of the number accepted samples over the number of trials. Therefore, taking $q(x)$ as close to $p(x)$ as possible to avoid large $p(x)/q(x)$ ratios and taking $M = \sup_x p(x)/q(x)$ are sensible choices to make the acceptance probability $\mathbb{P}(Accept)$ as high as possible.

The validity of the rejection sampling method can be verified by considering the distribution of the accepted samples. Using Bayes' theorem,

$$p_X(x) = p_{X'}(x|\text{Accept}) = \frac{p_{X'}(x)\mathbb{P}(\text{Accept}|X' = x)}{\mathbb{P}(\text{Accept})} = \frac{q(x)\frac{1}{M}\frac{p(x)}{q(x)}}{1/M} = p(x). \tag{2.12}$$

Example 2.8. *Suppose we want to sample $X \sim \Gamma(\alpha, 1)$ with $\alpha > 1$, where $\Gamma(\alpha, \beta)$ is the Gamma distribution with shape parameter α and scale parameter β. The density of $\Gamma(\alpha, 1)$ is*

$$p(x) = \frac{x^{\alpha-1}e^{-x}}{\Gamma(\alpha)}, \quad x > 0.$$

As possible instrumental distributions, consider the family of exponential distributions $Q_\lambda = \text{Exp}(\lambda)$, $0 < \lambda < 1$,[7] *with pdf*

$$q_\lambda(x) = \lambda e^{-\lambda x}, \quad x > 0.$$

[6] The pmf of $\mathcal{PO}(\lambda)$ evaluated at k is $\frac{e^{-\lambda}\lambda^k}{k!}$, and the pdf of Laplace$(\mu, b)$ evaluated at x is $\frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$

[7] For $\lambda > 1$, the ratio $\frac{p(x)}{q_\lambda(x)}$ is unbounded in x hence rejection sampling cannot be applied.

Recall that M has to satisfy $p(x) \leq Mq(x), \quad x \in \mathcal{X}$ and therefore given $q_\lambda(x)$, a sensible choice for M_λ is $M_\lambda = \sup_x p(x)/q_\lambda(x)$, hence we wish to use λ which minimises the required M_λ. Given $0 < \lambda < 1$, the ratio

$$\frac{p(x)}{q_\lambda(x)} = \frac{x^{\alpha-1}e^{(\lambda-1)x}}{\lambda\Gamma(\alpha)}$$

is maximised at $x = (\alpha-1)/(1-\lambda)$, so we have

$$M_\lambda = \frac{\left(\frac{\alpha-1}{1-\lambda}\right)^{\alpha-1} e^{-(\alpha-1)}}{\lambda\Gamma(\alpha)}$$

resulting in the acceptance probability

$$\frac{p(x)}{q_\lambda(x)M_\lambda} = \left(\frac{x(1-\lambda)}{\alpha-1}\right)^{\alpha-1} e^{(\lambda-1)x+\alpha-1}$$

Now, we have to minimise M_λ with respect to λ so that $\mathbb{P}(\text{Accept}) = 1/M_\lambda$ is maximised. M_λ is minimised at $\lambda^* = 1/\alpha$[8], yielding

$$M^* = \frac{\alpha^\alpha e^{-(\alpha-1)}}{\Gamma(\alpha)}.$$

Overall, the rejection sampling algorithm we choose to sample from $\Gamma(\alpha, 1)$ is

1. *Sample $X' \sim \text{Exp}(1/\alpha)$ and $U \sim \text{Unif}(0,1)$.*
2. *If $U \leq (x/\alpha)^{\alpha-1}e^{(1/\alpha-1)x+\alpha-1}$, accept $X = X'$, else go to 1.*

Check Figure 2.3 for the roles of optimum choice for λ and M. Also, Figure 2.4 illustrates the computational advantage of choosing λ optimally.

2.2.4.1 When $p(x)$ is known up to a normalising constant

One advantage of rejection sampling is that we can implement it even when we know $p(x)$ and $q(x)$ only up to some proportionality constants Z_p and Z_q, that is, when

$$p(x) = \frac{\widehat{p}(x)}{Z_p}, \quad Z_p = \int \widehat{p}(x)dx \tag{2.13}$$

$$q(x) = \frac{\widehat{q}(x)}{Z_q}, \quad Z_q = \int \widehat{q}(x)dx. \tag{2.14}$$

(Usually $q(x)$ is fully known in which case the following should be read with $\widehat{q}(x) = q(x)$ and $Z_q = 1$.) It is easy to check that one can perform the rejection sampling method as in Algorithm 2.2 for any M such that $\widehat{p}(x) \leq M\widehat{q}(x)$ for all $x \in \mathcal{X}$.

Justification of Algorithm 2.2 would follow from similar steps to those in (2.12). Also, in that case, the acceptance probability would be $\frac{1}{M}\frac{Z_p}{Z_q}$.

[8] That is why we constrain $\alpha > 1$; otherwise λ^* would be greater than 1, yielding an unbounded ratio.

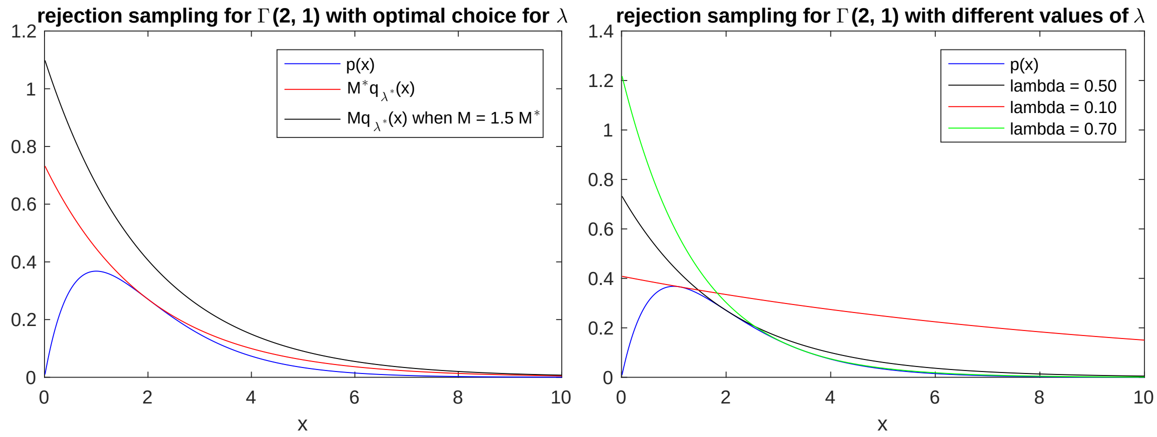


Figure 2.3: Rejection sampling for $\Gamma(2,1)$

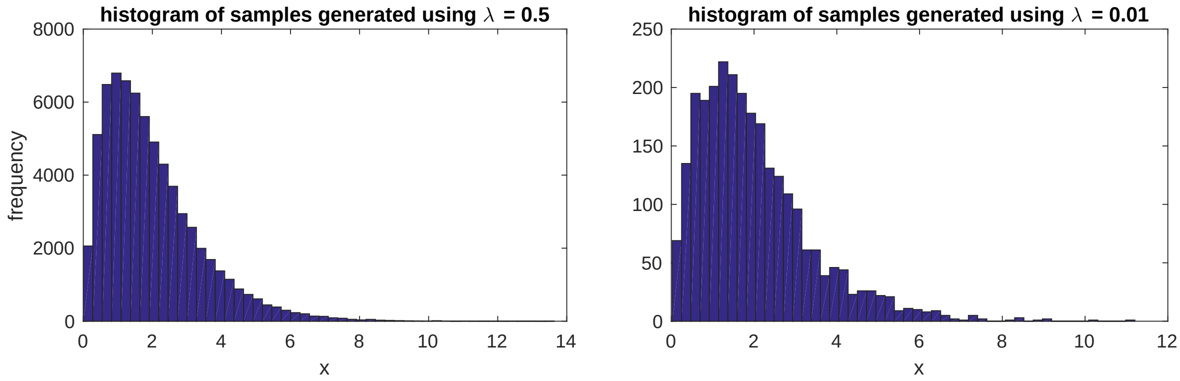


Figure 2.4: Rejection sampling for $\Gamma(2,1)$: Histograms with $\lambda = 0.5$ (68061 samples out of 10^5) and $\lambda = 0.01$ (2664 samples out of 10^5 trials).

Algorithm 2.2: Rejection sampling with unnormalised densities

```
1 Generate $X' \sim Q$ and $U \sim \mathrm{Unif}(0,1)$.
2 If $U \leq \frac{\widehat{p}(X')}{M\widehat{q}(X')}$, accept $X = X'$; else go to 1.
```

Example 2.9. *Sometimes we want to sample from truncated versions of well known distributions, i.e. where X is contained in an interval with density proportional to the density of the original distribution on that interval. For example, take the truncated standard normal distribution $\mathcal{N}_a(0,1)$ with density*

$$p(x) = \frac{\phi(x;0,1)\mathbb{I}(|x| \leq a)}{\int_{-a}^{a} \phi(y;0,1)dy} \tag{2.15}$$

$$= \frac{\widehat{p}(x)}{Z_p} \tag{2.16}$$

where $\widehat{p}(x) = \phi(x;0,1)\mathbb{I}(|x| \leq a)$ and $Z_p = \int_{-a}^{a} \phi(y;0,1)dy$. We can perform rejection sampling using $q(x) = \phi(x;0,1)$, (that is $\widehat{q}(x) = q(x)$ and $Z_q = 1$). Since $\widehat{p}(x)/\phi(x;0,1) = \mathbb{I}(|x| \leq a) \leq 1$, we can choose $M = 1$. Since $Z_q = 1$, the acceptance probability is $\frac{1}{M}\frac{Z_p}{Z_q} = \int_{-a}^{a} \phi(y;0,1)dy$.

The rejection sampling method for this distribution reduces to sampling from $Y \sim \phi$ and accepting $X = Y$ if $|Y| \leq a$, which is intuitive.

Example 2.10. *The unknown normalising constant issue mostly arises in Bayesian inference when we want to sample from the posterior distribution. The posterior density of X given $Y = y$ is proportional to*

$$p_{X|Y}(x|y) \propto p_X(x)p_{Y|X}(y|x) \tag{2.17}$$

where the normalising constant $p_Y(y) = \int p_X(x)p_{Y|X}(y|x)dx$ is usually intractable. Suppose we want to sample from $p_{X|Y}(x|y)$. When $p_{X|Y}(x|y)$ is not the density of a well known distribution, we may be able to use rejection sampling. If we can find $M > 0$ such that $p_{Y|X}(y|x) \leq M$ for all $x \in \mathcal{X}$, and the prior distribution P_X with density $p_X(x)$ is easy to sample from, then we can use rejection sampling with Q with $q(x) = p_X(x)$.

1. *Sample $X' \sim Q$ and $U \sim \mathrm{Unif}(0,1)$,*
2. *If $U \leq p_{Y|X}(y|X')/M$, accept $X = X'$; otherwise go to step 1.*

2.2.4.2 Squeezing

The drawback of rejection sampling is that in practice a rejection based procedure is usually not viable when $\mathcal{X}$ is high-dimensional, since $\mathbb{P}(\text{Accept})$ gets smaller and more computation is required to evaluate acceptance probabilities as the dimension increases. In the literature there exist approaches to improve the computational efficiency of rejection sampling. For example, assuming the densities exist, when it is difficult to compute $q(x)$, tests like $u \leq \frac{1}{M}\frac{p(x)}{q(x)}$ can be slow to evaluate. In this case, one may use a squeezing function $s : \mathcal{X} \to [0,\infty)$ such that $\frac{s(x)}{q(x)}$ is cheap to evaluate and $\frac{s(x)}{p(x)}$ is tightly bounded from above by 1. For such an s, not only $u \leq \frac{1}{M}\frac{s(x)}{q(x)}$ would guarantee $u \leq \frac{1}{M}\frac{p(x)}{q(x)}$, hence acceptance, but also if $u \leq \frac{1}{M}\frac{p(x)}{q(x)}$ then $u \leq \frac{1}{M}\frac{s(x)}{q(x)}$ would hold with a high probability. Therefore, in case of acceptance evaluation of $\frac{p(x)}{q(x)}$ would largely be avoided by checking $u \leq \frac{1}{M}\frac{s(x)}{q(x)}$ first. In Marsaglia (1977), the author proposed to squeeze $p(x)$ from above and below by $q(x)$ and $s(x)$ respectively, where $q(x)$ is easy to sample from and $s(x)$ is easy to evaluate. There are also adaptive methods to squeeze π from both below and above; they involve an adaptive scheme to gradually modify $q(x)$ and $s(x)$ from the samples that have already been obtained (Gilks, 1992; Gilks and Wild, 1992; Gilks et al., 1995).

Exercises

1. Use change of variables to show that X defined in (2.3) in Example 2.3 is distributed from $\text{Unif}(a, b)$.

2. Suggest a way to sample from $\mathcal{PO}(\lambda)$ using uniform random numbers.

3. Suggest a way to sample from $\text{Laplace}(a, b)$ using uniform random numbers. (Hint: Notice the similarity between the Laplace distribution and the exponential distribution.)

4. Show that the modified rejection sampling method described in Section 2.2.4.1 for unnormalised densities is valid, i.e. the accepted sample $X \sim P$, and it has the acceptance probability $\frac{Z_p}{Z_q M}$ as claimed. The derivation is similar to those in (2.11), (2.12).

5. Write your own function that takes a vector of non-negative numbers $w = [w_1 \dots w_K]$ of any size and outputs a matrix `X` (of the specified size, `size1` × `size2`) of i.i.d. integers in $\{1, \dots, K\}$, each with probability *proportional to* w_k *(i.e. their sum may not be normalised to* 1*)*. In MATLAB, your function should look something similar to `[X] = randsamp(w, size1, size2)`

6. Learn the *polar rejection* method, another method used to sample from $\mathcal{N}(0, 1)$. Write two different functions that produce i.i.d. standard normal random variables as many as it is specified (as an input to the function): one using the Box-Muller method and the other using the polar rejection method. Plot the histograms of 10^6 samples that you obtain from each function; make sure nothing strange happens in your code. Compare the speeds of your functions. Which method is faster? Why do you think is the reason?

7. Write your own function for generating a given number N of samples (specified as an input argument) from a multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ with given mean vector μ and covariance matrix Σ as inputs.

8. Derive the rejection sampling method for $\text{Beta}(a, b)$ $a, b \geq 1$ using the uniform distribution as the instrumental distribution. Write a function that implements this method. Is it still possible to use the uniform distribution as Q when $a < 1$ or $b < 1$? Why or why not?

Chapter 3

Monte Carlo Estimation

Summary: *This is a small chapter on the use of Monte Carlo to estimate certain quantities regarding a given distribution. Specifically, we will look at the importance sampling method for Monte Carlo integration.*

Let's go back to the beginning and consider the expectation in (1.3) once again

$$P(\varphi) = \mathbb{E}_P(\varphi(X)) = \int_{\mathcal{X}} \varphi(x)p(x)dx.$$

In order to estimate $P(\varphi)$ by the *plug-in estimator* (1.8)

$$P^N_{\text{MC}}(\varphi) = \frac{1}{N}\sum_{i=1}^{N} \varphi(X^{(i)}), \tag{3.1}$$

we need i.i.d. samples from P and in the previous chapter we covered some exact sampling methods for generating $X^{(i)} \sim P$, $i = 1, \ldots, N$.

However, there are many cases where $X \sim P$ is either impossible or too difficult, or wasteful. For example, rejection sampling uses only about $1/M$ of generated random samples to construct an approximation to P. In order to generate N samples, we need on average NM iterations of rejection sampling. The number M can be very large, especially in high dimensions, and rejection sampling may be wasteful.

3.1 Importance sampling

In contrast to rejection sampling, *importance sampling* uses every sample but weights each one according to the degree of similarity between the target and instrumental distributions. We describe the importance sampling method for continuous variables where P has a pdf $p(x)$ - the discrete version should be easy to figure out afterwards:

Suppose there exists a distribution Q with density $q(x)$ such that $q(x) > 0$ whenever $p(x) > 0$. Given $p(x)$ and $q(x)$, define the weight function $w : \mathcal{X} \to \mathbb{R}$

$$w(x) := \begin{cases} p(x)/q(x), & q(x) > 0, \\ 0 & q(x) = 0. \end{cases} \tag{3.2}$$

The idea of importance sampling follows from the *importance sampling fundamental identity* (Robert and Casella, 2004): We can rewrite $P(\varphi)$ as

$$\begin{aligned} P(\varphi) = \mathbb{E}_P(\varphi(X)) &= \int_{\mathcal{X}} \varphi(x)p(x)dx \\ &= \int_{\mathcal{X}} \varphi(x)\frac{p(x)}{q(x)}q(x)dx \\ &= \int_{\mathcal{X}} \varphi(x)w(x)q(x)dx \\ &= \mathbb{E}_Q(\varphi(X)w(X)) = Q(\varphi w) \end{aligned}$$

where φw stands for the product of the functions φ and w. This identity can be used with a Q which is easy to sample from, which leads to importance sampling given in Algorithm 3.1

Algorithm 3.1: Importance sampling

```
for $i = 1, \dots, N$ do
    Sample $X^{(i)} \sim Q$, and calculate $w(X^{(i)})$ according to (3.2).
Calculate the approximation of the expectation $P(\varphi)$ as
    $$P_{\text{IS}}^N(\varphi) := \frac{1}{N}\sum_{i=1}^{N} \varphi(X^{(i)})w(X^{(i)}). \qquad (3.3)$$
```

The weights $w(X^{(i)})$ are known as the *importance sampling weights*. Note that $P_{IS}^N(\varphi)$ is another plug-in estimator but for different distribution and function, namely it is the plug-in estimator for $Q\left(\varphi w\right)$. Therefore the estimator in (3.3) is unbiased and justified by the strong law of large numbers and the central limit theorem, provided that $Q(\varphi w) = \mathbb{E}_Q\left(\varphi(X)w(X)\right)$ and $\mathbb{V}_Q\left[w(X)\varphi(X)\right]$ are finite.

Example 3.1. *Suppose we have two variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ with joint pdf $p_{X,Y}(x, y)$. As we recall, we can write the joint pdf as*

$$p_{X,Y}(x, y) = p_X(x)p_{Y|X}(y|x)$$

In the Bayesian framework where X is the unknown parameter and Y is the observed variable (or data), $p_X(x)$ is called the prior density and it is usually easy to sample from, and $p_{Y|X}(y|x)$ is the conditional density of data, or the likelihood, *which is easy to compute.*[1]

[1] In fact, this is how one usually constructs the joint pdf in Bayesian framework: First define the prior $X \sim p_X(x)$, then define the data likelihood $Y|X = x \sim p_{Y|X}(y|x)$, so that the $p_{X,Y}(x, y)$ is constructed as above. When the starting point to define the joint density is to define the prior and the likelihood, it is notationally convenient to define the marginal and conditional pdfs $\mu(x) := p_X(x)$ and $g(y|x) := p_{Y|X}(y|x)$ and write $p(x, y) = \mu(x)g(y|x)$, $p(x|y) \propto \mu(x)g(y|x)$, $p(y) = \int \mu(x)g(y|x)dx$, etc.

In certain applications, we want to compute the evidence $p_Y(y)$ *at a given value* y *of the data. We can write* p_Y *as*

$$\begin{aligned} p_Y(y) &= \int_{\mathcal{X}} p_{X,Y}(x,y)dx && (3.4)\\ &= \int_{\mathcal{X}} p_X(x)p_{Y|X}(y|x)dx && (3.5)\\ &= \mathbb{E}_{P_X}(p_{Y|X}(y|X)) && (3.6)\end{aligned}$$

where the last line highlights the crucial observation that given y*, the likelihood can be thought as a function* $\varphi(x) = p_{Y|X}(y|x)$ *and* $p_Y(y)$ *can be written as an expectation of* $\varphi(X)$ *with respect to the prior with density* $p_X(x)$*. Therefore,* $p_Y(y)$ *can be estimated using a plug-in estimator where we sample* $X^{(1)}, \dots, X^{(N)} \sim p_X(x)$ *and estimate* $p_Y(y)$ *as*

$$p_Y(y) \approx \frac{1}{N}\sum_{i=1}^{N} p_{Y|X}(y|X^{(i)}), \quad X^{(1)}, \dots, X^{(N)} \sim p_X(x).$$

However, we do not need to sample from $p_X(x)$*. In fact, we can use importance sampling with an importance density* $q(x)$*.*

$$p_Y(y) \approx \frac{1}{N}\sum_{i=1}^{N} \frac{p_X(X^{(i)})}{q(X^{(i)})} p_{Y|X}(y|X^{(i)}), \quad X^{(1)}, \dots, X^{(N)} \sim q(x).$$

Being able to approximate a marginal distribution as in $p_Y(y)$ *will have an important role later on when we discuss sequential importance sampling methods.*

3.1.1 Variance reduction

As we have freedom to choose Q, we can control the variance of importance sampling (Robert and Casella, 2004).

$$\begin{aligned} \mathbb{V}_Q\left[P_{IS}^N(\varphi)\right] &= \frac{1}{N}\mathbb{V}_Q\left[w(X)\varphi(X)\right]\\ &= \frac{1}{N}\left(Q(w^2\varphi^2) - Q(w\varphi)^2\right)\\ &= \frac{1}{N}\left(Q(w^2\varphi^2) - P(\varphi)^2\right).\end{aligned}$$

Therefore, minimising $\mathbb{V}_Q\left[P_{IS}^N(\varphi)\right]$ is equivalent to minimising $Q(w^2\varphi^2)$, which can be lower bounded as

$$Q(w^2\varphi^2) \geq Q(w|\varphi|)^2 = P(|\varphi|)^2$$

using the Jensen's inequality. Considering $Q(w^2\varphi^2) = P(w\varphi^2)$, this bound is attainable if we choose q such that it satisfies

$$w(x) = \frac{p(x)}{q(x)} = \frac{P(|\varphi|)}{|\varphi(x)|}, \quad x \in \mathcal{X}, \varphi(x) \neq 0.$$

This results in the optimum choice of q to be

$$q(x) = p(x)\frac{|\varphi(x)|}{P(|\varphi|)}$$

for points $x \in \mathcal{X}$ such that $\varphi(x) \neq 0$, and the resulting minimum variance is given by

$$\min_Q \mathbb{V}_Q\left[P_{IS}^N(\varphi)\right] = \frac{1}{N}\left([P(|\varphi|)]^2 - [P(\varphi)]^2\right).$$

Note that this minimum value is 0 if $\varphi(x) \geq 0$ for all $x \in \mathcal{X}$. Therefore, importance sampling in principle can achieve a lower variance than perfect Monte Carlo. Of course, if we can not compute $P(\varphi)$ already, it is unlikely that we can compute $P(|\varphi|)$. Also, it will be rare that we can easily simulate from the optimal Q even if we can construct it. Instead, we are guided to seek a Q close to the optimal one, but from which it is easy to sample.

Example 3.2. *We wish to implement importance sampling in order to approximate* $\mathbb{E}(\varphi(X))$ *where* $X \sim P = \mathcal{N}(\mu, \sigma^2)$*. Instead of sampling from* P *directly, we want to sample from* $Q_k = \mathcal{N}(\mu, \sigma^2/k)$*. We want to choose the best* k *for* φ *in terms of the variance of the importance sampling estimate. Recall that minimising* $\mathbb{V}_{Q_k}\left[P_{\mathrm{IS}}^N(\varphi)\right]$ *is equivalent to minimising* $Q_k(w_k^2\varphi^2)$ *where* $w_k(x) = p(x)/q_k(x)$.

$$\begin{aligned} Q_k(w_k^2\varphi^2) &= \int_{\mathcal{X}} q_k(x)\frac{p(x)^2}{q_k(x)^2}\varphi(x)^2 dx \\ &= \int_{\mathcal{X}} \frac{p(x)^2}{q_k(x)}\varphi(x)^2 dx \end{aligned}$$

The ratio $\frac{p(x)^2}{q_k(x)}$ *is*

$$\begin{aligned} \frac{p(x)^2}{q_k(x)} &= \frac{1}{2\pi\sigma^2}e^{-(x-\mu)^2/\sigma^2}\frac{\sqrt{2\pi\sigma^2}}{\sqrt{k}}e^{\frac{k}{2}\frac{(x-\mu)^2}{\sigma^2}} \\ &= \frac{1}{\sqrt{k}\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2}(2-k)\frac{(x-\mu)^2}{\sigma^2}} \end{aligned}$$

This ratio diverges when $k > 2$*, and unless* $\varphi(x)^2$ *balances it the second moment* $Q_k(w_k^2\varphi^2)$ *diverges. Therefore, let us confine* k *to* $k \in (0, 2)$*. In that case, we can rewrite*

$$\begin{aligned} \frac{p(x)^2}{q_k(x)} &= \frac{1}{\sqrt{k(2-k)}}\frac{\sqrt{2-k}}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2}(2-k)\frac{(x-\mu)^2}{\sigma^2}} \\ &= \frac{1}{\sqrt{k(2-k)}}q_{2-k}(x) \end{aligned}$$

Therefore,

$$Q_k(w_k^2\varphi^2) = \frac{1}{\sqrt{k(2-k)}}Q_{2-k}(\varphi^2) = \frac{1}{\sqrt{k(2-k)}}\mathbb{E}_{Q_{2-k}}(\varphi(X)^2)$$

When $\varphi(x) = x$ and $\mu = 0$, $Q_{2-k}(\varphi^2) = \mathbb{E}_{Q_{2-k}}(X^2) = \frac{\sigma^2}{2-k}$. Therefore, we need to minimise

$$\frac{1}{\sqrt{k(2-k)}}\frac{\sigma^2}{2-k} = \sigma^2(2-k)^{-3/2}k^{-1/2}.$$

The minimum is attained at $k = 1/2$ and is

$$\mathbb{V}_{Q_{1/2}}(P^N_{\mathrm{IS}}(\varphi)) = \frac{1}{N}\left\{\sigma^2(2-k)^{-3/2}k^{-1/2}\right\}\Big|_{k=1/2} = 0.7698\frac{\sigma^2}{N}$$

The variance of the plug-in estimator $P^N_{\mathrm{MC}}(\varphi)$ for $P(\varphi)$ is $\frac{\sigma^2}{N}$, which is larger!

Effective sample size: One approximation of the importance sampling estimator is proposed in Kong et al. (1994) to be

$$\begin{aligned}\mathbb{V}_Q\left[P^N_{\mathrm{IS}}(\varphi)\right] &\approx \frac{1}{N}\mathbb{V}_P\left[\varphi(X)\right]\{1 + \mathbb{V}_Q\left[w(X)\right]\} \\ &= \mathbb{V}_P\left[P^N_{\mathrm{MC}}(\varphi)\right]\{1 + \mathbb{V}_Q\left[w(X)\right]\}.\end{aligned}$$

This approximation might be confusing at the first instance since it suggests that the variance of importance sampling is always greater than that of perfect Monte Carlo, which we have just seen is not the case. However, it is useful as it provides an easy way of monitoring the efficiency of the importance sampling method. Consider the ratio of variances of the importance sampling method with N particles and perfect Monte Carlo with N' particles, which is given according to this approximation by

$$\frac{\mathbb{V}_Q\left[P^N_{\mathrm{IS}}(\varphi)\right]}{\mathbb{V}_P\left[P^{N'}_{\mathrm{MC}}(\varphi)\right]} \approx \frac{N'}{N}\{1 + \mathbb{V}_Q\left[w(X)\right]\}.$$

The number N' for which this ratio is 1 would suggest how many samples for perfect Monte Carlo would be equivalent to N samples for importance sampling. For this reason this number is defined as the *effective sample size* (Kong et al., 1994; Liu, 1996) and it is given by

$$N_{\mathit{eff}} = \frac{N}{1 + \mathbb{V}_Q\left[w(X)\right]}.$$

Obviously, the term $\mathbb{V}_Q\left[w(X)\right]$ itself is usually estimated using the samples $X^{(1)}, \dots, X^{(N)}$ with weights $w(X^{(i)}), \dots, w(X^{(N)})$ obtained from the importance sampling method.

3.1.2 Self-normalised importance sampling

Like rejection sampling, the importance sampling method can be modified for the cases when $p(x) = \frac{\widehat{p}(x)}{Z_p}$ and/or $q(x) = \frac{\widehat{q}(x)}{Z_q}$ and we only have $\widehat{p}(x)$ and $\widehat{q}(x)$. This time, letting

$$w(x) := \begin{cases} \frac{\widehat{p}(x)}{\widehat{q}(x)}, & \widehat{q}(x) > 0 \\ 0, & \widehat{q}(x) = 0, \end{cases}$$

observe that

$$\begin{aligned} Q(w) = \mathbb{E}_Q(w(X)) &= \int \frac{\widehat{p}(x)}{\widehat{q}(x)} q(x)dx \\ &= \int \frac{p(x)Z_p}{q(x)Z_q} q(x)dx \\ &= Z_p/Z_q. \end{aligned}$$

and

$$\begin{aligned} Q\left(w\varphi\right) = \mathbb{E}_Q(w(X)\varphi(X)) &= \int \frac{\widehat{p}(x)}{\widehat{q}(x)} \varphi(x)q(x)dx \\ &= \int \frac{p(x)Z_p}{q(x)Z_q} \varphi(x)q(x)dx \\ &= P(\varphi)Z_p/Z_q. \end{aligned}$$

Therefore, we can write the importance sampling fundamental identity in terms of $\widehat{p}$ and $\widehat{q}$ as

$$P(\varphi) = \frac{Q\left(\varphi w\right)}{Z_p/Z_q} = \frac{Q\left(w\varphi\right)}{Q\left(w\right)}.$$

The importance sampling method can be modified to approximate both the nominator, the unnormalised estimate, and the denominator, the normalisation constant, by using Monte Carlo. Sampling $X^{(1)}, \dots, X^{(N)}$ from Q, we have the approximation

$$P_{\text{IS}}^N(\varphi) = \frac{\frac{1}{N}\sum_{i=1}^N \varphi(X^{(i)})w(X^{(i)})}{\frac{1}{N}\sum_{i=1}^N w(X^{(i)})} = \sum_{i=1}^N W^{(i)}\varphi(X^{(i)}). \tag{3.7}$$

where

$$W^{(i)} = \frac{w(X^{(i)})}{\sum_{j=1}^N w(X^{(j)})}$$

are called the *normalised importance weights* as they sum up to 1. The resulting method, which is called *self-normalised importance sampling* is given in Algorithm 3.2: Being the ratio of two unbiased estimators, estimator of the self-normalised importance sampling is biased for finite N. However, its consistency and stability are provided by a strong law of large numbers and a central limit theorem in Geweke (1989). In the same work, the variance of the self normalised importance sampling estimator is analysed and an approximation is provided, from which it reveals that it can provide lower variance estimates than the unnormalised importance sampling method. Also normalised importance sampling has the nice property of estimating a constant by itself, unlike the unnormalised importance sampling method. Therefore, this method can be preferable to its unnormalised version even if it is not the case that P and Q are known only up to proportionality constants.

Self-normalised importance sampling is also called Bayesian importance sampling in Geweke (1989), since in most Bayesian inference problems normalising constant of posterior distribution is unknown.

Algorithm 3.2: Self-normalised importance sampling

```
for $i = 1, \dots, N$ do
    Generate $X^{(i)} \sim Q$, calculate $w(X^{(i)}) = \frac{\widehat{p}(X^{(i)})}{\widehat{q}(X^{(i)})}$.
for $i = 1, \dots, N$ do
    Set $W^{(i)} = \frac{w(X^{(i)})}{\sum_{j=1}^{N} w(X^{(j)})}$.
Calculate the approximation to the expectation
    $$P_{\mathrm{IS}}^{N}(\varphi) = \sum_{i=1}^{N} W^{(i)} \varphi(X^{(i)})$$
```

Example 3.3. *Let us consider the posterior distribution in Example 2.10*

$$p_{X|Y}(x|y) \propto p_X(x) p_{Y|X}(y|x)$$

and the unknown normalising constant is $p_Y(y) = \int p_X(x) p_{Y|X}(y|x) dx$. Given the data $Y = y$, we want to calculate the expectation of $\varphi : \mathcal{X} \to \mathbb{R}$ with respect to $p_{X|Y}(x|y)$

$$P_X(\varphi|Y = y) = \mathbb{E}(\varphi(X)|Y = y) = \int p_{X|Y}(x|y)\varphi(x)dx.$$

Since we know $p_{X|Y}(x|y)$ only up to a proportionality constant, we use self-normalised importance sampling. With the choice of Q with density $q(x)$, self-normalised importance sampling becomes

1. *For $i = 1, \dots, N$; generate $X^{(i)} \sim Q$, calculate*

$$w(X^{(i)}) = \frac{p_X(X^{(i)}) p_{Y|X}(y|X^{(i)})}{q(X^{(i)})}.$$

2. *For $i = 1, \dots, N$; set $W^{(i)} = \frac{w(X^{(i)})}{\sum_{j=1}^{N} w(X^{(j)})}$.*

3. *Approximate $\mathbb{E}(\varphi(X)|Y = y) \approx \sum_{i=1}^{N} W^{(i)} \varphi(X^{(i)})$.*

If we choose $q(x) = p_X(x)$, i.e. the prior density, then $w(x) = p_{Y|X}(y|x)$ reduces to the likelihood. But this is not always a good idea as we will see in the next example.

Example 3.4. *Suppose we have an unknown mean parameter $X \in \mathbb{R}$ whose prior distribution is represented by $X \sim \mathcal{N}(\mu, \sigma^2)$. Conditional on $X = x$, n data samples $Y = (Y_1, \dots, Y_n) \in \mathbb{R}^n$ are generated independently*

$$Y_1, \dots, Y_n | X = x \overset{\text{i.i.d.}}{\sim} \text{Unif}(x - a, x + a).$$

We want to estimate the posterior mean of X *given* $Y = y = (y_1, \dots, y_n)$*, i.e.* $\mathbb{E}(X|Y = y) = \int p_{X|Y}(x|y)x dx$*, where*

$$p_{X|Y}(x|y) \propto p_X(x)p_{Y|X}(y|x)$$

The prior density and likelihood are $p_X(x) = \phi(x; \mu, \sigma^2)$ *and* $p_{Y|X}(y|x) = \prod_{i=1}^n \frac{1}{2a}\mathbb{I}_{(x-a,x+a)}(y_i)$*, so the posterior distribution can be written as*

$$p_{X|Y}(x|y) \propto \phi(x; \mu, \sigma^2)\frac{1}{(2a)^n}\prod_{i=1}^n \mathbb{I}_{(x-a,x+a)}(y_i)$$

Densities $p_X(x)$ *and* $p_{X,Y}(x, y)$ *versus* x *for a fixed* $Y = y = (y_1, \dots, y_n)$ *with* $n = 10$ *generated from the marginal distribution of* Y *with* $a = 2$*,* $\mu = 0$*, and* $\sigma^2 = 10$ *are given in Figure 3.1. Note that the second plot is proportional to the posterior density.*

We can use self-normalised importance sampling to estimate $\mathbb{E}(X|Y = y)$*. The choice of the importance density is critical here: Suppose we chose* Q *to be the prior distribution for* X*, i.e.* $q(x) = \phi(x; \mu, \sigma^2)$*. This is a valid choice, however if* a *is small and* σ^2 *is relatively large, it is likely that the resulting weight function*

$$w(x) = \frac{1}{(2a)^n}\prod_{i=1}^n \mathbb{I}_{(x-a,x+a)}(y_i).$$

will end up being zero for most of the generated samples from Q *and it will be* $\frac{1}{(2a)^n}$ *for few samples. This results in a high variance in the importance sampling estimator. What is worse, it is possible to have all weights to be zeros and hence the denominator in (3.7) can be zero. Therefore the estimator is a poor one.*

Let $y_{\max} = \max_i y_i$ *and* $y_{\min} = \min_i y_i$*. A careful inspection of* $p_{X|Y}(x|y)$ *reveals that given* $y = (y_1, \dots, y_n)$*,* X *must be contained in* $(y_{\max} - a, y_{\min} + a)$*. In other words,*

$$x \in (y_{\max} - a, y_{\min} + a) \Leftrightarrow x - a < y_i < x + a, \quad \forall i = 1, \dots, n$$

Therefore, a better importance density does not waste its time outside the interval $(y_{\max} - a, y_{\min} + a)$ *and generate samples in that interval. As an example, we can choose* $Q = \text{Unif}(y_{\max} - a, y_{\min} + a)$*. With that choice, the weight function will be*

$$w(x) = \begin{cases} \frac{\phi(x;\mu,\sigma^2)\frac{1}{(2a)^n}}{1/(2a+y_{\min}-y_{\max})}, & x \in (y_{\max} - a, y_{\min} + a) \\ 0, & \text{else} \end{cases}$$

Note that since we are using the self-normalised importance sampling estimator and hence we normalise the weights $W^{(i)} = w(X^{(i)})/\sum_{j=1}^N w(X^{(j)})$*, we do not need to calculate the constant factor* $(2a + y_{\min} - y_{\max})/(2a)^n$ *for the weights.*

Figure 3.2 compares the importance sampling estimators with the two different importance distributions mentioned above. The histograms are generated from 10000 *Monte Carlo runs (*10000 *independent estimates of the posterior mean) for each estimator. Observe that the estimates obtained when the importance distribution is the prior is more wide-spread, exhibiting a higher variance.*

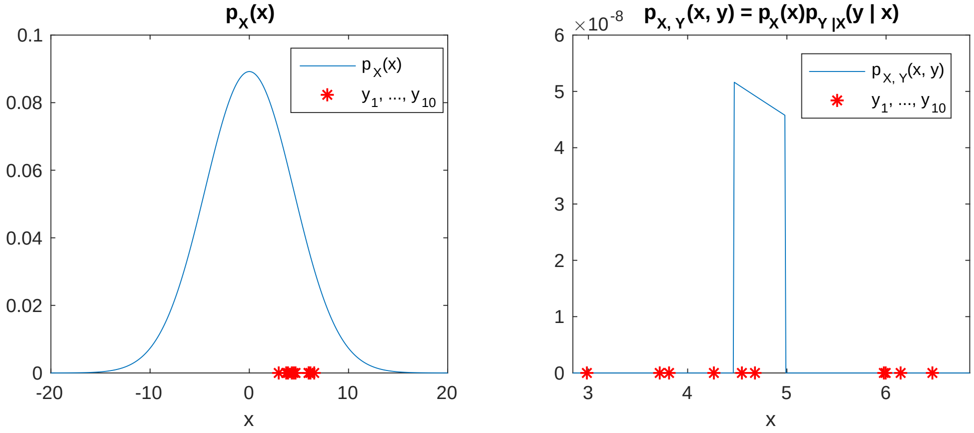


Figure 3.1: $p_X(x)$ and $p_{X,Y}(x,y)$ vs x for the problem in Example 3.4 with $n = 10$ and $a = 2$

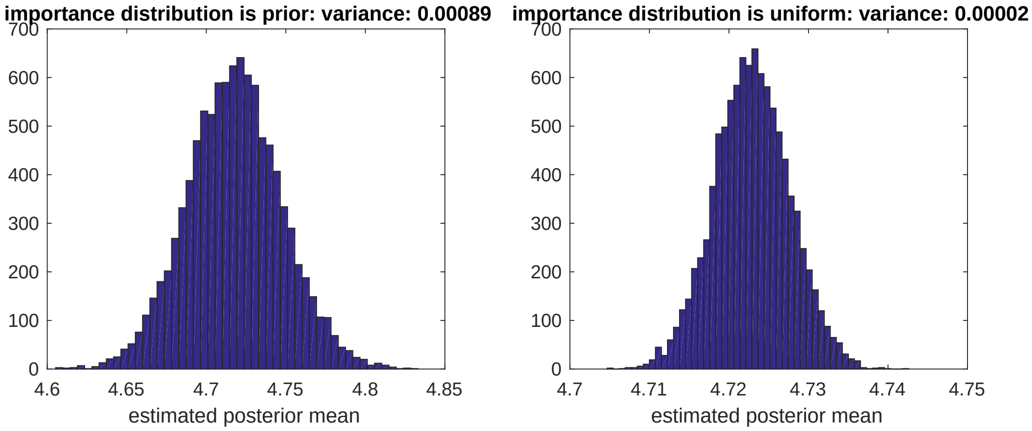


Figure 3.2: Histograms for the estimate of the posterior mean using two different importance sampling methods as described in Example 3.4 with $n = 10$ and $a = 2$.

Exercises

1. Consider Example 3.2, where importance sampling for $\mathcal{N}(\mu, \sigma^2)$ is discussed.

    - This time, take $\mu = 0$ and $\varphi(x) = x^2$. Find the optimum k for this ϕ and calculate the gain due to variance reduction compared to the plug-in estimator $P^N_{\text{MC}}(\varphi)$.
    - Implement importance sampling (e.g., in MATLAB) for both $\varphi(x) = x$ and $\varphi(x) = x^2$, and verify that in each case the variance of the IS estimator is lower than that of the plug-in estimator $P^N_{\text{MC}}(\varphi)$. Verify also that the $k = 1/2$ estimator is inferior for calculating the second moment, and likewise the $k = 1/3$ estimator is inferior for the first moment.

2. This example is based on Project Evaluation and Review Technique (PERT), a project planning tool.[2] Consider the software project described in Table 3.1 with 10 tasks (activities), indexed by $j = 1, \ldots, 10$. The project is completed when all of the tasks are completed. A task can begin only after all of its predecessors have been completed. The project starts at time 0. Task j starts at time S_j, takes time T_j and ends at time $E_j = S_j + T_j$. Any task j with no predecessors (here only task 1) starts at $S_j = 0$. The start time for a task with predecessors is the maximum of the ending times of its predecessors. For example, $S_4 = E_2$ and $S_9 = \max(E_5, E_6, E_7)$. The project as a whole ends at time E_{10}.

j	Task	Predecessors	mean duration θ_j
1	Planning	None	4
2	Database Design	1	4
3	Module Layout	1	2
4	Database Capture	2	5
5	Database Interface	2	2
6	Input Module	3	3
7	Output Module	3	2
8	GUI Structure	3	3
9	I/O Interface Implementation	5, 6, 7	2
10	Final Testing	4, 8, 9	2

Table 3.1: PERT: Project tasks, predecessor-successor relations, and mean durations

- Using predecessor-successor relations in Table 3.1, draw a diagram (for example, an acyclic directed graph) that shows the predecessor-successor relations in this example, with a node for each activity.

[2] The example is largely taken from `http://statweb.stanford.edu/~owen/mc/Ch-var-is.pdf` The original source can be reached at `http://optlab-server.sce.carleton.ca/POAnimations2007/PERT.html`.

- Write a MATLAB function that takes duration times $T_j, j = 1, \ldots, 10$ and outputs the completion time for the project.
- Assume $T_j \sim \text{Exp}(1/\theta_j)$ independent exponentially distributed random variables with means θ_j given in the final column of the table. Simulate this project (i.e. task durations and completion time) and generate $N = 10000$ independent realisations of the completion time. Plot the histogram of completion times and estimate the mean completion time.
- The completion time E_{10} can be seen as a function of task times $X = (T_1, \ldots, T_{10})$, and the function is what you just coded above. Now suppose that there will be a severe penalty should the project miss a deadline in 70 days time. Derive the Monte Carlo estimator for $\mathbb{P}(E_{10} > 70) = \mathbb{E}(\mathbb{I}(E_{10} > 70))$ and implement it $M = 1000$ times with $N = 10000$ samples. Out of the $M = 1000$ samples, calculate the sample variance of $P^N_{\text{MC}}(\varphi)$ with $\varphi(X) = \mathbb{I}(E_{10} > 70)$
- This time, estimate the same probability using importance sampling, taking Q the distribution of independent task times that are exponentially distributed with means λ_j (instead of θ_j), that is

$$X^{(i)} = (T_1^{(i)}, \ldots, T_{10}^{(i)}) \sim \text{Exp}(1/\lambda_1) \times \ldots \times \text{Exp}(1/\lambda_{10})$$

  Write down the expression for $P^N_{\text{IS}}(\varphi)$ in terms of λ_j's and θ_j's and $T_j^{(i)}$'s. Try $\lambda_j = \kappa\theta_j$ for various values of κ to see if you can come up with a better estimator, that is one with lower variance, than the plug-in estimator $P^N_{\text{MC}}(\varphi)$.

3. This is a simple example that illustrates the source localisation problem. We have a source (or target) on the 2-D plane whose unknown location

$$X = (X(1), X(2)) \in \mathbb{R}^2$$

we wish to find. We collect distance measurements for the source using three sensors, located at positions s_1, s_2, and s_3, see Figure 3.3. The measured distances $Y = (Y_1, Y_2, Y_3)$, however, are noisy with independent normally distributed noises with equal variance:[3]

$$Y_i|X = x \sim \mathcal{N}(||x - s_i||, \sigma_y^2), \quad i = 1, 2, 3,$$

where $||\cdot||$ denotes the Euclidean distance. Letting $r_i = ||x - s_i||$, the likelihood evaluated at $y = (y_1, y_2, y_3)$ given x can be written as

$$p_{Y|X}(y|x) = \prod_{i=1}^{3} \phi(y_i; r_i, \sigma_y^2) \tag{3.8}$$

[3] In this way we allow negative distances, which makes the normal distribution not the most proper choice. However, for the sake of ease with computations, we overlook that in this example.

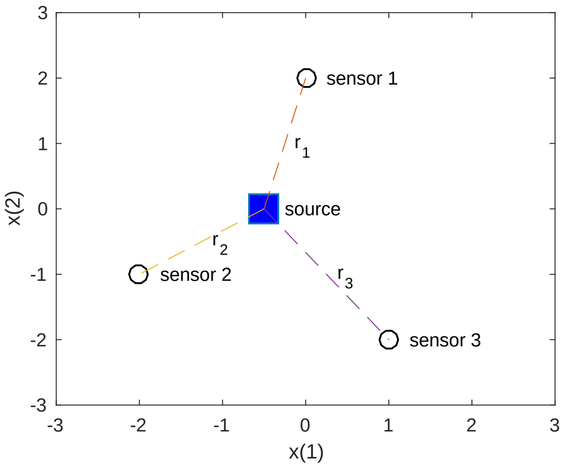


Figure 3.3: Source localisation problem with three sensors and one source

We do not know much *a priori* information about X, therefore we take the prior distribution X as the bivariate normal distribution with zero mean vector and a diagonal covariance matrix, $X \sim \mathcal{N}(0_2, \sigma_x^2 I_2)$, so that the density is

$$p_X(x) = \phi(x(1); 0, \sigma_x^2)\phi(x(2); 0, \sigma_x^2). \tag{3.9}$$

See Figure 3.4 for an illustration of prior, likelihood, and posterior densities for this problem.

Given noisy measurements, $Y = y = (y_1, y_2, y_3)$, we want to locate X, so we are interested in the posterior mean vector

$$\mathbb{E}(X|Y = y) = [\mathbb{E}(X(1)|Y = y), \mathbb{E}(X(2)|Y = y)].$$

Write a function that takes y, positions of the sensors s_1, s_2, s_3, the prior and likelihood variances σ_x^2 and σ_y^2, and the number of samples N as inputs, implements self-normalised importance sampling (why this version?) in order to approximate $\mathbb{E}(X|Y = y)$ and outputs its estimate. Try your code with $s_1 = (0, 2)$, $s_2 = (-2, -1)$, $s_3 = (1, -2)$, $y_1 = 2$, $y_2 = 1.6$, $y_3 = 2.5$, $\sigma_x^2 = 100$, and $\sigma_y^2 = 1$ which are the values used to generate the plots in Figure 3.4.

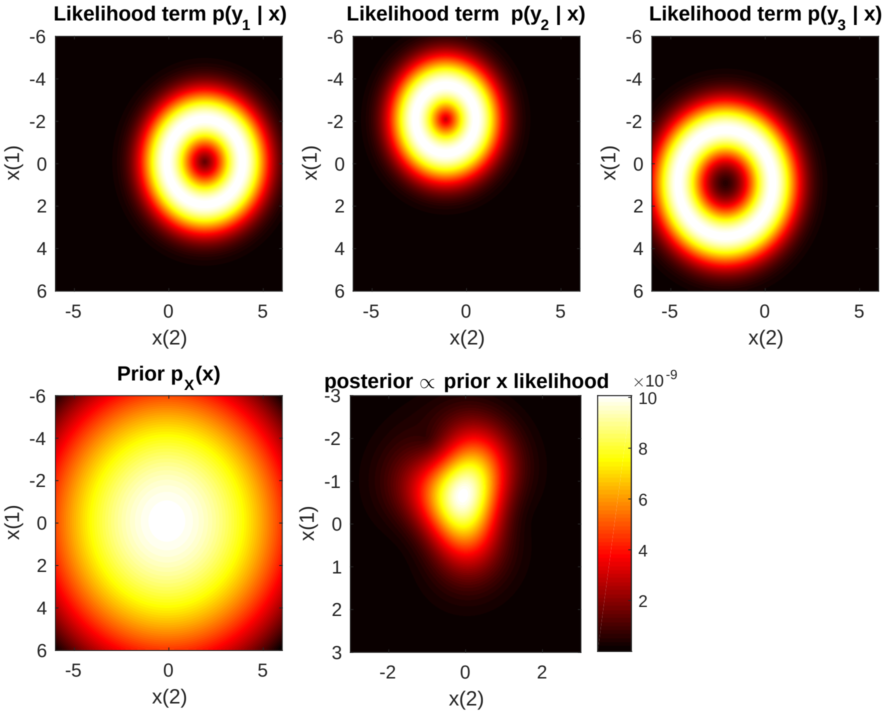


Figure 3.4: Source localisation problem with three sensors and one source: The likelihood terms, prior, and the posterior. The parameters and the variables are $s_1 = (0, 2)$, $s_2 = (-2, -1)$, $s_3 = (1, -2)$, $y_1 = 2$, $y_2 = 1.6$, $y_3 = 2.5$, $\sigma_x^2 = 100$, and $\sigma_y^2 = 1$

Chapter 4

Bayesian Inference

Summary: *In this chapter, we provide a brief introduction to Bayesian statistics. Some quantities of interest that are calculated from the posterior distribution will be explained. We will see some examples where one can find the exact form of the posterior distribution. In particular, we will discuss conjugate priors that are useful for deriving tractable posterior distributions. This chapter also introduces a relaxation in the notation to be adopted in the later chapters.*

4.1 Conditional probabilities

Recall Bayes' rule from Appendix A.3. Consider the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Given two sets $A, B \in \mathcal{F}$, the conditional distribution of A given B is

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{\mathbb{P}(A)\mathbb{P}(B|A)}{\mathbb{P}(B)} \tag{4.1}$$

Here we see some examples where Bayes' rule is in action to calculate posterior probabilities.

Example 4.1 (Conditional probabilities of sets). *A pair of fair (unbiased) dice are rolled independently. Let the outcomes be X_1 and X_2.*

- *It is observed that the sum $S = X_1 + X_2 = 8$. What is the probability that the outcome of at least one of the dice is 3?*

  We apply the Bayes rule: Define the sets $A = \{(X_1, X_2) : X_1 = 3 \text{ or } X_2 = 3\}$. $B = \{(X_1, X_2) : S = 8\}$, so that the desired probability is $\mathbb{P}(A|B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$.

  $$B = \{(2,6), (3,5), (4,4), (5,3), (6,2)\}, \quad A \cap B = \{(3,5), (5,3)\}.$$

  Since the dice are fair, every outcome is equiprobable, having probability $1/36$. Therefore,

  $$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{2/36}{5/36} = \frac{2}{5}.$$

- *It is observed that the sum is even. What is the probability that the sum is smaller than or equal to 4? Similarly, we define the sets $A = \{(X_1, X_2) : X_1 + X_2 \leq 4\}$.*

$B = \{(X_1, X_2) : X_1 + X_2 \text{ is even}\}$. *Explicity, we have*

$$B = \{(X_1, X_2) : X_1, X_2 \text{ are both even}\} \cup \{(X_1, X_2) : X_1, X_2 \text{ are both odd}\}.$$
$$A \cap B = \{(1, 1), (1, 3), (3, 1), (2, 2)\}.$$

Therefore,

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{4/36}{3/6 \times 3/6 + 3/6 \times 3/6} = \frac{2}{9}.$$

Example 4.2 (Model selection). *There are two coins in an urn, one fair and one biased with probability of tail $\rho = 0.3$. Someone picks up one of the coins at random (with half probability for picking up either coin) and tosses it n times and reports the outcomes: $\mathcal{D} = (H, T, H, H, T, H, H, H, T, H)$. Conditional on $\mathcal{D}$, what is the probability that the fair dice was picked up?*

We have two hypotheses (models): H_1: The coin picked up was the fair one, H_2: The coin picked was the biased one. The prior probabilities for these models are the same: $\mathbb{P}(H_1) = \mathbb{P}(H_2) = 0.5$. The likelihood of data, that is the conditional probability of the outcomes is:

$$\mathbb{P}(\mathcal{D}|H_i) = \begin{cases} 1/2^{10}, & i = 1, \\ \rho^{n_T}(1-\rho)^{n_H}, & i = 2, \end{cases}$$

where n_T and n_H are the number of times the coin showed tail and head, respectively. From Bayes' rule, we have

$$\begin{aligned}\mathbb{P}(H_1|\mathcal{D}) = \frac{\mathbb{P}(\mathcal{D}, H_1)}{\mathbb{P}(\mathcal{D})} &= \frac{\mathbb{P}(H_1)\mathbb{P}(\mathcal{D}|H_1)}{\mathbb{P}(\mathcal{D}|H_1)\mathbb{P}(H_1) + \mathbb{P}(\mathcal{D}|H_2)\mathbb{P}(H_2)} \\ &= \frac{1/2 \times 1/2^{10}}{1/2 \times 1/2^{10} + 1/2 \times \rho^{n_T}(1-\rho)^{n_H}} \\ &= \frac{1/2^{10}}{1/2^{10} + \rho^{n_T}(1-\rho)^{n_H}}\end{aligned}$$

and, of course, $\mathbb{P}(H_2|\mathcal{D}) = 1 - \mathbb{P}(H_1|\mathcal{D})$. Substituting $\rho = 0.3$ and $n_T = 3$, we have $\mathbb{P}(H_1|\mathcal{D}) = 0.3052$ and $\mathbb{P}(H_2|\mathcal{D}) = 0.6948$.

4.2 Deriving Posterior distributions

In this section, we study posterior distributions and discuss their use. We introduce the notion of conjugacy, a very important tool for deriving exact posterior distributions for some likelihood models. Then, we will look at some useful inferential quantities that are calculated from the posterior distribution.

When random variables $X \in \mathcal{X}, Y \in \mathcal{Y}$ with joint pdf/pmf $p_{X,Y}(x, y)$ are considered, the conditional pdf/pmf $p_{X|Y}(x|y)$ is

$$p_{X|Y}(x|y) = \frac{p_{X,Y}(x, y)}{p_Y(y)} = \frac{p_X(x)p_{Y|X}(y|x)}{p_Y(y)} \tag{4.2}$$

4.2.1 A note on future notational simplifications

It may be tedious to keep the subscripts in pdf's or pmf's such as $p_{X,Y}$, $p_{X|Y}$, etc. Formally, this is necessary to indicate what random variables are considered and what probability distribution exactly we mean. However, it is common practice to drop the cumbersome subscripts and use $p(x,y)$, $p(x)$, $p(x|y)$, etc. whenever it is clear from the context what distribution we mean. We will also adopt this simplification in this document. For example, we will frequently write Bayes' rule as

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)}$$

It is also common to use densities as well as distributions to indicate the distribution of a random variable. For example, all the expressions below mean the same thing: X is distributed from the distribution P, whose pdf or pmf is $p(x)$

$$X \sim P, \quad X \sim p(\cdot), \quad X \sim p(x), \quad x \sim P, \quad x \sim p(\cdot), \quad x \sim p(x).$$

In the rest of this document, we will use the aforementioned notations interchangeably, choosing the most suitable one depending on the context.

When the statistical model is prone to misunderstandings in case p is used for everything, perhaps a nicer approach than using p generically from the beginning is to start with different letters such as f, g, h, μ, etc. for different pdf's or pmf's when constructing the joint distribution for the random variables of interest.

Example 4.3. *Consider random variables X, Y, Z, U and assume that Y and Z are conditionally independent given X, and U is independent from X given Y and Z; see Figure 4.1. In such a case, it may be convenient to construct the joint density by first declaring the density for X, $\mu(x)$. Next, define the conditional densities $f(y|x)$ and $g(z|x)$ for Y given X and Z given X. Finally define the conditional density for U given Y and Z, $h(u|y,z)$. Now, we can generically use the letter p to express any desired density regarding these variables. To start with, the joint density is*

$$p(x,y,z,u) = \mu(x)f(y|x)g(z|x)h(u|y,z)$$

Once we have the joint distribution $p(x,y,z,u)$, we can derive anything else from it in

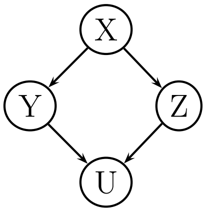


Figure 4.1: Directed acyclic graph showing the (hierarchical) dependency structure for X, Y, Z, U.

terms of the densities we defined μ, f, g, h. Some examples:

$$p(y,z,u) = \int p(x,y,z,u)dx = \int \mu(x)f(y|x)g(z|x)h(u|y,z)dx = h(u|y,z)\int \mu(x)f(y|x)g(z|x)dx$$

$$p(x) = \int p(x,y,z,u)dydzdu = \int \mu(x)f(y|x)g(z|x)h(u|y,z)dydzdu = \mu(x)$$

$$p(y,z,u|x) = \frac{p(x,y,z,u)}{p(x)} = f(y|x)g(z|x)h(u|y,z)$$

$$p(u|x) = \int p(y,z,u|x)dydz = \int f(y|x)g(z|x)h(u|y,z)dydz$$

$$p(x|u) = \frac{p(x)p(u|x)}{p(u)} = \frac{\mu(x)p(u|x)}{\int \mu(x)p(u|x)dx}$$

The dependency structure of this model can be exemplified with

$$x \sim \mu(x), \quad z|x,y \sim g(z|x), \quad u|x,y,z \sim h(\cdot|y,z) \quad etc.$$

or in terms of densities

$$p(x) = \mu(x), \quad p(z|x,y) = p(z|x) = g(z|x), \quad p(u|x,y,z) = p(u|y,z) = h(u|y,z), \quad etc.$$

4.2.2 Conjugate priors

Consider the variables X, Y and Bayes' theorem for $p(x|y)$ in words,

$$\text{posterior} \propto \text{prior} \times \text{likelihood}.$$

In Bayesian statistics, the usual first step to build a statistical model is to decide on the likelihood, i.e. the conditional distribution of the data given the unknown parameter. The likelihood represents the model choice for the data and it should reflect the real stochastic dynamics/phenomena of the data generation process as accurately as possible.

For convenience, it is common to choose a family of parametric distributions for the data likelihood. With such choices x in $p(y|x)$ becomes (some or all of the) parameters of the chosen distribution. For example, $X = (\mu, \sigma^2)$ may be the unknown parameters of a normal distribution from which the data samples $Y_1, \dots, Y_n$ are assumed to be distributed, i.e. $p(y_{1:n}|x) = \prod_{i=1}^n \phi(y_i; \mu, \sigma^2)$. As another example, let $X = \alpha$ be the shape parameter of the gamma distribution $\Gamma(\alpha, \beta)$ and $p(y_{1:n}|x) = \prod_{i=1}^n \frac{e^{-\beta y_i} y_i^{\alpha-1} \beta^\alpha}{\Gamma(\alpha)}$ and β is known.

Bayesian inference for the unknown parameter requires assigning a prior distribution to it. Given the family of distributions for the likelihood, it is sometimes useful to consider a certain family of distributions for the prior distribution so that the posterior distribution has the same form as the prior distribution but with different parameters, i.e. the posterior distribution is in the same family of distributions as the prior. When this is the case, the prior and posterior are then called *conjugate* distributions, and the prior is called a *conjugate prior* for the likelihood $p(y|x)$.

Example 4.4 (**Success probability of the Binomial distribution**). *A certain coin has $\mathbb{P}(\mathrm{T}) = \rho$ where ρ is unknown. The prior distribution is $X = \rho \sim \mathrm{Beta}(a, b)$. The coin is tossed n times, so that if the number of times it brought a tail is Y the conditional distribution for Y is $Y|\rho \sim Binom(n, \rho)$. We want to find the posterior distribution of ρ given $Y = k$ successes out of n trials.*

The posterior density is proportional to

$$p(x|y) \propto p(x)p(y|x) = \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)} \frac{n!}{k!(n-k)!} x^k (1-x)^{n-k} \tag{4.3}$$

where $B(a, b) = \int x^{a-1}(1-x)^{b-1}dx$.

Before continuing with deriving the expression, first note the important remark that our aim here is to recognise the form *of the density of a parametric distribution for x in (4.3). Therefore, we can get rid of any multiplicative term that does not depend on x. That is why we could start with the joint density as $p(x|y) \propto p(x, y)$; in fact we can do more simplification*

$$p(x|y) \propto x^{a+k-1}(1-x)^{b+n-k-1}$$

Since we observe that this has the form of a beta distribution, we can conclude that the posterior distribution has to be *a beta distribution*

$$X|Y = k \sim \mathrm{Beta}(a_{x|y}, b_{x|y})$$

where, from similarity with the prior distribution, we conclude that $a_{x|y} = a + k$ and $b_{x|y} = b + n - k$.

Example 4.5 (**Mean parameter of the normal distribution**). *It is believed that $Y_{1:n} = y_{1:n}$ are samples from a normal distribution with unknown μ and known variance σ^2. We want to estimate μ from $y_{1:n}$. The prior for $X = \mu$ is chosen as $\mathcal{N}(0, \sigma_x^2)$, the conjugate prior of the normal likelihood for the mean parameter. The joint density can be* *written as*

$$\begin{aligned}
p(x|y) \propto p(x,y) &= p(x)p(y|x) \\
&= \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left\{-\frac{1}{2\sigma_x^2}x^2\right\} \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - x)^2\right\} \\
&\propto \exp\left\{-\frac{1}{2\sigma_x^2}x^2 - \frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - x)^2\right\} \\
&= \exp\left\{-\frac{1}{2\sigma_x^2}x^2 - \frac{1}{2\sigma^2}\left(\sum_{i=1}^n y_i^2 + nx^2 - 2x\sum_{i=1}^n y_i\right)\right\} \\
&\propto \exp\left\{-\frac{1}{2\sigma_x^2}x^2 - \frac{1}{2\sigma^2}\left(nx^2 - 2x\sum_{i=1}^n y_i\right)\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[x^2\left(\frac{1}{\sigma_x^2} + \frac{n}{\sigma^2}\right) - 2x\frac{1}{\sigma^2}\sum_{i=1}^n y_i\right]\right\}
\end{aligned}$$

Since we observe that this has the form of a normal distribution, we can conclude that the posterior distribution has to be *a normal distribution*

$$X|Y_{1:n} = y_{1:n} \sim \mathcal{N}(\mu_{x|y}, \sigma_{x|y}^2)$$

for some $\mu_{x|y}$ *and* $\sigma_{x|y}^2$*. In order to find* $\mu_{x|y}$ *and* $\sigma_{x|y}^2$*, compare the expression above with* $\phi(x; m, \kappa^2) \propto \exp\left\{-\frac{1}{2}\left[x^2\frac{1}{\kappa^2} - 2x\frac{m}{\kappa^2} + \frac{m^2}{\kappa^2}\right]\right\}$*. Therefore, we must have*

$$\sigma_{x|y}^2 = \left(\frac{1}{\sigma_x^2} + \frac{n}{\sigma^2}\right)^{-1}, \quad \frac{\mu_{x|y}}{\sigma_{x|y}^2} = \frac{1}{\sigma^2}\sum_{i=1}^n y_i \Rightarrow \mu_{x|y} = \left(\frac{1}{\sigma_x^2} + \frac{n}{\sigma^2}\right)^{-1}\frac{1}{\sigma^2}\sum_{i=1}^n y_i$$

Example 4.6 (**Variance of the normal distribution**). *Consider the scenario in the previous example above but this time* μ *is known and the variance* σ^2 *is unknown. The prior for* $X = \sigma^2$ *is chosen as the conjugate prior of the normal likelihood for the variance parameter, i.e. the inverse gamma distribution* $\mathcal{IG}(\alpha, \beta)$ *with shape and scale parameters* α *and* β*, having the probability density function*

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left(-\frac{\beta}{x}\right).$$

The joint density can be written as

$$\begin{aligned}
p(x|y) \propto p(x,y) &= p(x)p(y|x) \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left(-\frac{\beta}{x}\right) \prod_{i=1}^n \frac{1}{\sqrt{2\pi x}} \exp\left\{-\frac{1}{2x}(y_i - \mu)^2\right\} \\
&\propto x^{-\alpha-n/2-1} \exp\left\{-\frac{\frac{1}{2}\sum_{i=1}^n (y_i - \mu)^2 + \beta}{x}\right\}
\end{aligned}$$

Comparing this expression to the density of $p(x)$, we observe that they have the same form and therefore,

$$X|Y_{1:n} = y_{1:n} \sim \mathcal{IG}(\alpha_{x|y}, \beta_{x|y})$$

for some $\alpha_{x|y}$ and $\beta_{x|y}$. From similarity, we can conclude

$$\alpha_{x|y} = \alpha + \frac{n}{2}, \quad \beta_{x|y} = \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2.$$

Example 4.7 (**Multivariate normal distribution**). *Let the likelihood for Y given X is chosen as $Y|X = x \sim \mathcal{N}(Ax, R)$ and the prior for the unknown X is chosen $X \sim \mathcal{N}(m, S)$. The posterior $p(x|y)$ is*

$$\begin{aligned}
p(x|y) \propto p(x,y) = p(x)p(y|x)& \\
&= \frac{1}{|2\pi S|^{1/2}} \exp\left\{-\frac{1}{2}(x-m)^T S^{-1}(x-m)\right\} \frac{1}{|2\pi R|^{1/2}} \exp\left\{-\frac{1}{2}(y-Ax)^T R^{-1}(y-Ax)\right\} \\
&\propto \exp\left\{-\frac{1}{2}(x^T S^{-1} x - 2m^T S^{-1} x + x^T A^T R^{-1} A x - 2y^T R^{-1} A x)\right\} \\
&= \exp\left\{-\frac{1}{2}\left[x^T (S^{-1} + A^T R^{-1} A) x - 2(m^T S^{-1} + y^T R^{-1} A) x\right]\right\} \\
&\propto \phi(x; m_{x|y}, S_{x|y}) \propto \exp\left\{-\frac{1}{2}\left[x^T S_{x|y}^{-1} x - 2 m_{x|y}^T S_{x|y}^{-1} x\right]\right\}
\end{aligned}$$

where the posterior covariance is

$$S_{x|y} = (S^{-1} + A^T R^{-1} A)^{-1}$$

and the posterior mean is

$$m_{x|y} = S_{x|y}(m^T S^{-1} + y^T R^{-1} A)^T = S_{x|y}(S^{-1} m + A^T R^{-1} y).$$

Computing the evidence: We saw that when conjugate priors are used for the prior, then $p(x)$ and $p(x|y)$ belong to the same family, i.e. their pdf/pmf have the same form. This is nice: since we know $p(x)$, $p(y|x)$, and $p(x|y)$ exactly, we can compute the evidence $p(y)$ for a given y as

$$p(y) = \frac{p(x,y)}{p(x|y)} = \frac{p(x)p(y|x)}{p(x|y)}$$

Example 4.8 (**Success probability of the Binomial distribution - ctd**). *Consider the setting in Example 4.4. Since we know $p_{X|Y}(x|y)$ and $p_{X,Y}(x,y)$ exactly, the evidence $p_Y(y)$ for $y = k$ can be found as*

$$\begin{aligned}
p_Y(k) &= \frac{\frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} \frac{n!}{k!(n-k)!} x^k (1-x)^{n-k}}{\frac{x^{\alpha+k-1}(1-x)^{\beta+n-k-1}}{B(\alpha+k,\beta+n-k)}} && (4.4) \\
&= \frac{n!}{k!(n-k)!} \frac{B(\alpha+k, \beta+n-k)}{B(\alpha,\beta)}. && (4.5)
\end{aligned}$$

which is the pmf, evaluated at k, of the Beta-Binomial distribution with trial parameter n and shape parameters α and β.

4.3 Quantities of interest in Bayesian inference

In Bayesian statistics, the ultimate goal is the posterior distribution $p(x|y)$ of the unknown variable given the available data $Y = y$. There are several quantities one might be interested; all of those these quantities are rooted from $p(x|y)$. The following are some examples of such quantities.

4.3.1 Posterior mean

If we want to have a point estimate about X, one quantity we can look at is the mean posterior

$$\mathbb{E}(X|Y = y) = \int p(x|y)xdx$$

Other than being an intuitive choice, $\mathbb{E}(X|Y)$, as a random function of Y, is justified in the frequentist setting as well, due to the fact that $\mathbb{E}(X|Y)$ minimises the expected mean squared error

$$\text{MSE} = \mathbb{E}\left([X - \hat{X}(Y)]^2\right) = \int (x - \hat{X}(y))^2 p(x, y)dxdy$$

where $\hat{X}(Y)$ is the estimator for X and the expectation is taken with respect to the joint distribution of X, Y.

Theorem 4.1. $\hat{X}(Y) = \mathbb{E}(X|Y)$ *minimises MSE.*

In general, if want to estimate $\varphi(X)$ given Y, we can target the posterior mean of φ

$$\mathbb{E}(\varphi(X)|Y = y) = \int p(x|y)\varphi(x)dx,$$

which minimises the expected mean squared error for $\varphi(X)$

$$\mathbb{E}\left([\varphi(X) - \hat{\varphi}(Y)]^2\right) = \int (\varphi(x) - \hat{\varphi}(y))^2 p(x, y)dxdy.$$

Although it has nice statistical properties as mentioned above, the posterior mean may not always be a good choice. For example, suppose the posterior is a mixture of Gaussians with pdf $p(x|y) = 0.5\phi(x; -10, 0.01) + 0.5\phi(x; 10, 0.01)$. The posterior mean is 0 but density of $p(x|y)$ at 0 is almost 0 and the distribution has almost no mass around 0!

4.3.2 Maximum a posteriori estimation

Another point estimate that is derived from the posterior is the maximum a posteriori (MAP) estimate which is the maximising argument of $p(x|y)$

$$\hat{x}_{\text{MAP}} = \arg\max_{x\in\mathcal{X}} p(x|y) = \arg\max_{x\in\mathcal{X}} p(x,y).$$

Note that this procedure is different than maximum likelihood estimation (MLE), which yields the maximising argument of the likelihood

$$\hat{x}_{\text{MLE}} = \arg\max_{x\in\mathcal{X}} p(y|x),$$

since in the MAP estimate there is the additional factor due to prior $p(x)$.

4.3.3 Posterior predictive distribution

Assume we are interested in the distribution that a new data point Y_{n+1} would have, given a set of n existing observations $Y_{1:n} = y_{1:n}$. In a frequentist context, this might be derived by computing the maximum likelihood estimate $\hat{x}_{\text{MLE}}$ (or some other point estimate) of X given $y_{1:n}$, and then plugging it into the distribution function of the new observation Y_{n+1} so that the predictive distribution is $p(y_{n+1}|\hat{x}_{\text{MLE}})$.

In a Bayesian context, the natural answer to this is the posterior predictive distribution, which is the distribution of unobserved observations (prediction) conditional on the observed data $p(y_{n+1}|y_{1:n})$. In order to find the posterior predictive distribution, we make use of the entire posterior distribution of the parameter(s) given the observed data to yield a probability distribution rather than simply a point estimate. Specifically, we compute $p(y_{n+1}|y_{1:n})$ by marginalising over the unknown variable x, using its posterior distribution:

$$\begin{aligned} p(y_{n+1}|y_{1:n}) &= \int p(y_{n+1}, x|y_{1:n})dx \\ &= \int p(y_{n+1}|x, y_{1:n})p(x|y_{1:n})dx \end{aligned}$$

In many cases, Y_{n+1} is independent from $Y_{1:n}$ given X. This happens, for example, when $\{Y_i\}_{i\geq 1}$ are i.i.d. given X, that is $Y_i|X = x \sim p(y|x)$, $i \geq 1$. In that case, the density above reduces to

$$p(y_{n+1}|y_{1:n}) = \int p(y_{n+1}|x)p(x|y_{1:n})dx$$

Note that this is equivalent to the expected value of the distribution of the new data point, when the expectation is taken over the posterior distribution, i.e.:

$$p(y_{n+1}|y_{1:n}) = \mathbb{E}[p(y_{n+1}|X)|Y_{1:n} = y_{1:n}].$$

Conjugate priors and posterior predictive density: We saw that when conjugate priors are used for the prior, then $p(x)$ and $p(x|y)$ belong to the same family, i.e. their pdf/pmf have the same form. This implies that, when Y_i's are i.i.d. conditional on X, the posterior predictive density $p(y_{n+1}|y_{1:n})$ has the same form as the marginal density of a single sample

$$p(y) = \int p(x)p(y|x)dx.$$

Example 4.9 (**Success probability of the Binomial distribution - ctd**). *Consider the setting in Example 4.4. Given the prior $X \sim \mathrm{Beta}(\alpha, \beta)$ and $Y = k$ successes out of n trials, what is the probability of having $Z = r$ successes out of the next m trials?*

Here Z is the next sample that is to be predicted. We can employ the posterior predictive probability for Z. We know from the derivation of Example 4.8 that Z will be distributed from the Beta-Binomial distribution with parameters m (trials), $\alpha' = \alpha + k$ and $\beta' = \beta + n - k$ since the prior and the posterior of X are in the same form and Z given $X = x$ and Y given $X = x$ are both Binomial.

$$p_{Z|Y}(r|k) = \frac{m!}{r!(m-r)!} \frac{B(\alpha' + r, \beta' + m - r)}{B(\alpha', \beta')}.$$

Exercises

1. Consider the discrete random variables $X \in \{1, 2, 3\}$ and $Y \in \{1, 2, 3, 4\}$ whose joint probabilities are given in Table 4.1

$p_{X,Y}(x, y)$	$y = 1$	$y = 2$	$y = 3$	$y = 4$	$p_X(x)$
$x = 1$	$1/40$	$3/40$	$4/40$	$2/40$	
$x = 2$	$5/40$	$7/40$	$6/40$	$5/40$	
$x = 3$	$1/40$	$2/40$	$2/40$	$2/40$	
$p_Y(y)$					

Table 4.1: Joint probability table

- Find the marginal probabilities $p_X(x)$ and $p_Y(y)$ for all $x = 1, 2, 3$, $y = 1, 2, 3, 4$ and fill in the rest of Table 4.1.
- Find the conditional probabilities $p_{X|Y}(x|y)$ and $p_{Y|X}(y|x)$ for all $x = 1, 2, 3$, $y = 1, 2, 3, 4$ and fill in the relevant empty tables.

$p_{X\|Y}(x\|y)$	$y = 1$	$y = 2$	$y = 3$	$y = 4$
$x = 1$				
$x = 2$				
$x = 3$				

$p_{Y\|X}(y\|x)$	$y = 1$	$y = 2$	$y = 3$	$y = 4$
$x = 1$				
$x = 2$				
$x = 3$				

2. Show that the gamma distribution is the conjugate prior of the exponential distribution for, i.e. if $X \sim \Gamma(\alpha, \beta)$ and $Y|X = x \sim \text{Exp}(x)$, then $X|Y = y \sim \Gamma(\alpha_{x|y}, \beta_{x|y})$ for some $\alpha_{x|y}$ and $\beta_{x|y}$. Find $\alpha_{x|y}$ and $\beta_{x|y}$ in terms of α, β, and y.

3. Prove Theorem 4.1 [Hint: write the estimator as $\hat{X}(Y) = \mathbb{E}(X|Y) + (\hat{X}(Y) - \mathbb{E}(X|Y))$ and consider conditional expectation of the MSE given $Y = y$ first. You should conclude that for any y, $\hat{X}(y) - \mathbb{E}(X|Y = y)$ should be zero.]

4. Suppose we observe a noisy sinusoid with period T and unknown amplitude X for n steps: $Y|X = x \sim \mathcal{N}(y_t; f(x, t), \sigma_y^2)$, for $t = 1, \ldots, n$ where $f(t; x) = x \sin(2\pi t/T)$ is the sinusoid. The prior for the amplitude is Gaussian: $X \sim \mathcal{N}(0, \sigma_x^2)$.

- Find $p(x|y_{1:n})$ and $p(y_{1:n})$.
- What is distribution of $f(n + 1, X)$ given $Y_{1:n} = y_{1:n}$?

- Find $p(y_{n+1})$ and $p(y_{n+1}|y_{1:n})$. Compare their variances. What can you comment on the difference between the variances?
- Generate your own samples $Y_{1:n}$ up to time $n = 100$, with period $T = 40$, $\sigma_x^2 = 100$, $\sigma_y^2 = 10$. Calculate $p(x|y_{1:n})$; plot $p(y_{n+1})$ and $p(y_{n+1}|y_{1:n})$ on the same axis.

Chapter 5

Markov Chain Monte Carlo

Summary: *In this chapter, we will see an essential and vast family of methods in Monte Carlo, namely Markov chain Monte Carlo methods, for approximately sampling from complex distributions. We will start the chapter with a review of discrete time Markov chains, which is required for understanding the working principles of Markov chain Monte Carlo methods. Then, we will see two most commonly used Markov chain Monte Carlo methods in the literature: Metropolis-Hastings and Gibbs sampling methods.*

5.1 Introduction

Remark 5.1 (**Change of notation**)**.** *So far we have used P and p to denote the distribution and its pdf/pmf we are ultimately interested in. We will make a change of notation here, and denote the distribution as well as its pdf/pmf as π. This change of notation is necessary since p will be used generically to denote the pdf/pmf of various distributions.*

We have already discussed the difficulties of generating a large number of i.i.d. samples from π. One alternative was importance sampling which involved weighting every generated sample in order not to waste it, but it has its own drawbacks mostly due to issues related to controlling variance. Another alternative is to use *Markov chain Monte Carlo* (MCMC) methods (Metropolis et al., 1953; Hastings, 1970; Gilks et al., 1996; Robert and Casella, 2004). These methods are based on the design of a suitable ergodic Markov chain whose stationary distribution is π. The idea is that if one simulates such a Markov chain, after a long enough time the samples of the Markov chain will approximately distributed according to π. Although the samples generated from the Markov chain are not i.i.d., their use is justified by convergence results for dependent random variables in the literature. First examples of MCMC can be found in Metropolis et al. (1953); Hastings (1970), and book length reviews are available in Gilks et al. (1996); Robert and Casella (2004).

5.2 Discrete time Markov chains

In order to adequately summarise the MCMC methodology, we first need reference to the theory of discrete time Markov chains defined on general state spaces. Discrete time Markov chains also constitute an important part of the rest of this course, especially when we discuss sequential Monte Carlo methods. The review made here is very brief and limited by the relation of Markov chains to the topics covered in the course. For more details one can see Meyn and Tweedie (2009) or Shiryaev (1995); a more related introduction to Monte Carlo methods is present in Robert and Casella (2004, Chapter 6) and Cappé et al. (2005, Chapter 14), Tierney (1994) and Gilks et al. (1996, Chapter 4).

Definition 5.1 (**Markov chain**). *A stochastic process $\{X_n\}_{n\geq 1}$ on $\mathcal{X}$ is called a Markov chain if its probability law is defined from the initial distribution $\eta(x)$ and a sequence of Markov transition (or transition, state transition) kernels (or probabilities, densities) $\{M_n(x'|x)\}_{n\geq 2}$ by finite dimensional joint distributions as*

$$p(x_1, \dots, x_n) = \eta(x_1)M_2(x_2|x_1)\dots M_n(x_n|x_{n-1})$$

for all $n \geq 1$.

The random variable X_t is called the *state* of the chain at time t and $\mathcal{X}$ is called the *state-space* of the chain. For uncountable $\mathcal{X}$, we have a discrete-time continuous-state Markov chain, and $\eta(\cdot)$ and $M_n(\cdot|x_{n-1})$ are pdf's[1]. Similarly, $\mathcal{X}$ is countable (finite or infinite), then the chain is a discrete-time discrete-state Markov chain and $\eta(\cdot)$ and $M_n(\cdot|x_{n-1})$ are pmf's. Moreover, when $\mathcal{X} = \{x_1, \dots, x_m\}$ is finite with m states, the transition kernel can sometimes be expressed in terms of an $m \times m$ transition matrix $M_n(i,j) = \mathbb{P}(X_n = j|X_{n-1} = i)$.

The definition of the Markov chain leads to the characteristic property of a Markov chain, which is also referred to as the *weak Markov property*: The current state of the chain at time n depends only on the previous state at time $n-1$.

$$p(x_n|x_{1:n-1}) = p(x_n|x_{n-1}) = M_n(x_{n-1}, x_n)$$

From now on, we will consider *time-homogenous* Markov chains where $M_n = M$ for all $n \geq 2$, and we will denote them as $\text{Markov}(\eta, M)$.

Example 5.1. *The simplest examples of a Markov chain are those with a finite state-space, say of size m. Then, the transition rule can be expressed by an $m \times m$ transition probability matrix M, which in this example is the following*

$$M = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 0 & 1 & 0 \end{bmatrix}$$

Also, the state-transition diagram of such a Markov chain with $m = 3$ states is given in Figure 5.1, where the state-space is simply $\{1, 2, 3\}$.

[1] In fact, there are exceptions where the transition kernels do not have a probability density; and this is indeed the case for the transition kernel of the Markov chain of the Metropolis-Hastings algorithm which we will see in Section 5.3. However, for the sake of brevity we ignore this technical issue and with abuse of notation pretend as if we always have a density for $M_n(\cdot|x_{n-1})$ for continuous states

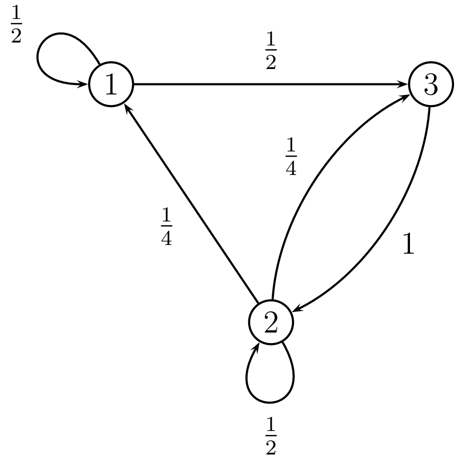


Figure 5.1: State transition diagram of a Markov chain with 3 states, $1, 2, 3$.

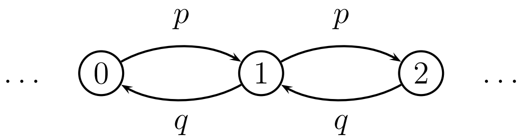


Figure 5.2: State transition diagram of the symmetric random walk on $\mathbb{Z}$.

Example 5.2. *Let $\mathcal{X} = \mathbb{Z}$ be the set of integers, $X_1 = 0$, and for $n > 1$ define X_n as*

$$X_n = X_{n-1} + V_n,$$

where $V_n \in \{-1, 1\}$ with $p = \mathbb{P}(V_n = 1) = 1 - \mathbb{P}(V_n = -1) = 1 - q$. This is a random walk (of step-size 1) on $\mathbb{Z}$ and it is a time homogenous discrete-time discrete state Markov chain with $\eta(x_1) = \delta_0(x_1)$ and

$$M(x'|x) = \begin{cases} p, & x' = x + 1 \\ q, & x' = x - 1 \end{cases}$$

When $p = q$, the process is called a symmetric random walk.

Example 5.3. *Let $\mathcal{X} = \mathbb{R}$, $X_1 = 0$, and for $n > 1$ define X_n as*

$$X_n = X_{n-1} + V_n,$$

but this time $V_n \in \mathbb{R}$ with $V_n \sim \mathcal{N}(0, \sigma^2)$. This is a Gaussian random walk process on $\mathbb{R}$ with normally distributed step sizes, and it is a time homogenous discrete-time continuous state Markov chain with $\eta(x_1) = \delta_0(x_1)$ and

$$M(x'|x) = \phi(x'; x, \sigma^2).$$

Example 5.4. *A generalisation of the Gaussian random walk is the first order autoregressive process, or shortly AR(1). Let $\mathcal{X} = \mathbb{R}$ the set of integers, $X_1 = 0$, and for $n > 1$ define X_n as*

$$X_n = aX_{n-1} + V_n,$$

for some $a \in \mathbb{R}$, and $V_n \in \mathbb{R}$ with $V_n \sim \mathcal{N}(0, \sigma^2)$. AR(1) is a time homogenous discrete-time continuous state Markov chain with $\eta(x_1) = \delta_0(x_1)$ and

$$M(x'|x) = \phi(x'; ax, \sigma^2).$$

When $|a| < 1$, another choice for the initial distribution is $X_1 \sim \mathcal{N}(0, \frac{\sigma^2}{1-a^2})$, which is the stationary distribution of $\{X_t\}_{t \geq 1}$. We will see more on the stationary distributions below.

5.2.1 Properties of Markov(η, M)

For MCMC, we require the Markov chain to have a unique invariant distribution π and to converge to π. Before discussing that, we need to review some fundamental properties of a discrete time Markov chain to understand when the existence of an invariant distribution and convergence to it are ensured. Those properties will be discussed in specific to discrete-state Markov chains only, for sake of simplicity and delivering the intuition behind the concepts. Although for general state-space Markov chains similar concepts also exist, they are more complicated and with less intuition, due to which we mostly omit them from our review.

5.2.1.1 Irreducibility

In a discrete state Markov chain, for two states $x, x' \in \mathcal{X}$, we say x *leads to* x' and show it by $x \to x'$ if the chain can travel from x to x' with a positive probability, i.e.

$$\exists n > 1 \text{ s.t. } \mathbb{P}(X_n = x'|X_1 = x) > 0$$

If both $x \to x'$ and $x' \to x$, we say x and x' *communicate* and we show it by $x \leftrightarrow x'$.

A subset of states $C \subseteq \mathcal{X}$ is called a *communicating class*, or simply *class*, if (i) all $x, x' \in C$ communicate, and (ii) $x \in C$, $x \leftrightarrow y$ together imply $y \in C$, too (that is, there is no such $y \notin C$ such that $x \leftrightarrow y$ for some $x \in C$).

A communicating class is *closed* if $x \in C$ and $x \to y$ imply $y \in C$, that is there is no path with positive probability from outside the class to any of the states of the class.

Definition 5.2 (Irreducibiliy). *A discrete state Markov chain is called irreducible if the whole $\mathcal{X}$ is a communication class, i.e. all its states communicate.*

For general state-spaces, we need to generalise the concept of irreducibility to ϕ-*irreducibility.*

Example 5.5. *Figure 5.3 shows two chains that are not irreducible. In the first chain, the communication classes are $\{1, 2, 3\}$ and $\{4, 5\}$; both are closed. In the second chain, the communication classes are $\{1, 2\}$ and $\{3, 4\}$; the first one is closed and the second one is not.*

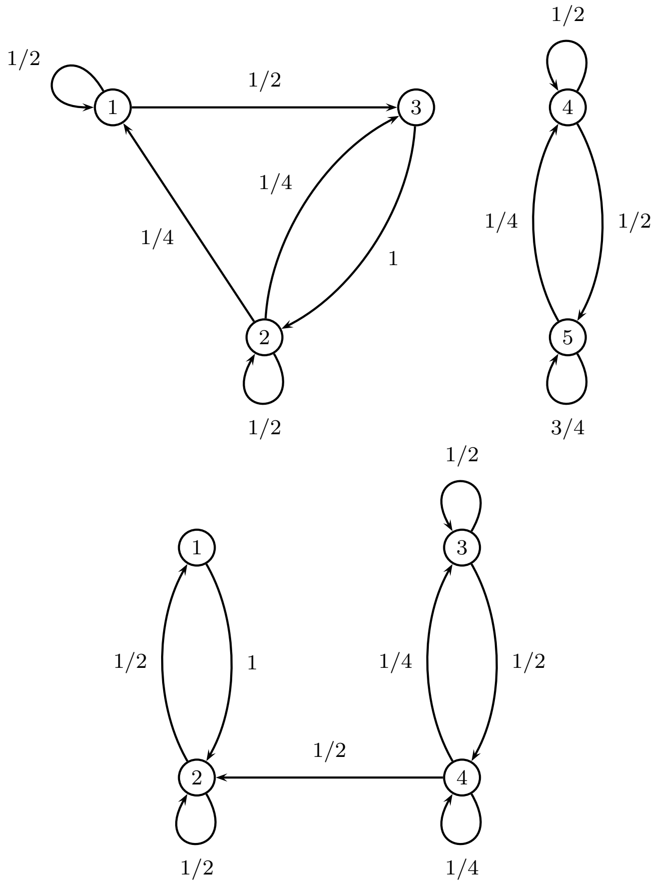


Figure 5.3: State transition diagrams of two Markov chains that are not irreducible.

5.2.1.2 Recurrence and Transience

In the discrete state-space, we say that a Markov chain is *recurrent* if every of its states is expected to be visited by the chain infinitely often, otherwise it is *transient*. More precisely, define the return time

$$\tau_x = \min\{n \geq 1 : X_{n+1} = x\}$$

Definition 5.3 (Recurrence). *We say the state $x \in \mathcal{X}$ is recurrent if*

$$\mathbb{P}(\tau_x < \infty | X_1 = x) = 1 \tag{5.1}$$

or equivalently $\sum_{n=1}^{\infty} \mathbb{P}(X_n = x | X_1 = x) = \infty$. If a state is not recurrent, it is called transient.

If M is irreducible, then either every state is recurrent (and M is said to be recurrent) or every state is transient (and M is said to be transient).

Example 5.6. *The random walk on integers in Example 5.2 is an irreducible chain. It can be shown that, in the symmetric case when $p = q = 1/2$, the chain is recurrent; if $p \neq q$, the chain is transient.*

Definition 5.4 (Positive recurrence and null recurrence). *We say a state $x \in \mathcal{X}$ is positive recurrent if*

$$\mathbb{E}(\tau_x | X_1 = x) < \infty \tag{5.2}$$

(Note that (5.2) is a stronger condition than (5.1).) If a recurrent state is not positive recurrent, it is called null recurrent.

If M is irreducible and recurrent, then either every state is positive recurrent (and M is said to be positive recurrent) or every state is null recurrent (and M is said to be null recurrent).

To talk about recurrence in general state-space chains, instead of states we consider *accessible sets* in relation to ϕ-irreducibility.

Example 5.7. *It can be shown that the random walk on integers in Example 5.2 is a null recurrent chain for $p = q = 1/2$.*

5.2.1.3 Invariant distribution

A probability distribution π is called M-invariant if

$$\pi(x) = \int \pi(x') M(x|x') dx'$$

where we have assumed that $\{X_t\}_{t\geq 1}$ is continuous (hence π is a pdf). When $\{X_t\}_{t\geq 1}$ is discrete (hence π is a pmf), this relation is written as

$$\pi(x) = \sum_{x'} \pi(x') M(x|x')$$

The expressions on the RHS of the two equations above are short-handedly written as πM, so that for invariant π we have $\pi = \pi M$. In fact, when $\mathcal{X} = \{x_1, \ldots, x_m\}$ is finite with $M(i,j) = \mathbb{P}(X_n = j | X_{n-1} = i)$ and $\pi = \begin{bmatrix} \pi(1) & \ldots & \pi(m) \end{bmatrix}$, we can indeed write $\pi = \pi M$ using notation for vector matrix multiplication.

Theorem 5.1 (**Existence and uniqueness of invariant distribution**). *Suppose M is irreducible. M has a unique invariant distribution if and only if it is positive recurrent.*

Example 5.8. *The chain in Example 5.1 has the invariant distribution $\pi = \begin{bmatrix} 1/4 & 1/2 & 1/4 \end{bmatrix}$. By solving $\mu = \mu M$, it can be shown that π is the only invariant distribution, so the chain is positive recurrent.*

Example 5.9. *The random walk on integers in Example 5.2 is irreducible. Therefore, it does not have an invariant distribution since it is not positive recurrent for any choice of $p = 1 - q$.*

Example 5.10. *The Markov chain on top of Figure 5.3 has two invariant distributions $\pi = \begin{bmatrix} 1/4 & 1/2 & 1/4 & 0 & 0 \end{bmatrix}$ and $\pi = \begin{bmatrix} 0 & 0 & 0 & 1/3 & 2/3 \end{bmatrix}$ although every state is positive recurrent. Note that the chain is not irreducible with two isolated communication classes, that is why Theorem 5.1 is not applicable and uniqueness may not follow.*

Example 5.11. *The Markov chain at the bottom of Figure 5.3 is neither irreducible nor all of its states are positive recurrent (the states of the second class are transient). However, it has a unique invariant distribution, namely $\pi = \begin{bmatrix} 1/3 & 2/3 & 0 & 0 \end{bmatrix}$. Note that for this chain Theorem 5.1 is not applicable since the chain is not irreducible.*

5.2.1.4 Reversibility and detailed balance

One useful way for spotting the existence of an invariant probability measure for a Markov chain is to check for its *reversibility*, which is a sufficient (but not necessary) condition for existence of a stationary distribution.

Definition 5.5 (**reversibility**). *Let M be a transitional kernel having an invariant distribution and assume the associated Markov chain is started from π. We say that M is reversible if the reversed process $\{X_{n-m}\}_{0 \leq m < n}$ is also $Markov(\pi, M)$ for all $n \geq 1$.*

According to the definition above, M is reversible with respect to π if the backward transition density of the process $\{X_n\}_{n \geq 1}$ with $X_1 \sim \pi$ is the same as its forward transition density, i.e.

$$p(x_{n-1}|x_n) = \frac{p(x_{n-1})p(x_n|x_{n-1})}{p(x_n)} = \frac{p(x_{n-1})M(x_n|x_{n-1})}{\int p(x_{n-1})M(x_n|x_{n-1})dx_{n-1}} = M(x_{n-1}|x_n).$$

This immediately leads to the necessary and sufficient condition for reversibility of M is the detailed balance condition.

Proposition 5.1 (**detailed balance**). *We say a Markov kernel M is reversible with respect to a probability distribution π if and only if the following condition, known as the detailed balance condition, holds:*

$$\pi(x)M(y|x) = \pi(y)M(x|y), \quad x, y \in \mathcal{X}.$$

Also, then π is an invariant distribution for M.

Being a sufficient condition for stationarity, the detailed balance condition is quite useful for designing transition kernels for MCMC algorithms.

5.2.1.5 Ergodicity

Let π_n be the distribution of X_n of a Markov chain $\{X_n\}_{n\geq 1}$ with initial distribution η and transition kernels M. We have $\pi_1(x_1) = \eta(x_1)$ and the rest can be written recursively as $\pi_n = \pi_{n-1}M$, or explicitly

$$\pi_n(x_n) = \int \pi_{n-1}(x_{n-1})M(x_n|x_{n-1})dx_{n-1}$$

for continuous state chains, or

$$\pi_n(x_n) = \sum_{x_{n-1}\in\mathcal{X}} \pi_{n-1}(x_{n-1})M(x_n|x_{n-1}),$$

for discrete state chains, which reduces to

$$\pi_n = \pi_{n-1}M$$

when the state space is finite and π and M are considered as a vector and a matrix, respectively.

In MCMC methods that aim to approximately sample from π, we generate a Markov chain $\{X_n\}_{n\geq 1}$ with invariant distribution π and hope that for n large enough X_n is approximately distributed from π. This relies on the hope that π_n converges to π.

We have shown the conditions for a unique stationary distribution of a Markov chain. Note that having a unique invariant distribution does not mean that the chain will converge to its stationary distribution. For that to happen the Markov chain is required to have *aperiodicity*, a property which restricts the chain from getting trapped in cycles.

Definition 5.6 (**aperiodicity**). *In a discrete state Markov chain, a state $x \in \mathcal{X}$ is called* aperiodic *if the set*

$$\{n > 0 : \mathbb{P}(X_{n+1} = x|X_1 = x)\}$$

has no common divisor other than 1. *Otherwise, the state is* periodic *and its period is the greatest common divisor of state x. The Markov chain is said to be aperiodic if all of its states are aperiodic.*

If the Markov chain is irreducible, then aperiodicity of one state implies the aperiodicity of all the states.

Definition 5.7 (ergodic state). *A state is called ergodic if it is positive recurrent and aperiodic.*

Finally, the definition of ergodicity for a Markov chain follows.

Definition 5.8 (ergodic Markov chain). *An irreducible Markov chain is called ergodic if it is positive recurrent and aperiodic.*

Ergodic chains ensure that the sequence of distributions $\{\pi_n\}_{n\geq 1}$ for $\{X_n\}_{n\geq 1}$ converge to the invariant distribution π.

Theorem 5.2. *Suppose $\{X_n\}_{n\geq 1}$ is a discrete-state ergodic Markov chain with any initial distribution η and Markov transition kernel M with invariant distribution π. Then,*

$$\lim_{n\to\infty} \pi_n(x) = \pi(x) \tag{5.3}$$

In particular, for all $x, x' \in \mathcal{X}$,

$$\lim_{n\to\infty} P(X_n = x | X_1 = x') = \pi(x)$$

Example 5.12. *The Markov chain illustrated in Figure 5.4 is irreducible and positive recurrent; so it has a unique invariant distribution, which is $\pi = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$. However, it is periodic with period 3, and as a result π_n does not converge to π unless $\eta = \pi$. Indeed, one can show that for $\eta = \begin{bmatrix} \eta(1) & \eta(2) & \eta(3) \end{bmatrix}$, we have*

$$\pi_n = \eta M^{n-1} = \begin{bmatrix} \eta(\mathrm{mod}(n-1,3)+1) & \eta(\mathrm{mod}(n-1,3)+2) & \eta(\mathrm{mod}(n-1,3)+3) \end{bmatrix}.$$

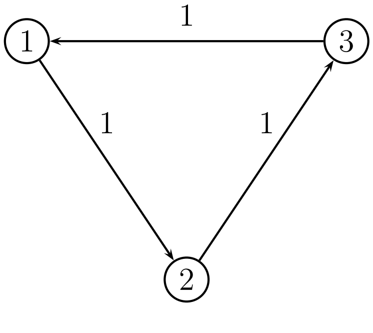


Figure 5.4: An irreducible, positive recurrent, and periodic Markov chain.

5.3 Metropolis-Hastings

As previously stated, an MCMC method is based on a discrete-time ergodic Markov chain which has its stationary distribution as π. The most widely used MCMC algorithm up to date is the *Metropolis-Hastings* algorithm (Metropolis et al., 1953; Hastings, 1970).

The Metropolis-Hastings algorithm requires a Markov transition kernel Q on $\mathcal{X}$ for proposing new values from the old ones. Assume that the pdf/pmf of $Q(\cdot|x)$ is $q(\cdot|x)$ for any x. Given the previous sample X_{n-1} a new value for X_n is *proposed* as $X' \sim Q(\cdot|X_{n-1})$. The proposed sample X' is accepted with the acceptance probability $\alpha(X_{n-1}, X')$, where the function $\alpha : \mathcal{X} \times \mathcal{X} \to [0, 1]$ is defined as

$$\alpha(x, x') = \min\left\{1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right\}, \quad x, x' \in \mathcal{X}.$$

If the proposal is accepted, $X_n = X'$ is taken. Otherwise, the proposal is rejected and $X_n = X_{n-1}$ is taken.

Algorithm 5.1: Metropolis-Hastings

```
Begin with some $X_1 \in \mathcal{X}$.
for $n = 2, 3, \dots$ do
    Sample $X' \sim Q(\cdot|X_{n-1})$.
    Set $X_n = X'$ with probability
        $$\alpha(X_{n-1}, X') = \min\left\{1, \frac{\pi(X')q(X_{n-1}|X')}{\pi(X_{n-1})q(X'|X_{n-1})}\right\},$$
    else set $X_n = X_{n-1}$.
```

The ratio in the acceptance probability

$$r(x, x') = \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}$$

is called the acceptance ratio, or the acceptance rate.

The invariant distribution of the Metropolis-Hastings algorithm described exists and it is π. In order to show this, we can check for the detailed balance condition. According to Algorithm 5.1, the transition kernel M of the Markov chain from which the samples are obtained is

$$M(y|x) = q(y|x)\alpha(x, y) + p_r(x)\delta_x(y),$$

where $p_r(x)$ is the rejection probability at x and

$$p_r(x) = \left[1 - \int q(x'|x)\alpha(x, x')dx'\right], \quad \text{or} \quad p_r(x) = \left[1 - \sum_{x'} q(x'|x)\alpha(x, x')\right]$$

depending on the nature of the state-space. For all $x, y \in \mathcal{X}$, we have

$$\begin{aligned}\pi(x)M(y|x) &= \pi(x)q(y|x)\alpha(x,y) + \pi(x)p_r(x)\delta_x(y)\\ &= \pi(x)q(y|x)\min\left\{1, \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)}\right\} + \pi(x)p_r(x)\delta_x(y)\\ &= \min\left\{\pi(x)q(y|x), \pi(y)q(x|y)\right\} + \pi(x)p_r(x)\delta_x(y)\\ &= \min\left\{\pi(y)q(x|y), \pi(x)q(y|x)\right\} + \pi(y)p_r(y)\delta_y(x)\end{aligned}$$

which is symmetric with respect to x, y, so $\pi(x)M(y|x) = \pi(y)M(x|y)$ and the detailed balance condition holds for π which implies that M is reversible with respect to π and π is invariant for M.

Note that, as long as discrete-state chains are considered, existence of the invariant distribution π for M ensures the positive recurrence of M. There are also various sufficient conditions for the M of the Metropolis-Hastings algorithm to be irreducible and aperiodic. For example, if Q is irreducible and $\alpha(x,y) > 0$ for all $x, y \in \mathcal{X}$, then M is irreducible. If $p_r(x) > 0$ for all x or Q is aperiodic then M is aperiodic (Roberts and Smith, 1994). More detailed results on the convergence of Metropolis-Hastings are also available, see e.g. Tierney (1994); Roberts and Tweedie (1996) and Mengersen and Tweedie (1996).

Historically, the original MCMC algorithm was introduced by Metropolis et al. (1953) for the purpose of optimisation on a discrete state-space. This algorithm, called the *Metropolis algorithm*, used symmetrical proposal kernels Q, that is $q(x'|x) = q(x|x')$. When a symmetric proposal is used, the acceptance probability involves only the ratio of the target distribution evaluated at x and x',

$$\alpha(x, x') = \min\left\{1, \frac{\pi(x')}{\pi(x)}\right\}, \quad \text{if} \quad q(x'|x) = q(x|x').$$

The Metropolis algorithm was later generalised by Hastings (1970) such that it permitted continuous state-spaces and asymmetrical proposal kernels, preserving the Metropolis algorithm as a special case. A more historical survey on Metropolis-Hastings algorithms is provided by Hitchcock (2003).

Another version is the independence Metropolis-Hastings algorithm, where, as the name suggests, the proposal kernel Q is chosen to be independent from the current value, i.e. $q(x'|x) = q(x')$, in which case the acceptance probability is

$$\alpha(x, x') = \min\left\{1, \frac{\pi(x')q(x)}{\pi(x)q(x')}\right\}.$$

5.3.1 Toy example: MH for the normal distribution

This is a toy example where $\pi(x) = \phi(x; \mu, \sigma^2)$ for which we do not need to use MH since we can obviously sample from $\mathcal{N}(\mu, \sigma^2)$ easily. But for the sake of example assume that we have decided to use MH to generate approximate samples from π.

For the proposal kernel, we have several options:

- Symmetric random walk: We can take $q(x'|x) = \phi(x'; x, \sigma_q^2)$, that is x' is proposed from the current value x by adding a normal random variable with zero mean and variance σ_q^2, or $Q(\cdot|x) \sim \mathcal{N}(x, \sigma_q^2)$. Since

  $$q(x'|x) = \phi(x'; x, \sigma_q^2) = \phi(x; x', \sigma_q^2) = q(x|x'),$$

  this results in the acceptance ratio

  $$\begin{aligned} r(x, x') &= \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)} \\ &= \frac{\phi(x'; \mu, \sigma^2)}{\phi(x; \mu, \sigma^2)} \\ &= \frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x'-\mu)^2}}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}} \\ &= e^{-\frac{1}{2\sigma^2}\left[(x'-\mu)^2-(x-\mu)^2\right]} \end{aligned}$$

  The choice of σ_q^2 is important for good performance of MH. We want the Markov chain generated by the algorithm to *mix* well, that is we want the samples to forget the previous values fast. Consider the acceptance ratio above:

  - A too small value for σ_q^2 will result in the acceptance ratio $r(x, x')$ being very close to 1, and hence the proposed values will be accepted with high probability. However, the chain will be very slowly mixing, that is the samples will be highly correlated; because any accepted sample x' will most likely be only slightly different than the current x due to a small step-size of the random walk.
  - A too large value for σ_q^2 will likely result in the proposed value x' to be far from the region where π has most of its mass, hence $\pi(x')$ will be very small compared to $\pi(x)$ and the chain will likely reject the proposed value and stick to the old value x. This will create a *sticky* chain.

  Therefore, the optimum value for σ_q^2 should be neither too small or too large. See Figure 5.5 for the both bad choices and one in between those. This phenomenon of having to choose the variance of the random walk proposals neither too small nor too big is also valid for most distributions than the normal distribution.

- Another option for the proposal is to sample x' independently from x, i.e. $q(x'|x) = q(x')$. For example, suppose we chose $q(x) = \phi(x; \mu_q, \sigma_q^2)$. Then the acceptance ratio

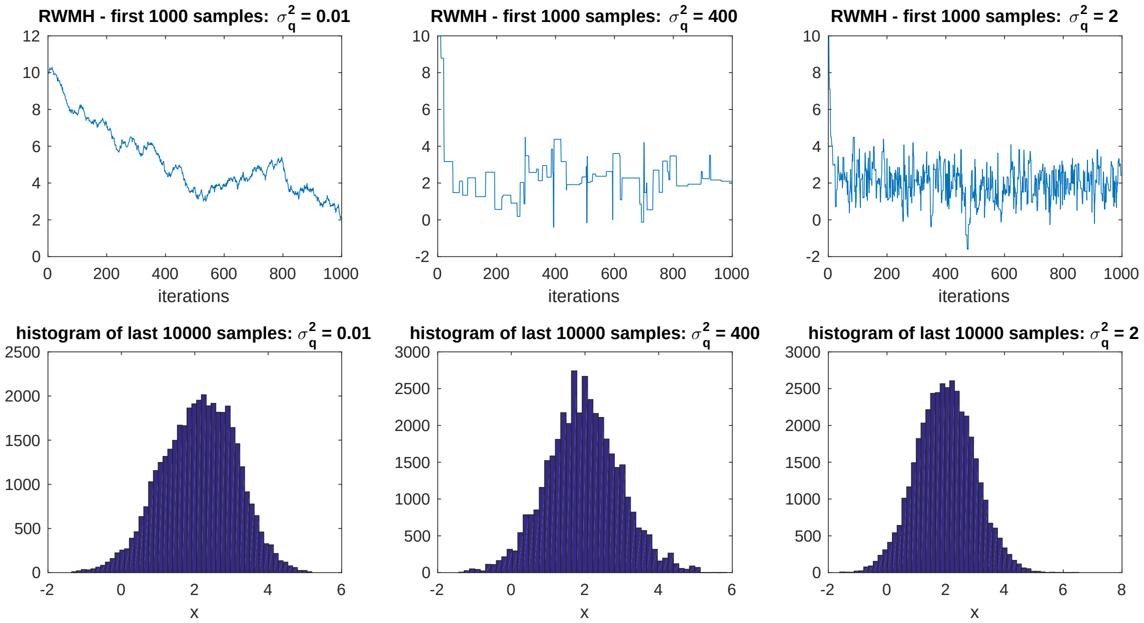


Figure 5.5: Random walk MH for $\pi(x) = \phi(x; 2, 1)$. The left and middle plots correspond to a too small and a too large value for σ_q^2, respectively. All algorithms are run for 50000 iterations. Both the trace plots and the histograms show that the last choice works the best.

is

$$\begin{aligned} r(x, x') &= \frac{\pi(x')q(x)}{\pi(x)q(x')} \\ &= \frac{\phi(x'; \mu, \sigma^2)\phi(x; \mu_q, \sigma_q^2)}{\phi(x; \mu, \sigma^2)\phi(x'; \mu_q, \sigma_q^2)} \\ &= \frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x'-\mu)^2} \frac{1}{\sqrt{2\pi\sigma_q^2}} e^{-\frac{1}{2\sigma_q^2}(x-\mu_q)^2}}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \frac{1}{\sqrt{2\pi\sigma_q^2}} e^{-\frac{1}{2\sigma_q^2}(x'-\mu_q)^2}} \\ &= e^{-\frac{1}{2\sigma^2}\left[(x'-\mu)^2-(x-\mu)^2\right]+\frac{1}{2\sigma_q^2}\left[(x'-\mu_q)^2-(x-\mu_q)^2\right]} \end{aligned}$$

See Figure 5.6 for examples of MH with this choice.

- Another alternative is to use a gradient-guided proposal. We may want to 'guide' the chain towards the high-probability region of $\pi(x)$; one proposal that can be chosen for that purpose is

$$q(x'|x) = \phi(x'; g(x), \sigma_q^2)$$

where the mean for the proposal $g(x)$ is constructed by using the gradient of the

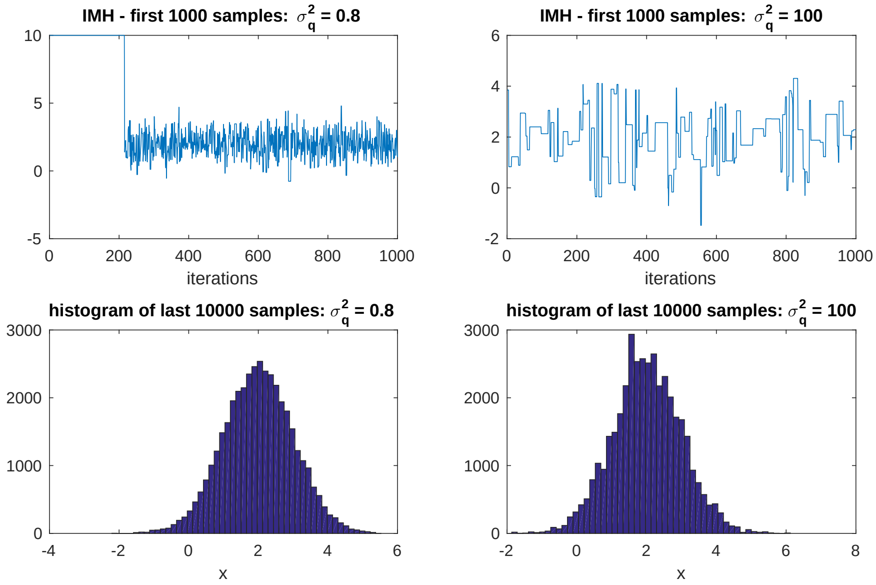


Figure 5.6: Independence MH for $\pi(x) = \phi(x; 2, 1)$.

logarithm of the target density,

$$g(x) = x + \gamma \frac{\partial \log \pi(x)}{\partial x}.$$

Here, γ is a step-size parameter that needs to be adjusted. For $\pi(x) = \phi(x; \mu, \sigma^2)$, $g(x) = x - \frac{\gamma}{\sigma^2}(x - \mu)$. The acceptance ratio for this choice of proposal becomes

$$r(x, x') = e^{-\frac{1}{2\sigma^2}\left[(x'-\mu)^2-(x-\mu)^2\right]+\frac{1}{2\sigma_q^2}\left[\left(x'-x+\frac{\gamma}{\sigma^2}(x-\mu)\right)^2-\left(x-x'+\frac{\gamma}{\sigma^2}(x'-\mu)\right)^2\right]}$$

See Figure 5.7 for examples of MH with this choice.

Example 5.13 **(Normal distribution with unknown mean and variance).** *We have observations $Y_1, \ldots, Y_n \sim \mathcal{N}(z, s)$ and z and s are unknown. The parameters $x = (z, s)$ are* a priori *independent with $z \sim \mathcal{N}(m, \kappa^2)$ and $s \sim \mathcal{IG}(\alpha, \beta)$, so that the prior density is*

$$p(x) = p(z)p(s) = \frac{1}{\sqrt{2\pi\kappa^2}} e^{-\frac{1}{2\kappa^2}(z-m)^2} \frac{\beta^\alpha}{\Gamma(\alpha)} s^{-\alpha-1} e^{-\frac{\beta}{s}}$$

Given the data $Y_{1:n} = y_{1:n}$, we want to run the MH algorithm to sample from the posterior distribution $\pi(x) = p(x|y_{1:n})$, which is given by

$$\pi(x) = p(x|y_{1:n}) \propto p(x)p(y_{1:n}|x) = p(z)p(s)\prod_{i=1}^{n} \phi(y_i; z, s)$$

For this problem, $\pi(x)$ indeed lacks a well-known form, so it is justified to use a Monte Carlo method for it.

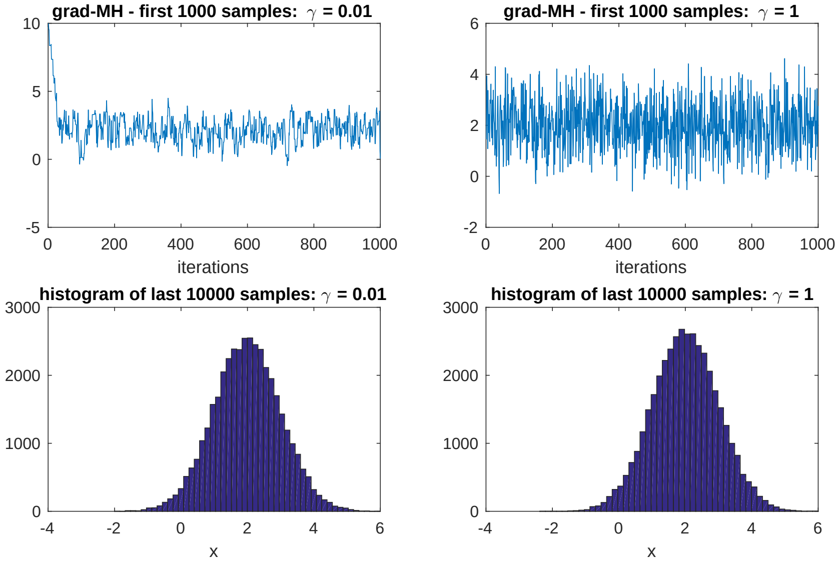


Figure 5.7: Gradient-guided MH for $\pi(x) = \phi(x; 2, 1)$.

To run the MH algorithm, we need a proposal distribution for proposing $x' = (z', s')$. In this example, given $x = (z, s)$, we decide to propose $z' \sim \mathcal{N}(z, \sigma_q^2)$ and $s' \sim \mathcal{IG}(\alpha, \beta)$, i.e. we use a random walk for the mean component and the prior distribution for the variance parameter. With this choice, the proposal density becomes

$$\begin{aligned} q(x'|x) &= \phi(z'; z, \sigma_q^2)p(s') \\ &= \phi(z'; z, \sigma_q^2)\frac{\beta^\alpha}{\Gamma(\alpha)}(s')^{-\alpha-1}e^{-\frac{\beta}{s'}} \end{aligned}$$

The acceptance ratio in this case is

$$\begin{aligned} r(x, x') &= \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)} \\ &= \frac{p(z')p(s')\left[\prod_{i=1}^n p(y_i|z', s')\right]\phi(z; z', \sigma_q^2)p(s)}{p(z)p(s)\left[\prod_{i=1}^n p(y_i|z, s)\right]\phi(z'; z, \sigma_q^2)p(s')} \\ &= \frac{\phi(z'; m, \kappa^2)\prod_{i=1}^n \phi(y_i; z', s')}{\phi(z; m, \kappa^2)\prod_{i=1}^n \phi(y_i; z, s)} \end{aligned}$$

See Figure 5.8 for results obtained from this MH algorithm.

Example 5.14 (A changepoint model). *In this example, we consider a changepoint model. In this model, at each time t we observe the count of an event Y_t. All the counts up to an unknown time τ come from the same distribution after which the distribution changes. We assume that the changepoint τ is uniformly distributed over $\{1, \ldots, n\}$ where n is the number of time steps. The two different distributional regimes up to τ and after τ*

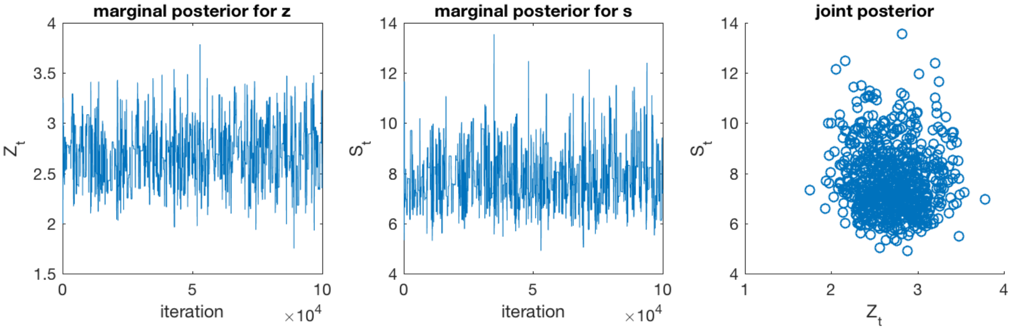


Figure 5.8: MH for parameters of $\mathcal{N}(z, s)$. $\sigma_q^2 = 1$, $\alpha = 5$, $\beta = 10$, $m = 0$, $\kappa^2 = 100$.

are indicated by the random variables λ_i, $i = 1, 2$, *which are* a priori *assumed to follow a Gamma distribution*

$$\lambda_i \sim \Gamma(\alpha, \beta), \quad i = 1, 2.$$

Under regime i, *the counts are assumed to be identically Poisson distributed*

$$Y_t \sim \begin{cases} \mathcal{PO}(\lambda_1), & 1 \leq t \leq \tau \\ \mathcal{PO}(\lambda_2), & \tau < t \leq n. \end{cases}$$

A typical draw from this model is shown in Figure 5.9. The inferential goal is, given $Y_{1:n} = y_{1:n}$, *to sample from the posterior distribution of the changepoint location* τ *and the intensities* λ_1, λ_2 *given the count data, i.e., letting* $x = (\tau, \lambda_1, \lambda_2)$, *the target distribution is* $\pi(x) = p(\tau, \lambda_1, \lambda_2 | y_{1:n})$ *which is given by*

$$\begin{aligned} p(\tau, \lambda_1, \lambda_2 | y_{1:n}) &\propto p(\tau, \lambda_1, \lambda_2, y_{1:n}) \\ &= p(\tau, \lambda_1, \lambda_2) p(y_{1:n} | \tau, \lambda_1, \lambda_2) \\ &= p(\tau) p(\lambda_1) p(\lambda_2) p(y_{1:n} | \tau, \lambda_1, \lambda_2) \\ &= \frac{1}{n} \frac{\beta^\alpha \lambda_1^{\alpha-1} e^{-\beta\lambda_1}}{\Gamma(\alpha)} \frac{\beta^\alpha \lambda_2^{\alpha-1} e^{-\beta\lambda_2}}{\Gamma(\alpha)} \prod_{t=1}^{\tau} \frac{e^{-\lambda_1} \lambda_1^{y_t}}{y_t!} \prod_{t=\tau+1}^{n} \frac{e^{-\lambda_2} \lambda_2^{y_t}}{y_t!} \end{aligned} \tag{5.4}$$

Two choices for the proposal will be considered. Let $x' = (\tau', \lambda_1', \lambda_2')$.

- *The first one is to use an independent proposal distribution, which is the prior distribution for* x

  $$q(x'|x) = q(x') = p(x') = p(\tau', \lambda_1', \lambda_2').$$

  This leads to the acceptance ratio being the ratio of the likelihoods

  $$r(x, x') = \frac{p(y_{1:n} | \tau', \lambda_1', \lambda_2')}{p(y_{1:n} | \tau, \lambda_1, \lambda_2)}$$

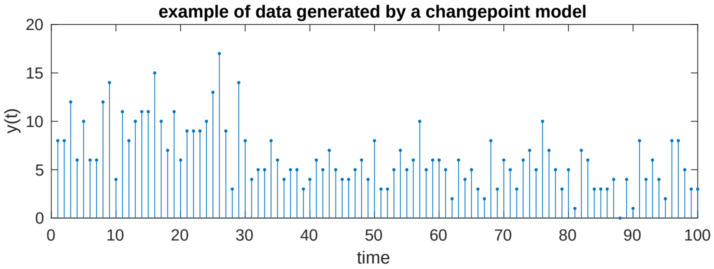


Figure 5.9: An example data sequence of length $n = 100$ generated from the Poisson changepoint model with parameters $\tau = 30$, $\lambda_1 = 10$ and $\lambda_2 = 5$.

- *The second choice is a symmetric proposal,*

$$q(x'|x) = \left[\frac{1}{2}\mathbb{I}_{\tau+1}(\tau') + \frac{1}{2}\mathbb{I}_{\tau-1}(\tau')\right]\phi(\lambda_1';\lambda_1,\sigma_\lambda^2)\phi(\lambda_2';\lambda_2,\sigma_\lambda^2).$$

The first factor involving τ indicates that we propose either $\tau' = \tau + 1$ or $\tau' = \tau - 1$ both with probability a half. Since $q(x'|x) = q(x|x')$, the acceptance ratio reduces to the ratio of the posteriors

$$\begin{aligned} r(x, x') =& \frac{p(\tau', \lambda_1', \lambda_2'|y_{1:n})}{p(\tau, \lambda_1, \lambda_2|y_{1:n})} \\ =& e^{-(\tau+\beta)(\lambda_1'-\lambda_1)} e^{-(n-\tau+\beta)(\lambda_2'-\lambda_2)} \left(\frac{\lambda_1'}{\lambda_1}\right)^{\alpha-1+\sum_{t=1}^{\tau} y_t} \left(\frac{\lambda_2'}{\lambda_2}\right)^{\alpha-1+\sum_{t=\tau+1}^{n} y_t} \\ &\times \begin{cases} e^{-\lambda_1'+\lambda_2'}\left(\frac{\lambda_1'}{\lambda_2'}\right)^{y_{\tau+1}}, & \tau' = \tau + 1, \\ e^{-\lambda_2'+\lambda_1'}\left(\frac{\lambda_2'}{\lambda_1'}\right)^{y_\tau}, & \tau' = \tau - 1. \end{cases} \end{aligned}$$

Figure 5.10 illustrates the results obtained from the two algorithms. The initial value for τ is taken $\lfloor n/2 \rfloor$ and for λ_1 and λ_2 we start from the mean of $y_{1:n}$. As we can see, the symmetric proposal algorithm is able to explore the posterior distribution much more efficiently. This is because the proposal distribution in independence MH, which is chosen as the prior distribution, does not take neither the posterior distribution (hence the data) nor the previous sample into account, and as a result it has a large rejection rate. The independence sampler would become even poorer if n were larger so that the posterior would be more concentrated in contrast to the ignorance of the prior distribution.

Example 5.15 (MCMC for source localisation). *Consider the source localisation scenario in Question 3 of Exercises in Chapter 3. From the likelihood and the prior in (3.8) and (3.9), the posterior distribution of the unknown position is*

$$p(x|y) \propto \phi(x(1); 0, \sigma_x^2)\phi(x(2); 0, \sigma_x^2) \prod_{i=1}^{3} \phi(y_i; r_i, \sigma_y^2) \qquad (5.5)$$

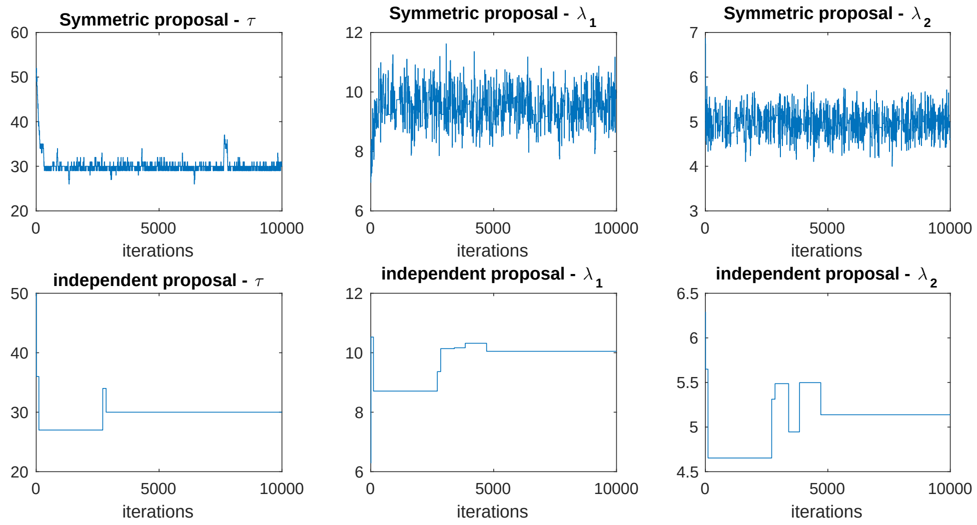


Figure 5.10: MH for parameters of the Poisson changepoint model

Due to the non-linearity in the $r_i = ||x - s_i|| = [(x(1) - s_i(1))^2 + (x(2) - s_i(2))^2]^{1/2}$, $i = 1, 2, 3$, $p(x|y)$ *does not admit a known distribution. We use the MH algorithm to generate approximate samples from* $p(x|y)$. *We use a symmetric random walk proposal distribution with* $q(x'|x) = \phi(x'; x, \sigma_q^2 I_2)$, *so that* $q(x'|x) = q(x|x')$. *The resulting acceptance rate*

$$\begin{aligned} r(x, x') &= \frac{p(x'|y)q(x|x')}{p(x|y)q(x'|x)} \\ &= \frac{p(x'|y)}{p(x|y)} \\ &= \frac{\phi(x'(1); 0, \sigma_x^2)\phi(x'(2); 0, \sigma_x^2) \prod_{i=1}^{3} \phi(y_i; r_i', \sigma_y^2)}{\phi(x(1); 0, \sigma_x^2)\phi(x(2); 0, \sigma_x^2) \prod_{i=1}^{3} \phi(y_i; r_i, \sigma_y^2)} \end{aligned}$$

where $r_i' = ||x' - s_i||$, $i = 1, 2, 3$ *is the distance between the proposed value* x' *and the location i'th source* s_i. *Figure 5.11 shows the samples and their histograms obtained from* 10000 *iterations of the MH algorithm. The chain was started from* $X_1 = (5, 5)$ *and its convergence to the posterior distribution is illustrated in the right pane of the figure where we see the first a few samples of the chain traveling to the high probability region of the posterior distribution.*

5.4 Gibbs sampling

The *Gibbs sampler* (Geman and Geman, 1984; Gelfand and Smith, 1990) is one of the most popular MCMC methods, which can be used when X has more than one dimension. If X has $d > 1$ components (of possibly different dimensions) such that $X = (X_1, \ldots, X_d)$, and

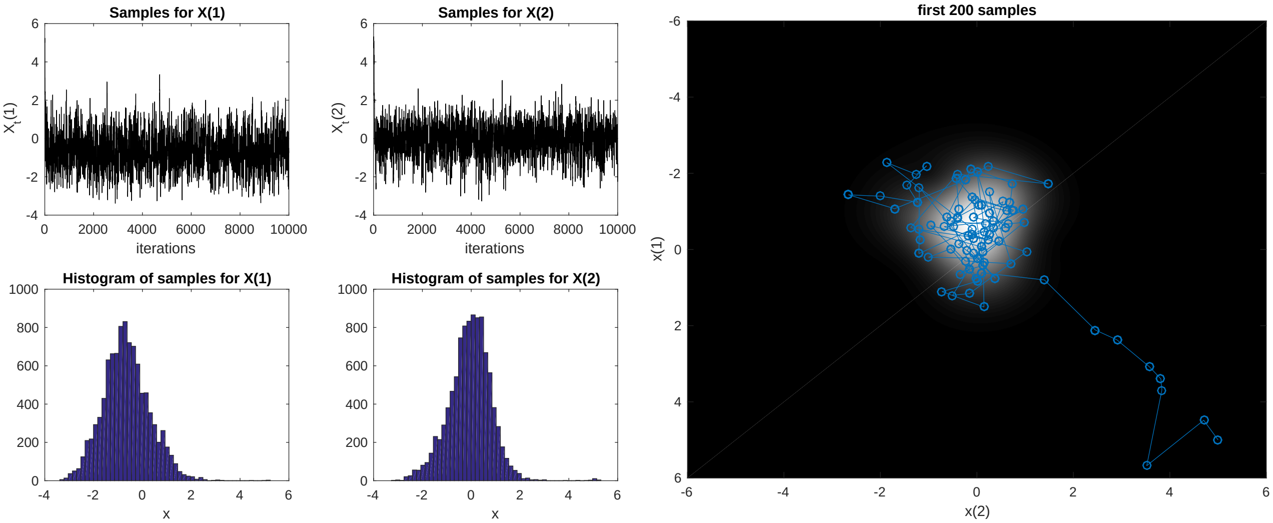


Figure 5.11: MH for the source localisation problem.

one can sample from each of the *full conditional distributions* $\pi_k\left(\cdot|X_{1:k-1}, X_{k+1:d}\right)$, then the Gibbs sampler produces a Markov chain by updating one component at a time using π_k's. One cycle of the Gibbs sampler successively samples from the conditional distributions $\pi_1, \ldots, \pi_d$ by conditioning on the most recent samples.

Algorithm 5.2: The Gibbs sampler:

```
1 Begin with some $X_1 \in \mathcal{X}$.
2 for $n = 2, 3, \ldots$ do
3     for $k = 1, \ldots, d$ do
4         $$X_{n,k} \sim \pi_k(\cdot|X_{n,1:k-1}, X_{n-1,k+1:d}).$$
```

For an $x \in \mathcal{X}$, let $x_{-k} = (x_{1:k-1}, x_{k+1:d})$ for $k = 1, \ldots, d$ denotes the components of x excluding x_k, and let us permit ourselves to write $x = (x_k, x_{-k})$. The corresponding MCMC kernel of the Gibbs sampler can be written as $M = M_1 M_2 \ldots M_d$, where each transition kernel M_k for $k = 1, \ldots, d$ can be written as

$$M_k(y|x) = \pi_k(y_k|x_{-k})\delta_{x_{-k}}(y_{-k})$$

where $y = (y_1, \ldots, y_d)$. The justification of the transitional kernel comes from the reversibility of each M_k with respect to π, which can be verified from the detailed balance condition as follows.

$$\begin{aligned}\pi(x)M_k(y|x) &= \pi(x)\pi_k(y_k|x_{-k})\delta_{x_{-k}}(y_{-k}) \\ &= \pi(x_{-k})\pi_k(x_k|x_{-k})\pi_k(y_k|x_{-k})\delta_{x_{-k}}(y_{-k}) \\ &= \pi(y_{-k})\pi_k(y_k|y_{-k})\pi_k(x_k|y_{-k})\delta_{y_{-k}}(x_{-k}) \\ &= \pi(y)M_k(x|y), \end{aligned} \tag{5.6}$$

where the third line follows the second since $\delta_{x_{-k}}(y_{-k})$ allows the interchange of x_{-k} and y_{-k}. Therefore, the detailed balance condition for M_k is satisfied with π and $\pi M_k = \pi$. If we apply $M_1, \dots, M_k$ sequentially, we get

$$\pi M = \pi M_1 \dots M_d = (\pi M_1)M_2 \dots M_d = \pi M_2 \dots M_d = \dots = \pi,$$

so π is indeed the invariant distribution for the Gibbs sampler.

Gibbs sampling as a special Metropolis-Hastings algorithm: An insightful interpretation of (5.6) is that each step of a cycle of the Gibbs sampler is a Metropolis-Hastings move whose MCMC kernel is equal to its proposal kernel which results in the acceptance probability being 1 uniformly. Indeed, if the k'th component of X is to be updated with $Q_k = M_k$, i.e. if we propose the new value y as

$$q_k(y|x) = M_k(y|x) = \pi_k(y_k|x_{-k})\delta_{x_{-k}}(y_{-k}),$$

the acceptance ratio $\alpha_k(x, y)$ for this move is

$$\alpha_k(x, y) = \min\left\{1, \frac{\pi(y)q_k(x|y)}{\pi(x)q_k(y|x)}\right\} = \min\left\{1, \frac{\pi(y)M_k(x|y)}{\pi(x)M_k(y|x)}\right\} = 1$$

as shown in (5.6).

Reversibility of each M_k with respect to π does not suffice to establish proper convergence of the Gibbs sampler, as none of the individual steps produces a irreducible chain. Only the combination of the d moves in the complete cycle has a chance of producing a ϕ-irreducible chain. We refer to Roberts and Smith (1994) some simple conditions for convergence of the classical Gibbs sampler. Note, also, that M is not reversible either, although this is not a necessary condition for convergence. A way of guaranteeing both ϕ-irreducibility and reversibility is to use a mixture of kernels

$$M_\beta = \sum_{k=1}^{d} \beta_k M_k, \quad \beta_k > 0, \quad k = 1, \dots, d, \quad \sum_{k=1}^{d} \beta_k = 1.$$

provided that at least one M_k is irreducible and aperiodic. This choice of kernel leads to the *random scan Gibbs sampler algorithm*. We refer to Tierney (1994), Robert and Casella (2004), and Roberts and Tweedie (1996) for more detailed convergence results pertaining to these variants of the Gibbs sampler.

Example 5.16. *Suppose we wish to sample from a bivariate normal distribution, where*

$$\pi(x) = \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp\left\{ -\frac{x_1^2 + x_2^2 - 2\rho x_1 x_2}{2(1-\rho^2)} \right\}, \quad \rho \in (-1, 1).$$

The full conditionals are

$$\pi(x_1|x_2) \propto \pi(x_1, x_2) \propto \exp\left\{ -\frac{(x_1 - \rho x_2)^2}{2(1-\rho^2)} \right\}$$

therefore $\pi(x_1|x_2) = \phi(x_1; \rho x_2, (1-\rho^2))$ and $X_1|X_2 = x_2 \sim \mathcal{N}(\rho x_2, (1-\rho)^2)$. Similarly, we have $X_2|X_1 = x_1 \sim \mathcal{N}(\rho x_1, (1-\rho)^2)$. So, the iteration $t \geq 2$ of the Gibbs sampling algorithm for this $\pi(x)$ is

- *Sample $X_{t,1} \sim \mathcal{N}(\rho X_{t-1,2}, (1-\rho)^2)$,*
- *Sample $X_{t,2} \sim \mathcal{N}(\rho X_{t,1}, (1-\rho)^2)$.*

Example 5.17 (ex: Normal distribution with unknown mean and variance). *Let us get back to the problem in Example 5.13 where we want to estimate the mean and the variance of the normal distributions $\mathcal{N}(z, s)$ given samples $y_1, \ldots, y_n$ generated from it. Let use the same prior distributions for z and s, namely $z \sim \mathcal{N}(m, \kappa^2)$ and $s \sim \mathcal{IG}(\alpha, \beta)$. Note that these are the conjugate priors for those parameters; and when one of the parameters is given, the posterior distribution of the other one has a known form. Indeed, in Examples 4.5 and 4.6, we derived these full conditional distributions. Example 4.5 can be revisited (but this time with a non-zero prior mean m) to see that*

$$Z|s, y_{1:n} \sim \mathcal{N}(\mu_{z|s,y}, \sigma^2_{z|s,y})$$

where

$$\sigma^2_{z|s,y} = \left(\frac{1}{\kappa^2} + \frac{n}{s} \right)^{-1}, \quad \mu_{z|s,y} = \left(\frac{1}{\kappa^2} + \frac{n}{s} \right)^{-1} \left(\frac{1}{s} \sum_{i=1}^{n} y_i + \frac{m}{\kappa^2} \right)$$

and from Example 4.6 we can deduce that

$$S|z, y_{1:n} = \mathcal{IG}(\alpha_{s|z,y}, \beta_{s|z,y})$$

where

$$\alpha_{s|z,y} = \alpha + \frac{n}{2}, \quad \beta_{s|z,y} = \beta + \frac{1}{2} \sum_{i=1}^{n} (y_i - z)^2.$$

Therefore, Gibbs sampling for Z, S given $Y_{1:n} = y_{1:n}$ is

- *Sample $Z_t \sim \mathcal{N}\left(\left(\frac{1}{\kappa^2} + \frac{n}{S_{t-1}} \right)^{-1} \left(\frac{1}{S_{t-1}} \sum_{i=1}^{n} y_i + \frac{m}{\kappa^2} \right), \left(\frac{1}{\kappa^2} + \frac{n}{S_{t-1}} \right)^{-1} \right)$*
- *Sample $S_t \sim \mathcal{IG}\left(\alpha + \frac{n}{2}, \beta + \frac{1}{2} \sum_{i=1}^{n} (y_i - Z_t)^2 \right)$.*

Data augmentation: Data augmentation is an application of the Gibbs sampler. It is useful if

1. there is missing data, and/or
2. the likelihood is intractable (hard to compute or does not admit conjugacy, etc), but given some additional unobserved (real or fictitious) data it would be tractable.

Let y_{obs} denote the observed data and y_{mis} the missing data (sometimes y_{mis} is called a latent variable).We suppose we can easily sample x from the posterior given the augmented data $(y_{\text{obs}}, y_{\text{mis}})$. Also, that we can sample y_{mis}, conditional on y_{obs} and X (this only involves the sampling distributions). Then we can use the Gibbs sampler of the pair (x, y_{mis}). Then we perform Monte Carlo marginalisation: If in the resulting joint distribution for x, y_{mis} given y_{obs} we simply ignore y_{mis}, we shall have our sample from the posterior of x given y_{obs} alone.

Example 5.18 (Genetic linkage). *Genetic linkage in an animal can be allocated to one of four categories, coded* 1,2, 3, *and* 4, *having respective probabilities*

$$(1/2 + \theta/4, (1-\theta)/4, (1-\theta)/4, \theta/4)$$

where θ is an unknown parameter in $(0,1)$. For a sample of 197 animals, the (multinomial) counts of those falling in the 4 categories are represented by random variables $Y = (Y_1, Y_2, Y_3, Y_4)$, with observed values $y = (y_1, y_2, y_3, y_4) = (125, 18, 20, 34)$. Suppose we place a Beta(α, β) *prior on θ. Then,*

$$\pi(\theta) = p(\theta|y) \propto \underbrace{\left(\frac{1}{2} + \frac{\theta}{4}\right)^{125} \left(\frac{1-\theta}{4}\right)^{18+20} \left(\frac{\theta}{4}\right)^{34}}_{\text{Multinomial likelihood}} \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

$$\propto (2+\theta)^{125}(1-\theta)^{38+\beta-1}\theta^{34+\alpha-1} \tag{5.7}$$

How can we sample from this? We can use a rejection sampler (probably with a very high rejection probability) or MH for this posterior distribution; in this example we seek for a suitable Gibbs sampler. Note that the problematic part in (5.7) is the first one; should it be like one of the others, the posterior would lend itself into a Beta distribution.

Suppose we divide category 1, with total probability $1/2 + \theta/4$, into two latent subcategories, a and b, with respective probabilities $\theta/4$ and $1/2$. We regard the number of animals Z falling in subcategory a as missing data. If, as well as the observed data y, we are given $Z = z$, we are in the situation of having observed counts $(z, 125 - z, 18, 20, 34)$ from a multinomial distribution with probabilities $(\theta/4, 1/2, (1-\theta)/4, (1-\theta)/4, \theta/4)$. The resulting joint distribution is

$$p(\theta, z|y) \propto p(\theta, z, y) = \left(\frac{1}{2}\right)^{125-z} \left(\frac{1-\theta}{4}\right)^{18+20} \left(\frac{\theta}{4}\right)^{34+z} \theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{5.8}$$

This easily leads to the posterior distribution

$$\theta | z, y \sim \text{Beta}(z + 34 + \alpha, 38 + \beta). \tag{5.9}$$

Also, simple properties of the multinomial distribution yield

$$Z | \theta, y \sim \text{Binom}\left(125, \frac{\theta/4}{1/2 + \theta/4}\right) \tag{5.10}$$

So we can now apply Gibbs sampling, cycling between updates given by (5.9) and (5.10).

Example 5.19 (A changepoint model, ctd.). *Consider the changepoint problem in Example 5.14, with the same likelihood and priors. It is possible to run Gibbs sampling algorithm for $\tau, \lambda_1, \lambda_2$. Observing (5.4), where the full posterior distribution is written as proportional to the full joint distribution*

$$p(\tau, \lambda_1, \lambda_2, y_{1:n}) = \frac{1}{n} \frac{\beta^\alpha \lambda_1^{\alpha-1} e^{-\beta\lambda_1}}{\Gamma(\alpha)} \frac{\beta^\alpha \lambda_2^{\alpha-1} e^{-\beta\lambda_2}}{\Gamma(\alpha)} \prod_{t=1}^{\tau} \frac{e^{-\lambda_1} \lambda_1^{y_t}}{y_t!} \prod_{t=\tau+1}^{n} \frac{e^{-\lambda_2} \lambda_2^{y_t}}{y_t!},$$

from which we can derive all the full conditionals

$$\begin{aligned}
\lambda_1 | \tau, \lambda_2, y_{1:n} &\sim \Gamma\left(\alpha + \sum_{t=1}^{\tau} y_t, \beta + \tau\right) \\
\lambda_2 | \tau, \lambda_1, y_{1:n} &\sim \Gamma\left(\alpha + \sum_{t=\tau+1}^{n} y_t, \beta + n - \tau\right) \\
\tau | \lambda_1, \lambda_2, y_{1:n} &\sim \text{Categorical}(a_1, \dots, a_n)
\end{aligned}$$

where the probabilities in the Categorical distribution (which is simply the discrete distribution with probabilities $a_1, \dots, a_n$, the generalisation of the Bernoulli distribution to the case of multiple (here, n) outcomes) are

$$a_i = \frac{e^{-i\lambda_1} \lambda_1^{\sum_{t=1}^{i} y_t} e^{-(n-i)\lambda_2} \lambda_2^{\sum_{t=i+1}^{n} y_t}}{\sum_{j=1}^{n} \left[e^{-j\lambda_1} \lambda_1^{\sum_{t=1}^{j} y_t} e^{-(n-j)\lambda_2} \lambda_2^{\sum_{t=j+1}^{n} y_t} \right]}$$

5.4.1 Metropolis within Gibbs

Having attractive computational properties, the Gibbs sampler is widely used. The requirement for easy-to-sample conditional distributions is the main restriction for the Gibbs sampler. Fortunately, though, replacing an exact simulation $X_k \sim \pi_k(\cdot | X_{n-1,1:k-1}, X_{n-1,k+1:d})$ by a Metropolis-Hastings step in a general MCMC algorithm does not violate its validity as long as the Metropolis-Hastings step has the correct invariant distribution. The most natural alternative to the Gibbs move in step k where sampling from the full conditional distribution $\pi_k(\cdot | x_{-k})$ is not directly feasible is to use Metropolis-Hastings move that updates x_k by using a Metropolis-Hastings kernel that targets $\pi_k(\cdot | x_{-k})$ (Tierney, 1994).

Exercises

1. Consider the toy example in Section 5.3.1 for the MH algorithm for sampling from the normal distribution $\mathcal{N}(\mu, \sigma^2)$.

   - Modify the code so that it stores the acceptance probability at each iteration in a vector and returns the vector as one of the outputs. In the next part of the exercise, you will use the last $T - t_b$ samples of the vector to find an estimate of the overall expected acceptance probability

     $$\alpha(\sigma_q) = \int \alpha(x, x')\pi(x)q_{\sigma_q}(x'|x)dxdx'.$$

     where $q_{\sigma_q}(x'|x) = \phi(x'; x, \sigma_q^2)$.

   - Choose $\mu = 0$, $\sigma^2 = 1$, and $\sigma_q = 1$. Run the MH algorithm (provided in SUCourse) 100 times, each with $T = 10000$ iterations with the symmetric proposal with various values for the proposal variance. For each run $i = 1, \ldots, 100$, use the samples $X_1^{(i)}, \ldots, X_T^{(i)}$ to calculate the mean estimate

     $$\mu^{(i)}(\sigma_q) = \frac{1}{T - t_b} \sum_{t=t_b+1}^{T} X_t^{(i)}$$

     where $t_b = 1000$ is the burn-in time up to when you ignore the samples generated by the algorithm. Similarly, calculate an estimate $\alpha^{(i)}(\sigma_q)$ of $\alpha(\sigma_q)$ in a similar way using the last $T - t_b$ samples.

   - Report the sample variance of $\mu^{(i)}(\sigma_q)$'s: This is approximately the variance of the mean estimate of the MH algorithm that uses $T - t_b$ samples. We wish this variance to be as small as possible. Also, report the average of $\alpha^{(i)}(\sigma_q)$'s.

   - Repeat above for $\sigma_q = 0.1, 0.2, \ldots, 9.9, 10$, and generate two plots: (i) sample variance of $\mu^{(i)}(\sigma_q)$'s vs σ_q and (ii) average of $\alpha^{(i)}(\sigma_q)$'s vs σ_q. From the first plot, suggest the (approximately) optimum value for σ_q and report the estimate of $\alpha(\sigma_q)$ for that σ_q.

2. Design and implement a symmetric random walk MH algorithm and the Gibbs sampling algorithm for the genetic linkage problem in Example 5.18 with hyperparameters $\alpha = \beta = 2$.

3. Implement the Gibbs sampler in Example 5.17 with $n = 100$ and hyperparameters $\alpha = 5$ and $\beta = 10$.

4. Consider the changepoint problem in Example 5.14.

   - Download `UK_coal_mining_disaster_days.mat` from SUCourse. The data consists of the day numbers of coal mining disasters between 1851 and 1962, where the first day is the start of the 1851. It is suspected that, due to a policy change, the accident rate over the years is a piecewise constant with a single changepoint time around the time of the policy change.

- From the data, create another data vector of length 112, where the i'th element contains the number of disasters in year i (starting from 1851). Note that some years are 366 days!
- Implement the MH algorithm (given in SUCourse) for the changepoint model given the data that you created. Take the priors for τ, λ_1 and λ_2 the same as in Example 5.14, i.e. with hyperparameters $\alpha = 10$ and $\beta = 1$. You can use the symmetric random walk proposal for the parameters.
- Implement Gibbs sampling algorithm for the same model given the same data using the same priors. All the derivations you need are in Example 5.19.

5. Suppose we observe a noisy sinusoid with with unknown amplitude a, angular frequency ω, phase z, and noise variance σ_y^2 for n steps. Letting $x = (a, \omega, z, \sigma_y^2)$,

$$Y|x \sim \mathcal{N}(y_t; a\sin(\omega t + z), \sigma_y^2), \quad t = 1, \ldots, n.$$

The unknown parameters are a priori independent with $a \sim \mathcal{N}(0, \sigma_a^2)$, $\omega \sim \Gamma(\alpha, \beta)$, $z \sim \text{Unif}(0, 2\pi)$, $\sigma_y^2 \sim \mathcal{IG}(\alpha, 1/\beta)$.

- Write down the likelihood of $p(y_{1:n}|x)$ and the joint density $p(x, y_{1:n})$.
- Download the data file `sinusoid_data.mat` from SUCourse; the observations in the file are your data $y_{1:n}$. Use hyperparameters $\sigma_a^2 = 100$, $\alpha = \beta = 0.01$ and design and implement an MH algorithm for generating samples from the posterior distribution $\pi(x) = p(x|y_{1:n})$.
- **Bonus - worth 50% of the base mark:** This time, design and implement a MH within Gibbs algorithm where in each loop contains four steps in each of which you update one component only, fixing the others, using an MH kernel that targets the full conditionals. This is an example where you can still update the components one by one even if the full conditional distributions are not easy to sample from.

Chapter 6

Sequential Monte Carlo

Summary: *This chapter contains a brief and limited review of sequential Monte Carlo methods, another large family of Monte Carlo methods that are used for many applications including sequential inference, sampling from complex distributions, rare event analysis, density estimation, optimisation, etc. In this chapter we will introduce two main methods, sequential importance sampling, and sequential importance sampling-resampling, in a generic setting.*

6.1 Introduction

Let $\{X_n\}_{n\geq 1}$ be a sequence of random variables where each X_n takes values at some space $\mathcal{X}_n$. Define the sequence of distributions $\{\pi_n\}_{n\geq 1}$ where π_n is defined on $\mathcal{X}^n$. Also, let $\{\varphi_n\}_{n\geq 1}$ be a sequence of functions where $\varphi_n : \mathcal{X}^n \to \mathbb{R}$ is a real-valued function on $\mathcal{X}^n$.[1] We are interested in sequential inference, i.e. approximating the following integrals sequentially in n

$$\pi_n(\varphi_n) = \mathbb{E}_{\pi_n}\left[\varphi_n(X_{1:n})\right] = \int \pi_n(x_{1:n})\varphi(x_{1:n})dx_{1:n}, \quad n = 1, 2, \ldots$$

Despite their versatility and success, it might be impractical to apply MCMC algorithms to sequential inference problems. This chapter discusses *sequential Monte Carlo* (SMC) methods, that can provide with approximation tools for a sequence of varying distributions. Good tutorials on the subject are available, see for example Doucet et al. (2000b) for and Doucet et al. (2001) for a book length review. Also, Robert and Casella (2004) and Cappé et al. (2005) contain detailed summaries. Finally, the book Del Moral (2004) contains a more theoretical work on the subject in a more general framework, namely Feynman-Kac formulae.

6.2 Sequential importance sampling

The first method which is usually considered a sequential Monte Carlo (SMC) method is *sequential importance sampling* (SIS), which is a sequential version of the importance

[1] In a more general setting X_n takes values at some space $\mathcal{X}_n$ which may not be the same set for all n. Then the sequence of distributions $\{\pi_n\}_{n\geq 1}$ would be on $\mathscr{X}_n = \prod_{i=1}^{n} \mathcal{X}_i$ and we would have $\varphi_n : \mathscr{X}_n \to \mathbb{R}$.

sampling. First use of SIS can be recognised in works back in 1960s and 1970s such as Mayne (1966); Handschin and Mayne (1969); Handschin (1970), see Doucet et al. (2000b) for a general formulation of the method for Bayesian filtering.

Naive approach: Consider the naive importance sampling approach to the sequential problem where we have a sequence of importance densities $\{q_n(x_{1:n})\}_{n\geq 1}$ where each q_n is defined on $\mathcal{X}^n$ such that

$$w_n(x_{1:n}) = \frac{\pi_n(x_{1:n})}{q_n(x_{1:n})}.$$

It is obvious that we can approximate $\pi_n(\varphi_n)$ by generating independent samples from q_n at each n and exploiting the relation

$$\pi_n(\varphi_n) = \mathbb{E}_{q_n}\left[w_n(X_{1:n})\varphi_n(X_{1:n})\right].$$

This approach would require the design of a separate $q_n(x_{1:n})$ and sampling the whole path $X_{1:n}$ at each n, which is obviously inefficient.

Sequential design of the importance density: An efficient alternative to the naive approach is SIS which can be used when it is possible to choose $q_n(x_{1:n})$ to have the form

$$q_n(x_{1:n}) = q(x_1)\prod_{t=2}^{n} q(x_t|x_{1:t-1}), \tag{6.1}$$

where $q(x_1)$ is some initial density that is easy to sample from and $q(x_t|x_{1:t-1})$ are conditional densities which we design so that it is possible to sample from $q(\cdot|x_{1:t-1})$ for any $x_{1:t-1}$ and $t \geq 1$. This selection of q_n leads to the following useful recursion on the importance weights

$$\begin{aligned} w_n(x_{1:n}) &= \frac{\pi_n(x_{1:n})}{q_n(x_{1:n})} \\ &= \frac{\pi_n(x_{1:n})}{q_{n-1}(x_{1:n-1})q(x_n|x_{1:n-1})}\frac{\pi_{n-1}(x_{1:n-1})}{\pi_{n-1}(x_{1:n-1})} \\ &= w_{n-1}(x_{1:n-1})\frac{\pi_n(x_{1:n})}{\pi_{n-1}(x_{1:n-1})q(x_n|x_{1:n-1})}. \end{aligned} \tag{6.2}$$

We remark that the sequence of distributions are usually known up to a normalising constant as

$$\pi_n(x_{1:n}) = \frac{\hat{\pi}_n(x_{1:n})}{Z_{\pi_n}},$$

where we know $\hat{\pi}_n(x_{1:n})$ for any $x_{1:n}$ but not Z_{π_n}. Hence, from now on we will only consider self-normalised importance sampling where π_{n-1} and π_n are replaced by $\hat{\pi}_{n-1}$ and $\hat{\pi}_n$ in calculation of (and the recursion for) $w_n(x_{1:n})$ in (6.2).

Approximation to π_n: As long as self-normalised importance sampling is concerned, given the samples and their weights, it is practical to define the weighted empirical distribution

$$\pi_n^N(x_{1:n}) := \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{1:n}^{(i)}}(x_{1:n}), \tag{6.3}$$

as an approximation to π_n, where $W_n^{(i)}$, $i = 1, \dots, N$ are the self-normalised importance weights

$$W_n^{(i)} = \frac{w_n(X_{1:n}^{(i)})}{\sum_{i=1}^{N} w_n(X_{1:n}^{(i)})}. \tag{6.4}$$

This is another way of viewing the self-normalised importance sampling: The self-normalised importance sampling approximation of the desired expectation $\pi_n(\varphi_n)$ is actually the exact expectation of φ with respect to π_n^N. This expectation is given by

$$\pi_n^N(\varphi_n) = \sum_{i=1}^{N} W_n^{(i)} \varphi_n(X_{1:n}^{(i)}).$$

Note that this is indeed the same as the self-normalised importance sampling estimate, see (3.7) for example.

The SIS algorithm: In many applications of (6.2), the importance density is designed in such a way that the ratio

$$\frac{\pi_n(x_{1:n})}{\pi_{n-1}(x_{1:n-1})q(x_n|x_{1:n-1})}$$

is easy to calculate (at least up to a proportionality constant if we use the unnormalised densities). For example, this may be due to the design of $q(x_n|x_{1:n-1})$'s in such a way that the ratio depends only on x_{n-1} and x_n. Hence, one can exploit this recursion by sampling only X_n from $q(\cdot|x_{1:n-1})$ at time n and updating the weights with a small effort.

More explicitly, assume a set of $N \geq 1$ samples, termed as *particles*, $X_{1:n-1}^{(i)}$ with weights $w_{n-1}(X_{1:n-1}^{(i)})$ and normalised weights $W_{n-1}^{(i)}$ for $i = 1, \dots, N$ are available at time $n-1$, so that we have

$$\pi_{n-1}^N(x_{1:n-1}) = \sum_{i=1}^{N} W_{n-1}^{(i)} \delta_{X_{1:n-1}^{(i)}}(x_{1:n-1}).$$

The update from π_{n-1}^N to π_n^N can be performed by first sampling $X_n^{(i)} \sim q(\cdot|X_{1:n-1}^{(i)})$ and computing the weights w_n at points $X_{1:n}^{(i)} = (X_{1:n-1}^{(i)}, X_n^{(i)})$ using the update rule in (6.2), and finally obtain the normalised weights $W_n^{(i)}$ using (6.4).

The SIS method is summarised in Algorithm 6.1. Being a special case of importance sampling approximation, this SIS approximation $\pi_n^N(\varphi_n)$ has almost sure convergence to π_n for any n (under regular conditions) as the number of particles N tends to infinity; it is also possible to have a central limit theorem for $\pi_n^N(\varphi_n)$ (Geweke, 1989).

Algorithm 6.1: Sequential importance sampling (SIS)

```
for $n = 1, 2, \ldots$ do
    for $i = 1, \ldots, N$ do
        if $n = 1$ then
            sample $X_1^{(i)} \sim q(\cdot)$, calculate $w_1(X_1^{(i)}) = \pi_1(X_1^{(i)})/q_1(X_1^{(i)})$.
        else
            if $n \geq 2$ sample $X_n^{(i)} \sim q(\cdot|X_{1:n-1}^{(i)})$, set $X_{1:n}^{(i)} = (X_{1:n-1}^{(i)}, X_n^{(i)})$, and calculate
                $w_n(X_{1:n}^{(i)}) = w_{n-1}(X_{1:n-1}^{(i)}) \frac{\pi_n(X_{1:n}^{(i)})}{\pi_{n-1}(X_{1:n-1}^{(i)}) q(X_n^{(i)}|X_{1:n-1}^{(i)})}.$
    for $i = 1, \ldots, N$ do
        Calculate
            $W_n^{(i)} = \frac{w_n(X_{1:n}^{(i)})}{\sum_{i=1}^N w_n(X_{1:n}^{(i)})}.$
```

Optimal choice of importance density: As in the non-sequential case, it is important to choose $\{q_n\}_{n\geq 1}$ such that the variances of $\{\pi_n^N(\varphi_n)\}_{n\geq 1}$ are minimised. Recall that in the SIS algorithm we restrict ourselves to $\{q_n(x_{1:n})\}_{n\geq 1}$ satisfying (6.1), therefore selection of the optimal proposal distributions suggested in Section 3.1 may not be possible. Instead, define the *incremental importance weights* as.

$$w_{n|n-1}(x_{1:n}) = \frac{\pi_n(x_{1:n})}{\pi_{n-1}(x_{1:n-1}) q(x_n|x_{1:n-1})}.$$

A more relevant motivation for those $\{q_n(x_{1:n})\}_{n\geq 1}$ satisfying (6.1) might be to minimise the variance of $w_{n|n-1}(X_{1:n})$ conditional on $X_{1:n-1}$.

Note that the objective of minimising the conditional variance of $w_{n|n-1}$ is somehow more general in the sense that it is not specific to φ_n. It was shown in Doucet (1997) that $q^{opt}(x_n|x_{1:n-1})$ by which the variance is minimised is given by

$$q^{opt}(x_n|x_{1:n-1}) = \pi_n(x_n|x_{1:n-1}) = \frac{\pi_n(x_{1:n})}{\int \pi_n(x_{1:n}) dx_n}. \tag{6.5}$$

Before Doucet (1997), the optimum kernel was used in several works for particular applications, see e.g. Kong et al. (1994); Liu and Chen (1995); Chen and Liu (1996). The optimum kernel leads to the optimum incremental weight

$$w_{n|n-1}^{opt}(x_{1:n}) = \frac{\pi_n(x_{1:n-1})}{\pi_{n-1}(x_{1:n-1})} = \frac{\int \pi_n(x_{1:n}) dx_n}{\pi_{n-1}(x_{1:n-1})}. \tag{6.6}$$

which does not depend on the value of x_n.

6.3 Sequential importance sampling resampling

Weight degeneracy: The SIS method is an efficient way of implementing importance sampling sequentially. However; unless the proposal distribution is very close to the true distribution, the importance weight step will lead over a number of iterations to a small number of particles with very large weights compared to the rest of the particles. This will eventually result in one of the normalised weights to being ≈ 1 and the others being ≈ 0, effectively leading to a particle approximation with a single particle, see Kong et al. (1994) and Doucet et al. (2000b). This problem is called the *weight degeneracy* problem.

Resampling: In order to address the weight degeneracy problem, a *resampling* step is introduced at iterations of the SIS method, leading to the *sequential importance sampling resampling* (SISR) algorithm.

Generally, we can describe resampling as a method by which a weighted empirical distribution is replaced with an equally weighted distribution, where the samples of the equally weighted distribution are drawn from the weighted empirical distribution.

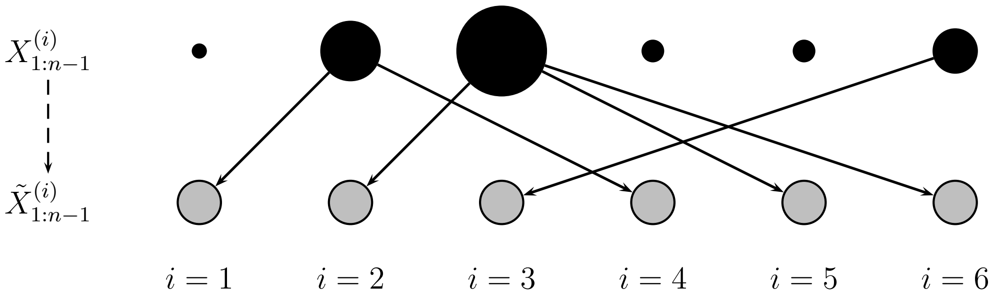


Figure 6.1: Resampling in SISR. Circle sizes represent weights.

In sequential Monte Carlo for $\{\pi_n(x_{1:n})\}_{n\geq 1}$, resampling is applied to $\pi^N_{n-1}(x_{1:n-1})$ before proceeding to approximate $\pi_n(x_{1:n})$. Assume, again, that $\pi_{n-1}(x_{1:n-1})$ is approximated by

$$\pi^N_{n-1}(x_{1:n-1}) = \sum_{i=1}^{N} W^{(i)}_{n-1}\delta_{X^{(i)}_{1:n-1}}(x_{1:n-1}),$$

We draw N independent samples $\widetilde{X}^{(i)}_{1:n-1}$, $i = 1,\ldots,N$ from π^N_{n-1}, such that

$$\mathbb{P}(\widetilde{X}^{(i)}_{1:n-1} = X^{(j)}_{1:n-1}) = W^{(j)}_{n-1}, \quad i,j = 1,\ldots,N.$$

Obviously, this corresponds to drawing N independent samples from a multinomial distribution, therefore this particular resampling scheme is called *multinomial resampling*. Now the resampled particles form an equally weighted discrete distribution

$$\tilde{\pi}^N_{n-1}(x_{1:n-1}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\widetilde{X}^{(i)}_{1:n-1}}(x_{1:n-1}),$$

We proceed to approximating $\pi_n(x_{1:n})$ using $\hat{\pi}^N_{n-1}(x_{1:n-1})$ instead of $\pi^N_{n-1}(x_{1:n-1})$ as follows. After resampling, for each $i = 1, \dots, N$ we sample $X_n^{(i)} \sim q(\cdot|\widetilde{X}^{(i)}_{1:n-1})$, weight the particles $X^{(i)}_{1:n} = (\widetilde{X}^{(i)}_{1:n-1}, X_n^{(i)})$ using

$$\begin{aligned} W_n^{(i)} &\propto w_{n|n-1}(X^{(i)}_{1:n}) \\ &= \frac{\pi_n(X^{(i)}_{1:n})}{\pi_{n-1}(\widetilde{X}^{(i)}_{1:n-1}) q(X_n^{(i)}|\widetilde{X}^{(i)}_{1:n-1})}, \quad \sum_{i=1}^N W_n^{(i)} = 1. \end{aligned}$$

The SISR method, also known as the *particle filter*, is summarised in Algorithm 6.2.

Algorithm 6.2: Sequential importance sampling resampling (SISR)

```
for $n = 1, 2, \dots,$ do
    if $n = 1$ then
        for $i = 1, \dots, N$ do
            sample $X_1^{(i)} \sim q_1(\cdot)$
        for $i = 1, \dots, N$ do
            Calculate
                $$W_1^{(i)} \propto \frac{\pi_1(X_1^{(i)})}{q_1(X_1^{(i)})}.$$
    else
        Resample from $\{X^{(i)}_{1:n-1}\}_{1 \le i \le N}$ according to the weights $\{W^{(i)}_{n-1}\}_{1 \le i \le N}$ to get
        resampled particles $\{\widetilde{X}^{(i)}_{1:n-1}\}_{1 \le i \le N}$ with weight $1/N$.
        for $i = 1, \dots, N$ do
            Sample $X_n^{(i)} \sim q(\cdot|\widetilde{X}^{(i)}_{1:n-1})$, set $X^{(i)}_{1:n} = (\widetilde{X}^{(i)}_{1:n-1}, X_n^{(i)})$
        for $i = 1, \dots, N$ do
            Calculate
                $$W_n^{(i)} \propto \frac{\pi_n(X^{(i)}_{1:n})}{\pi_{n-1}(\widetilde{X}^{(i)}_{1:n-1}) q(X_n^{(i)}|\widetilde{X}^{(i)}_{1:n-1})}.$$
```

Path degeneracy: The importance of resampling in the context of SMC was first demonstrated by Gordon et al. (1993) based on the ideas of Rubin (1987). Although the resampling step alleviates the weight degeneracy problem, it has two drawbacks. Firstly, since after successive resampling steps some of the distinct particles for $X_{1:n}$ are dropped in favour of more copies of highly-weighted particles. This leads to the impoverishment of particles such that for $k \ll n$, very few particles represent the marginal distribution of $X_{1:k}$ under π_n (Andrieu et al., 2005; Del Moral and Doucet, 2003; Olsson et al., 2008).

Hence, whatever being the number of particles, $\pi_n(x_{1:k})$ will eventually be approximated by a single unique particle for all (sufficiently large) n. As a result, any attempt to perform integrations over the path space will suffer from this form of degeneracy, which is called *path degeneracy*. The second drawback is the extra variance introduced by the resampling step. There are a few ways of reducing the effects of resampling.

- One way is adaptive resampling i.e. resampling only at iterations where the effective sample size drops below a certain proportion of N. For a practical implementation, the effective sample size at time n itself should be estimated from particles as well. One particle estimate of $N_{\mathit{eff},n}$ is given in Liu (2001, pp. 35-36)

$$\widetilde{N}_{\mathit{eff},n} = \frac{1}{\sum_{i=1}^{N} W_n^{(i)2}}.$$

- Another way to reduce the effects of resampling is to use alternative resampling methods to multinomial resampling. Let $I_n(i)$ is the number of times the i'th particle is drawn from $\pi_n^N(x_{1:n})$ in a resampling scheme. A number of resampling methods have been proposed in the literature that satisfy $\mathbb{E}\left[I_n(i)\right] = NW_n^{(i)}$ but have different $\mathbb{V}\left[I_n(i)\right]$. The idea behind $\mathbb{E}\left[I_n(i)\right] = NW_n^{(i)}$ is that the mean of the particle approximation to $\pi_n(\varphi_n)$ remains the same after resampling. Standard resampling schemes include multinomial resampling (Gordon et al., 1993), residual resampling (Whitley, 1994; Liu and Chen, 1998), stratified resampling (Kitagawa, 1996), and systematic resampling (Whitley, 1994; Carpenter et al., 1999). There are also some non-standard resampling algorithms such that the particle size varies (randomly) after resampling (e.g. Crisan et al. (1999); Fearnhead and Liu (2007)), or the weights are not constrained to be equal after resampling (e.g. Fearnhead and Clifford (2003); Fearnhead and Liu (2007)).

- A third way of avoiding path degeneracy is provided by the *resample-move* algorithm (Gilks and Berzuini, 2001), where each resampled particle $\widetilde{X}_{1:n}^{(i)}$ is moved according to a MCMC kernel K_n whose invariant distribution is $\pi_n(x_{1:n})$. In fact we could have included this MCMC move step in Algorithm 6.2 to make the algorithm more generic. However, the resample-move algorithm is a useful degeneracy reduction technique usually in a much more general setting. Although possible in principle, it is computationally infeasible to apply a kernel to the path space on which current particles exist as the state space grows at evert iteration of SISR.

- The final method we will mention here that is used to reduce path degeneracy is *block sampling* (Doucet et al., 2006), where at time n one samples components $X_{n-L+1:n}$ for $L > 1$, and previously sampled values for $X_{n-L+1:n-1}$ are simply discarded. In return of the computational cost introduced by L, this procedure reduces the variance of weights and hence reduces the number of resampling steps (if an adaptive resampling strategy is used) dramatically. Therefore, path degeneracy is reduced.

Chapter 7

Bayesian inference in Hidden Markov Models

Summary: *One main application of sequential Monte Carlo methods is Bayesian optimum filtering in hidden Markov models (HMM). We will first introduce HMMs. Then we will see exact sequential inference techniques for finite-state space HMMs and linear Gaussian HMMs, where we do not need SMC methods for certain distributions of interest. Then, we will move on to the general case where the HMM can be non-linear and/or non-Gaussian, see sequential Monte Carlo methods in action for sequential inference in such HMMs.*

7.1 Introduction

HMMs arguably constitute the widest class of time series models that are used for modelling stochastic behaviour of dynamic systems. In Section 7.2, we will introduce HMMs using a formulation that is appropriate for filtering and parameter estimation problems. We will restrict ourselves to discrete time homogenous HMMs whose dynamics for their hidden states and observables admit conditional probability densities which are parametrised by vector valued static parameters. However, this is our only restriction; we keep our framework general enough to cover those models with non-linear non-Gaussian dynamics.

One of the main problems dealt within the framework of HMMs is *optimal Bayesian filtering*, which has many applications in signal processing and related areas such as speech processing (Rabiner, 1989), finance (Pitt and Shephard, 1999), robotics (Gordon et al., 1993), communications (Andrieu et al., 2001), etc. Due to the non-linearity and non-Gaussianity of most of models of interest in real life applications, approximate solutions are inevitable and SMC is the main computational tool used for this; see e.g. Doucet et al. (2001) for a wide selection of examples demonstrating use of SMC. SMC methods have already been presented in its general form in the previous chapter, we will present their application to HMMs for optimal Bayesian filtering in Sections 7.3 and 7.3.4.

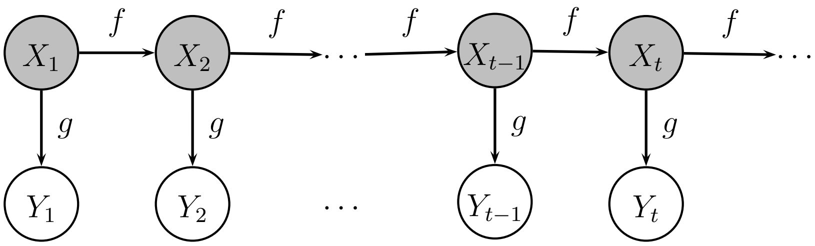


Figure 7.1: Acyclic directed graph for HMM

7.2 Hidden Markov models

We begin with the definition of a HMM. Let $\{X_t\}_{t\geq 1}$ be a homogenous Markov chain with state-space $\mathcal{X}$, initial density $\eta(x_1)$ and transition density $f(x_t|x_{t-1})$. Suppose that this process is observed as another process $\{Y_t\}_{t\geq 1}$ on $\mathcal{Y}$ such that the conditional distribution on Y_t given all the other random variables depends only on X_t and has the conditional density $g(y_t|x_t)$. Then the bivariate process $\{X_t, Y_t\}_{t\geq 1}$ is called a HMM. For any $n \geq 1$, the joint probability density of $(X_{1:n}, Y_{1:n})$ is given by

$$p(x_{1:n}, y_{1:n}) = \underbrace{\eta(x_1)\prod_{t=2}^{n} f(x_t|x_{t-1})}_{\text{latent Markov process}}\underbrace{\prod_{t=1}^{n} g(y_t|x_t)}_{\text{observations}} \tag{7.1}$$

Figure 7.1 shows the diagram for the HMM.

The joint law of all the variables of the HMM up to time n is summarised in (7.1) from which we derive several probability densities of interest. One example is the evidence of the observations up to time n which can be derived as

$$p(y_{1:n}) = \int p(x_{1:n}, y_{1:n})dx_{1:n}. \tag{7.2}$$

Another important probability density, which will be pursued in detail, is the density of the posterior distribution of $X_{1:n}$ given $Y_{1:n} = y_{1:n}$, which is obtained by using the Bayes' theorem

$$p(x_{1:n}|y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \tag{7.3}$$

In the time series literature, the term HMM has been widely associated with the case of $\mathcal{X}$ being finite (Rabiner, 1989) and those models with continuous $\mathcal{X}$ are often referred to as state-space models. Again, in some works the term 'state-space models' refers to the case of linear Gaussian systems (Anderson and Moore, 1979). We emphasise at this point that in this text we shall keep the framework as general as possible. We consider the general case of measurable spaces and we avoid making any restrictive assumptions on $\eta(x_1)$, $f(x_t|x_{t-1})$, and $g(y_t|x_t)$ that impose a certain structure on the dynamics of the

HMM. Also, we clarify that in contrast to previous restrictive use of terminology, we will use both terms 'HMM' and 'general state space model' to describe exactly the same thing.

Example 7.1 (**A finite state-space HMM for weather conditions**). *Assume that the weather condition in terms of atmospheric pressure is simplified to have two states, "Low" and "High", and on day t, $X_t \in \mathcal{X} = \{1, 2\}$ denotes the state of the atmospheric condition in terms of pressure, where 1 stands for "Low" and 2 stands for "High". Further $\{X_t\}_{t \geq 1}$ is modelled as a Markov chain with some initial distribution $\eta = [\eta(1), \eta(2)]$, and transition density*

$$F = \begin{bmatrix} 0.3 & 0.7 \\ 0.2 & 0.8 \end{bmatrix}$$

where $F(i, j) = \mathbb{P}(X_{t+1} = j | X_t = i) = f(j|i)$. What we observe is not the atmospheric pressure but, whether a day is "Dry", "Cloudy", or "Rainy", and these conditions are enumerated with 1, 2, and 3, respectively. Let $Y_t \in \mathcal{Y} = \{1, 2, 3\}$ is the observed weather condition on day t. It is known that low pressure is more likely to lead clouds or precipitation than high pressure, and assumed that given X_t, Y_t is conditionally independent from $Y_{1:t-1}$ and $X_{1:t-1}$. The conditional observation matrix that related X_t to Y_t is given by

$$G = \begin{bmatrix} 0.3 & 0.4 & 0.3 \\ 0.6 & 0.3 & 0.1 \end{bmatrix}$$

where $G(i, j) = \mathbb{P}(Y_t = j | X_t = i) = g(j|i)$. Then, $\{X_t, Y_t\}_{t \geq 1}$ forms a HMM, and since $\mathcal{X}$ is finite, it is called a finite state-space HMM.

Example 7.2 (**Linear Gaussian HMM**). *A generic linear Gaussian HMM $\{X_t, Y_t\}$, where $X_t \in \mathbb{R}^{d_x}$, and $Y_t \in \mathbb{R}^{d_y}$ are vector valued hidden and observed states, can be defined via the following generative definitions for the random variables $\{X_t, Y_t\}$:*

$$X_1 \sim \mathcal{N}(\mu_1, \Sigma_1), \quad X_t = AX_{t-1} + U_t, \quad U_t \sim \mathcal{N}(0, S), \quad t > 1 \tag{7.4}$$

$$Y_t = BX_t + V_t, \quad V_t \sim \mathcal{N}(0, R), \tag{7.5}$$

Here, A, B are $d_x \times d_x$ and $d_y \times d_x$ matrices, and S and R are $d_x \times d_x$ and $d_y \times d_y$ covariance matrices for the state and observation processes, respectively. In terms of densities, this HMM can be described as

$$\eta(x_1) = \phi(x_1; \mu_1, \Sigma_1), \quad f(x_t|x_{t-1}) = \phi(x_t; Ax_{t-1}, S), \quad g(y_t|x_t) = \phi(y_t; Bx_t, R). \tag{7.6}$$

Example 7.3 (**A partially observed moving target**). *We modify the source localisation problem in Example 5.15 by adding to the scenario that the source is moving in a Markovian fashion: The motion of the source is modelled as a Markov chain for its velocity and position. Let $V_t = (V_t(1), V_t(2))$ and $P_t = (P_t(1), P_t(2))$ be the velocity and the position vectors (in the xy plane) of the source at time t and assume that they evolve according to its following stochastic dynamics: $V_1(i) \sim \mathcal{N}(0, \sigma^2_{bv})$,*

$$\begin{aligned} &V_1(i) \sim \mathcal{N}(0, \sigma^2_{bv}), \quad P_1(i) \sim \mathcal{N}(0, \sigma^2_{bp}), \quad i = 1, 2, \\ &V_t(i) = aV_{t-1}(i) + U_t(i), \quad P_t(i) = P_{t-1}(i) + \Delta V_{t-1}(i) + Z_t(i), \quad i = 1, 2. \end{aligned}$$

where $U_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_v^2)$ and $Z_t(i) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_p^2)$. This model dictates that the velocity in each direction changes independently according to an autoregressive model with the regression parameter a and driving variance σ_v^2 and the position is the previous position plus the previous velocity multiplied by the factor Δ which corresponds to the time interval between successive time steps $t-1$ and t, plus some noise which counts for the discretisation error. Let $X_t = (V_t, P_t)$. X_t is a Markov chain with transition density

$$f(x_t|x_{t-1}) = \prod_{i=1}^{2} \phi(v_t(i); av_{t-1}(i), \sigma_v^2) \prod_{i=1}^{2} \phi(p_t(i); \Delta v_{t-1}(i) + p_{t-1}(i); \sigma_p^2)$$

or, in matrix form

$$f(x_t|x_{t-1}) = \phi(Fx_{t-1}, \Sigma_x), \quad F = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ \Delta & 0 & 1 & 0 \\ 0 & \Delta & 0 & 1 \end{bmatrix}, \quad \Sigma_x = \begin{bmatrix} \sigma_v^2 & 0 & 0 & 0 \\ 0 & \sigma_v^2 & 0 & 0 \\ 0 & 0 & \sigma_p^2 & 0 \\ 0 & 0 & 0 & \sigma_p^2 \end{bmatrix}$$

The observations are generated as before, i.e. at each time t three distance measurements $(R_{t,1}, R_{t,2}, R_{t,3})$ with

$$R_{t,i} = [(P_t(1) - S_i(1))^2 + (P_t(2) - S_i(2))^2]^{1/2}, \quad i = 1, 2, 3,$$

from three different sensors are collected in Gaussian noise with variance σ_y^2 and these measurements form $Y_t = (Y_{t,1}, Y_{t,2}, Y_{t,3})$

$$Y_{t,i} = R_{t,i} + E_{t,i}, \quad E_{t,i} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_y^2), \quad i = 1, 2, 3.$$

so that

$$g(y_t|x_t) = \prod_{i=1}^{3} \phi(y_{t,i}; r_{t,i}, \sigma_y^2).$$

This is an example to a non-linear HMM due to the non-linearity in its observation dynamics.

7.3 Bayesian optimal filtering and smoothing

In a HMM, one is usually interested in sequential inference on the variables of the hidden process $\{X_t\}_{t\geq1}$ given observations $\{Y_t\}_{t\geq1}$ up to time n. For example, one pursues for the sequence of posterior distributions $\{p(x_{1:t}|y_{1:t})\}_{t\geq1}$, where $p(x_{1:t}|y_{1:t})$ is given in equation (7.3). It is also straightforward to generalise $p(x_{1:t}|y_{1:t})$ to the posterior distributions of $X_{1:t'}$ for any $t' \geq 1$. For $t' > t$ we have

$$p(x_{1:t'}|y_{1:t}) = p(x_{1:t}|y_{1:t}) \prod_{\tau=t+1}^{t'} f(x_\tau|x_{\tau-1});$$

whereas for $t' < t$ the density $p(x_{1:t'}|y_{1:t})$ can be obtained simply by integrating out the variables $x_{t'+1:t}$, i.e.

$$p(x_{1:t'}|y_{1:t}) = \int p(x_{1:t}|y_{1:t})dx_{t'+1:t}.$$

7.3.1 Filtering, prediction, and smoothing

From a Bayesian point of view, the probability densities $p(x_{1:n'}|y_{1:n})$ are complete solutions to the inference problems as they contain all the information about the hidden states $X_{1:n'}$ given the observations $y_{1:n}$. For example, the expectation of a function $\varphi_{n'} : \mathcal{X}^{n'} \to \mathbb{R}$ conditional upon the observations $y_{1:n}$ can be evaluated as

$$\mathbb{E}\left[\varphi_n(X_{1:n'})|Y_{1:n} = y_{1:n}\right] = \int \varphi(x_{1:n'})p(x_{1:n'}|y_{1:n})dx_{1:n'}.$$

However, one can restrict their focus to a problem of smaller size, such as the marginal distribution of the random variable X_k, $k \leq n'$, given $y_{1:n}$. The probability density of such a marginal posterior distribution $p(x_k|y_{1:n})$ is called a *filtering, prediction, or smoothing* density if $k = n$, $k > n$ and $k < n$, respectively. Indeed, there are many cases where one is interested in calculating the expectations of functions $\varphi : \mathcal{X} \to \mathbb{R}$ of X_k given $y_{1:n}$

$$\mathbb{E}\left[\varphi(X_k)|Y_{1:n} = y_{1:n}\right] = \int \varphi(x_k)p(x_k|y_{1:n})dx_k.$$

Although once we have $p(x_{1:n'}|y_{1:n})$ for $n' \geq k$ the marginal density can directly be obtained by marginalisation, the recursion in (7.24) may be intractable or too expensive to calculate. Therefore it is useful to use alternative recursion techniques to effectively evaluate the marginal densities sequentially.

7.3.1.1 Forward filtering backward smoothing

Here, we will see the technique called *forward filtering backward smoothing* that combines the recursions for the filtering and one-step prediction densities as well as a backward recursion for smoothing densities.

Forward filtering (and prediction): We start with $p(x_1|y_0) := p(x_1) = \eta(x_1)$ and

$$p(x_1|y_1) = \frac{\eta(x_1)g(y_1|x_1)}{p(y_1)}.$$

where $p(y_1) = \int \eta(x_1')g(y_1|x_1')dx_1$. Given the filtering density $p(x_{t-1}|y_{1:t-1})$ at time $t-1$ and the new observation y_t at time t, the filtering density at time t can be obtained recursively in two stages, which are called prediction and update. These are given as

$$\begin{aligned} p(x_t|y_{1:t-1}) &= \int p(x_t|x_{t-1}, y_{1:t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \\ &= \int f(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}, \quad (7.7) \\ p(x_t|y_{1:t}) &= \frac{p(y_t|x_t, y_{1:t-1})p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \\ &= \frac{g(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}. \quad (7.8) \end{aligned}$$

where this time we write the normalising constant as

$$p(y_t|y_{1:t-1}) = \int p(x_t|y_{1:t-1})g(y_t|x_t)dx_t. \quad (7.9)$$

Actually, (7.9) is important for its own sake, since it leads to two important quantities: First, given $y_{1:n}$, the posterior predictive density for Y_{n+1} is simply $p(y_{n+1}|y_{1:n})$, which can be calculated from $p(x_{n+1}|y_{1:n})$. Secondly, (7.9) can be used to calculate the evidence recursively.

$$p(y_{1:n}) = \prod_{t=1}^{n} p(y_t|y_{1:t-1}) = p(y_{1:n-1})p(y_n|y_{1:n-1}). \quad (7.10)$$

The problem of evaluating the recursion given by equations (7.7) and (7.8) is called the *Bayesian optimal filtering* (or shortly *optimum filtering*) problem in the literature.

Backward smoothing: Once we have the forward filtering recursion to calculate the filtering and prediction densities $p(x_t|y_{1:t})$ and $p(x_t|y_{1:t-1})$ for $t = 1, \dots, n$, where n is the total number of observations, there are more than one ways of performing smoothing in a HMM to calculate $p(x_t|y_{1:n})$, $t = 1, \dots, n$. We will see the one that corresponds to *forward filtering backward smoothing*. As the name suggests, backward smoothing is performed via a backward recursion in time, i.e. $p(x_t|y_{1:n})$ is calculated in the order $t = n, n-1, \dots, 1$. Now let us see how one step of the backward recursion works: Given $p(x_{t+1}|y_{1:n})$, we find $p(x_t|y_{1:n})$ by exploiting the following relation

$$\begin{aligned} p(x_t|y_{1:n}) &= \int p(x_t, x_{t+1}|y_{1:n})dx_{t+1} \\ &= \int p(x_{t+1}|y_{1:n})p(x_t|x_{t+1}, y_{1:n})dx_{t+1}. \quad (7.11) \end{aligned}$$

which can be written for any time series model. Thanks to the particular structure of the HMM, given X_{t+1}, X_t is conditionally independent from the rest of the future variables (try to see this from Figure 7.1). Hence

$$p(x_t|x_{t+1}, y_{1:n}) = p(x_t|x_{t+1}, y_{1:t}) = \frac{p(x_t|y_{1:t})f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} \quad (7.12)$$

In fact, one can derive this analytically as

$$\begin{aligned}
p(x_t|x_{t+1}, y_{1:n}) &= p(x_t|x_{t+1}, y_{1:t}, y_{t+1:n}) \\
&= \frac{p(x_t|y_{1:t})p(x_{t+1}, y_{t+1:n}|x_t, y_{1:t})}{p(x_{t+1}, y_{t+1:n}|y_{1:t})} \\
&= \frac{p(x_t|y_{1:t})p(x_{t+1}|x_t, y_{1:t})p(y_{t+1:n}|x_{t+1}, x_t, y_{1:t})}{p(x_{t+1}|y_{1:t})p(y_{t+1:n}|x_{t+1}, y_{1:t})} \\
&= \frac{p(x_t|y_{1:t})f(x_{t+1}|x_t)p(y_{t+1:n}|x_{t+1})}{p(x_{t+1}|y_{1:t})p(y_{t+1:n}|x_{t+1})} \\
&= \frac{p(x_t|y_{1:t})f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})}
\end{aligned}$$

where the last expression is indeed exactly $p(x_t|x_{t+1}, y_{1:n})$. Substituting this into (7.11), we have

$$p(x_t|y_{1:n}) = \int p(x_{t+1}|y_{1:n})\frac{p(x_t|y_{1:t})f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})}dx_{t+1} \tag{7.13}$$

which involves the filtering and prediction distributions that we have calculated in the forward filtering stage already.

There are cases when the optimum filtering problem can be solved exactly. One such case is when $\mathcal{X}$ is a finite countable set (Rabiner, 1989). Also, in linear Gaussian state-space models the densities in (7.7) and (7.8) are obtained by the *Kalman filter* (Kalman, 1960).

7.3.1.2 Sampling from the full posterior

In order to sample from the posterior distribution $p(x_{1:n}|y_{1:n})$, we can exploit the following factorisation:

$$\begin{aligned}
p(x_{1:n}|y_{1:n}) &= p(x_n|y_{1:n}) \prod_{t=n-1}^{1} p(x_t|x_{t+1:n}, y_{1:n}) \\
&= p(x_n|y_{1:n}) \prod_{t=n-1}^{1} p(x_t|x_{t+1}, y_{1:t})
\end{aligned} \tag{7.14}$$

where the second line is crucial and it follows from the specific dependency structure of the HMM. Equation (7.14) suggests that we can start sampling X_n from the filtering distribution at time n, and go backwards to sample $X_{n-1}, X_{n-2}, \dots, X_1$, using the backward transition probabilities. Note that one needs all the filtering distributions up to time n in order to perform this backward sampling. That is why the algorithm that executes this scheme to sample from $p(x_{1:n}|y_{1:n})$ is called *forward filtering backward sampling*.

7.3.2 Exact inference in finite state-space HMMs

In a finite state-space HMM, as exemplified in Example 7.1, X_t takes values from a finite set $\mathcal{X}$ of size k, and for simplicity we assume that the states are enumerated from 1 to k, implying $\mathcal{X} = \{1, \ldots, k\}$. Define the $1 \times k$ vectors α_t, β_t, γ_t for $t = 1, \ldots, n$ that represent the filtering, prediction, and smoothing probabilities, respectively:

$$\alpha_t(i) := \mathbb{P}(X_t = i | Y_{1:t} = y_{1:t}), \quad i = 1, \ldots, k, \quad t = 1, \ldots, n$$
$$\beta_t(i) := \mathbb{P}(X_t = i | Y_{1:t-1} = y_{1:t-1}), \quad i = 1, \ldots, k, \quad t = 1, \ldots, n$$
$$\gamma_t(i) := \mathbb{P}(X_t = i | Y_{1:n} = y_{1:n}), \quad i = 1, \ldots, k, \quad t = 1, \ldots, n$$

The forward filtering backward smoothing algorithm for a finite-state HMM is given in Algorithm 7.1. The recursions given in the algorithms are simply the discrete versions of Equations (7.7), (7.8), and (7.13). In order to keep track of $p(y_t|y_{1:t-1})$ (hence $p(y_{1:t})$) as well, one can include the following (with the convention $p(y_1|y_0) = p(y_1)$)

$$p(y_t|y_{1:t-1}) = \sum_{i=1}^{k} \beta_t(i) g(y_t|i).$$

Algorithm 7.1: Forward filtering backward smoothing in finite state-space HMM

```
Input: Observations $y_{1:n}$, HMM transition and observation probabilities $\eta$, $f$, $g$
Output: $\alpha_t$, $\beta_t$, $\gamma_t$, $t = 1, \ldots, n$.
                         Forward filtering
for $t = 1, \ldots, n$ do
    Prediction: If $t = 1$, set $\beta_1(i) = \eta(i)$, $i = 1, \ldots, k$; else
        $\beta_t(i) = \sum_{j=1}^{k} \alpha_{t-1}(j) f(i|j), \quad i = 1, \ldots, k.$
    Filtering:
        $\alpha_t(i) = \frac{\beta_t(i) g(y_t|i)}{\sum_{j=1}^{k} \beta_t(j) g(y_t|j)}, \quad i = 1, \ldots, k.$
                         Backward smoothing
for $t = n, \ldots, 1$ do
    Smoothing: If $t = n$, set $\gamma_n(i) = \alpha_n(i)$, $i = 1, \ldots, k$; else
        $\gamma_t(i) = \sum_{j=1}^{k} \gamma_{t+1}(j) \frac{\alpha_t(i) f(j|i)}{\beta_{t+1}(j)}, \quad i = 1, \ldots, k.$
```

Forward filtering backward sampling: For the finite state-space HMM, the backward transition probabilities are given as

$$\mathbb{P}(X_t = i | X_{t+1} = j, Y_{1:t} = y_{1:t}) = \frac{\alpha_t(i) f(j|i)}{\beta_{t+1}(j)}$$

The resulting forward filtering backward sampling algorithm for finite state-space HMMs to sample from the full posterior $p(x_{1:n}|y_{1:n})$ is given in Algorithm 7.2.

Algorithm 7.2: Forward filtering backward sampling in finite state-space HMM

```
Input: Observations $y_{1:n}$, HMM transition and observation probabilities $\eta$, $f$, $g$
Output: $X_{1:n} \sim p(x_{1:n}|y_{1:n})$.
                          Forward filtering
1 Perform forward filtering in the first part ot Algorithm 7.1 to obtain $\alpha_t$, $\beta_t$,
  $t = 1, \dots, n$.
                          Backward sampling                                  .
for $t = n, \dots, 1$ do
2     Smoothing: If $t = n$, sample $X_n = i$ with probability $\alpha_n(i)$, $i = 1, \dots, k$; else,
      given $X_{t+1} = x_{t+1}$ sample $X_t = i$ with probability $\frac{\alpha_t(i) f(x_{t+1}|i)}{\beta_{t+1}(x_{t+1})}$, $i = 1, \dots, k$.
```

7.3.3 Exact inference in linear Gaussian HMMs

Consider the linear Gaussian HMM in Example 7.2, where the initial, state transition, and observation distributions are given in Equations 7.4 and 7.5, which are repeated here

$$\begin{aligned}
&X_1 \sim \mathcal{N}(\mu_1, \Sigma_1), \quad X_t = AX_{t-1} + U_t, \quad U_t \sim \mathcal{N}(0, S), \quad t > 1 \\
&Y_t = BX_t + V_t, \quad V_t \sim \mathcal{N}(0, R)
\end{aligned}$$

Since this is a linear and Gaussian HMM, the filtering, prediction, and smoothing distributions have to be Gaussian as well. For any k and n, let the mean and the covariance of the posterior distribution of X_k given $Y_{1:n} = y_{1:n}$ be $\mu_{k|n}$ and $P_{k|n}$, respectively. Then, we denote the distributions of interest as

$$\begin{aligned}
X_t | Y_{1:t} = y_{1:t} &\sim \mathcal{N}(\mu_{t|t}, P_{t|t}), & t &= 1, \dots, n, \\
X_t | Y_{1:t-1} = y_{1:t-1} &\sim \mathcal{N}(\mu_{t|t-1}, P_{t|t-1}), & t &= 1, \dots, n, \\
X_t | Y_{1:n} = y_{1:n} &\sim \mathcal{N}(\mu_{t|n}, P_{t|n}), & t &= 1, \dots, n;
\end{aligned}$$

or, in terms of densities,

$$\begin{aligned}
p(x_t|y_{1:t}) &= \phi(x_t; \mu_{t|t}, P_{t|t}), & t &= 1, \dots, n, \\
p(x_t|y_{1:t-1}) &= \phi(x_t; \mu_{t|t-1}, P_{t|t-1}), & t &= 1, \dots, n, \\
p(x_t|y_{1:n}) &= \phi(x_t; \mu_{t|n}, P_{t|n}), & t &= 1, \dots, n.
\end{aligned}$$

Moreover, as we will see, the mean and the covariance of these distributions are tractable.

Forward filtering: The prediction update from $(\mu_{t-1|t-1}, P_{t-1|t-1})$ to $(\mu_{t|t-1}, P_{t|t-1})$ can be deduced from Equation (7.7), but a simpler way of achieving this is noticing that the update is simply an application of linear transformation of Gaussian variables: Since $X_t = AX_{t-1} + U_t$, and U_t is independent from all the other variables and Gaussian, too, we have

$$\begin{aligned}
\mu_{t|t-1} &= \mathbb{E}[X_t|Y_{1:t-1} = y_{1:t-1}] \qquad (7.15)\\
&= \mathbb{E}[AX_{t-1} + U_t|Y_{1:t-1} = y_{1:t-1}]\\
&= \mathbb{E}[AX_{t-1}|Y_{1:t-1} = y_{1:t-1}] + \mathbb{E}[U_t|Y_{1:t-1} = y_{1:t-1}]\\
&= A\mathbb{E}[X_{t-1}|Y_{1:t-1} = y_{1:t-1}] + \mathbb{E}[U_t]\\
&= A\mu_{t-1|t-1} + 0\\
&= A\mu_{t-1|t-1}. \qquad (7.16)
\end{aligned}$$

For the covariance of the prediction distribution, we have

$$\begin{aligned}
P_{t|t-1} &= \mathrm{Cov}[X_t|Y_{1:t-1} = y_{1:t-1}]\\
&= \mathrm{Cov}[AX_{t-1} + U_t|Y_{1:t-1} = y_{1:t-1}]\\
&= \mathrm{Cov}[AX_{t-1}|Y_{1:t-1} = y_{1:t-1}] + \mathrm{Cov}[U_t|Y_{1:t-1} = y_{1:t-1}]\\
&= A\mathrm{Cov}[X_{t-1}|Y_{1:t-1} = y_{1:t-1}]A^T + \mathrm{Cov}[U_t]\\
&= AP_{t-1|t-1}A^T + S. \qquad (7.17)
\end{aligned}$$

By using (7.9), we can derive the mean $\mu^y_{t|t-1}$ and the covariance $P^y_{t|t-1}$of the conditional density $p(y_t|y_{1:t-1})$, which we know to be a Gaussian density. An alternative way is to derive the moments as above:

$$\begin{aligned}
\mu^y_{t|t-1} = \mathbb{E}[Y_t|Y_{1:t-1} = y_{1:t-1}] &= \mathbb{E}[BX_t + V_t|Y_{1:t-1} = y_{1:t-1}]\\
&= \mathbb{E}[BX_t|Y_{1:t-1} = y_{1:t-1}] + \mathbb{E}[V_t|Y_{1:t-1} = y_{1:t-1}]\\
&= B\mathbb{E}[X_{t-1}|Y_{1:t-1} = y_{1:t-1}] + \mathbb{E}[V_t]\\
&= B\mu_{t|t-1} \qquad (7.18)
\end{aligned}$$

and

$$\begin{aligned}
P^y_{t|t-1} &= \mathrm{Cov}[Y_t|Y_{1:t-1} = y_{1:t-1}]\\
&= \mathrm{Cov}[BX_t + V_t|Y_{1:t-1} = y_{1:t-1}]\\
&= \mathrm{Cov}[BX_t|Y_{1:t-1} = y_{1:t-1}] + \mathrm{Cov}[V_t|Y_{1:t-1} = y_{1:t-1}]\\
&= B\mathrm{Cov}[X_{t-1}|Y_{1:t-1} = y_{1:t-1}]B^T + \mathrm{Cov}[V_t]\\
&= BP_{t|t-1}B^T + R. \qquad (7.19)
\end{aligned}$$

The filtering distribution $p(x_t|y_{1:t})$ can be found by applying the Bayes theorem with prior $p(x_t|y_{1:t-1})$ and likelihood $g(y_t|x_t)$. Since both are Gaussian and the relation between

Y_t and X_t is linear, we can apply the conjugacy result for the mean parameter of the normal distribution in Example 4.7 and deduce

$$P_{t|t} = (P_{t|t-1}^{-1} + B^T R^{-1} B)^{-1}$$

and

$$\mu_{t|t} = P_{t|t}(P_{t|t-1}^{-1} m_{t|t-1} + B^T R^{-1} y_t)$$

Using the matrix inversion lemma[1] and letting $P_{t|t-1}^{xy} = P_{t|t-1}B^T$, we can rewrite

$$\begin{aligned} P_{t|t} &= P_{t|t-1} - P_{t|t-1}B^T(R + BP_{t|t-1}B^T)^{-1}BP_{t|t-1} \\ &= P_{t|t-1} - P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}{P_{t|t-1}^{xy}}^{T} \end{aligned} \tag{7.20}$$

and

$$\begin{aligned} \mu_{t|t} &= (P_{t|t-1} - P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}{P_{t|t-1}^{xy}}^{T})(P_{t|t-1}^{-1}\mu_{t|t-1} + B^T R^{-1} y_t) \\ &= \mu_{t|t-1} + P_{t|t-1}^{xy}R^{-1}y_t - P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}{P_{t|t-1}^{xy}}^{T}P_{t|t-1}^{-1}\mu_{t|t-1} - P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}{P_{t|t-1}^{xy}}^{T}B^T R^{-1} y_t \\ &= \mu_{t|t-1} + P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}(P_{t|t-1}^{y}R^{-1} - {P_{t|t-1}^{xy}}^{T}B^T R^{-1})y_t - P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}{P_{t|t-1}^{xy}}^{T}P_{t|t-1}^{-1}\mu_{t|t-1} \\ &= \mu_{t|t-1} + P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}([BP_{t|t-1}B^T + R]R^{-1} - {P_{t|t-1}^{xy}}^{T}B^T R^{-1})y_t - P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}B\mu_{t|t-1} \\ &= \mu_{t|t-1} + P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}(BP_{t|t-1}B^T R^{-1} + I - BP_{t|t-1}B^T R^{-1})y_t - P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}\mu_{t|t-1}^{y} \\ &= \mu_{t|t-1} + P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}y_t - P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}\mu_{t|t-1}^{y} \\ &= \mu_{t|t-1} + P_{t|t-1}^{xy}{P_{t|t-1}^{y}}^{-1}(y_t - \mu_{t|t-1}^{y}) \end{aligned} \tag{7.21}$$

Backward smoothing: For backward smoothing, we start from $\mu_{n|n}$ and $P_{n|n}$, which are already calculated in the last step of the forward filtering recursion, and go backwards to derive $\mu_{t|n}$ and $P_{t|n}$ from $\mu_{t+1|n}$ and $P_{t+1|n}$. Observing (7.13) first we need to derive the backward transition density

$$p(x_t|x_{t+1}, y_{1:t}) = \frac{p(x_t|y_{1:t})f(x_{t+1}|x_t)}{p(x_{t+1|y_{1:t}})}$$

This is in the form of the Bayes' rule, with prior $p(x_t|y_{1:t}) = \phi(x_t; \mu_{t|t}, P_{t|t})$ and likelihood $f(x_{t+1}|x_t) = \phi(x_{t+1}; Ax_t, S)$. Since the relation is Gaussian and the prior and likelihood densities are Gaussian, we know that $p(x_t|x_{t+1}, y_{1:t})$ is Gaussian, too,. In order to derive its mean $\mu_{t|t+1}^{x}$ and covariance $P_{t|t+1}^{x}$, we make use of the result in Example 4.7 again to arrive at

$$P_{t|t+1}^{x} = (P_{t|t}^{-1} + A^T S^{-1} A)^{-1}$$

$$\mu_{t|t+1}^{x} = P_{t|t+1}^{x}(P_{t|t}^{-1}\mu_{t|t} + A^T S^{-1} x_{t+1})$$

[1] For invertible matrices A, B and any two matrices U and V of suitable size, the lemma states that $(A + UBV)^{-1} = A^1 - A^{-1}U(B^{-1} + VA^{-1}U)^{-1}VA^{-1}$

Using the matrix inversion lemma again, and letting $\Gamma_{t|t+1} = P_{t|t} A^T P_{t+1|t}^{-1}$, we rewrite those moments as

$$P_{t|t+1}^x = P_{t|t} - \Gamma_{t|t+1} A P_{t|t}$$
$$\mu_{t|t+1}^x = \mu_{t|t} + \Gamma_{t|t+1}(x_{t+1} - \mu_{t+1|t})$$

One can (artificially but usefully) view this relation as follows: given $Y_{1:n} = y_{1:n}$, X_t can be written in terms of X_{t+1} as follows:

$$X_t = \mu_{t|t} + \Gamma_{t|t+1}(X_{t+1} - \mu_{t|t-1}) + E_t,$$

where $X_{t+1}|Y_{1:n} = y_{1:n} \sim \mathcal{N}(\mu_{t+1|n}, P_{t+1|n})$ and $E_t|Y_{1:n} = y_{1:n} \sim \mathcal{N}(0, P_{t|t+1}^x)$ and E_t is independent from X_{t+1} given $Y_{1:n}$. From this, we use the composition rule for the Gaussian distributions to derive $\mu_{t|n}$ and $P_{t|n}$ from $\mu_{t+1|n}$ and $P_{t+1|n}$. Letting , we have

$$\begin{aligned} \mu_{t|n} &= \mathbb{E}[\mu_{t|t} + \Gamma_{t|t+1}(X_{t+1} - \mu_{t+1|t}) + E_t|Y_{1:n} = y_{1:n}] \\ &= \mu_{t|t} + \Gamma_{t|t+1}(\mathbb{E}[X_{t+1}|Y_{1:n} = y_{1:n}] - \mu_{t+1|t}) \\ &= \mu_{t|t} + \Gamma_{t|t+1}(\mu_{t|n} - \mu_{t+1|t}) \end{aligned} \tag{7.22}$$

and

$$\begin{aligned} P_{t|n} &= \mathrm{Cov}[\mu_{t|t} + \Gamma_{t|t+1}(X_{t+1} - \mu_{t+1|t}) + E_t|Y_{1:n} = y_{1:n}] \\ &= \Gamma_{t|t+1}\mathrm{Cov}[X_{t+1} - \mu_{t+1|t}|Y_{1:n} = y_{1:n}]\Gamma_{t|t+1}^T + P_{t|t+1}^x \\ &= \Gamma_{t|t+1}\mathrm{Cov}[X_{t+1}|Y_{1:n} = y_{1:n}]\Gamma_{t|t+1}^T + P_{t|t+1}^x \\ &= \Gamma_{t|t+1} P_{t+1|n} \Gamma_{t|t+1}^T + P_{t|t} - \Gamma_{t|t+1} A P_{t|t} \\ &= P_{t|t} + \Gamma_{t|t+1}(P_{t+1|n} - P_{t+1|t})\Gamma_{t|t+1}^T \end{aligned} \tag{7.23}$$

The forward filtering backward smoothing recursions are given in Algorithm 7.3. In order to keep track of $p(y_t|y_{1:t-1})$ (hence $p(y_{1:t})$) as well, one can include the following (with the convention $p(y_1|y_0) = p(y_1)$)

$$p(y_t|y_{1:t-1}) = \phi(y_t; \mu_{t|t-1}^y, P_{t|t-1}^y)$$

Forward filtering backward sampling: Similar to the finite state-space case, with the help of backward transition distributions, we can sample from the full posterior $p(x_{1:n}|y_{1:n})$ in a linear Gaussian HMM by using forward filtering backward sampling, which is given in Algorithm 7.4.

7.3.4 Particle filters for optimal filtering in HMM

We saw the two cases where the optimum filtering problem can be solved exactly. In general, however, these densities do not admit a close form expression and one has to use methods based on numerical approximations. In the following, we will look at the SMC methodology in the context of general HMMs and review how SMC methods have been used to provide approximate solutions to the optimal filtering problem.

Algorithm 7.3: Forward filtering backward smoothing in linear Gaussian HMM

```
Input: Observations $y_{1:n}$, HMM transition and observation parameters $A$, $B$, $S$, $R$,
       $\mu_1$, $\Sigma_1$
Output: $\mu_{t|t-1}, P_{t|t-1}$, $\mu_{t|t}, P_{t|t}$, $\mu_{t|n}, P_{t|n}$, $t = 1, \ldots, n$.
                    Forward filtering (Kalman filtering)
for $t = 1, \ldots, n$ do
    Prediction: If $t = 1$, set $\mu_{1|0} = \mu_1$, $P_{1|0} = \Sigma_1$; else
        $\mu_{t|t-1} = A\mu_{t-1|t-1}$
        $P_{t|t-1} = AP_{t-1|t-1}A^T + S$
    Filtering:
        $P^y_{t|t-1} = BP_{t|t-1}B^T + R$
        $\mu^y_{t|t-1} = B\mu_{t|t-1}$
        $P^{xy}_{t|t-1} = P_{t|t-1}B^T$
        $\mu_{t|t} = \mu_{t|t-1} + P^{xy}_{t|t-1}{P^y_{t|t-1}}^{-1}(y_t - \mu^y_{t|t-1})$
        $P_{t|t} = P_{t|t-1} - P^{xy}_{t|t-1}{P^y_{t|t-1}}^{-1}{P^{xy}_{t|t-1}}^T$
                    Backward smoothing
for $t = n-1, \ldots, 1$ do
    Smoothing:
        $\Gamma_{t|t+1} = P_{t|t}A^T P^{-1}_{t+1|t}$
        $\mu_{t|n} = \mu_{t|t} + \Gamma_{t|t+1}(\mu_{t|n} - \mu_{t+1|t})$
        $P_{t|n} = P_{t|t} + \Gamma_{t|t+1}(P_{t+1|n} - P_{t+1|t})\Gamma^T_{t|t+1}$
```

Algorithm 7.4: Forward filtering backward sampling in linear Gaussian HMM

```
Input: Observations $y_{1:n}$, HMM transition and observation parameters $A$, $B$, $S$, $R$,
       $\mu_1$, $\Sigma_1$
Output: $\mu_{t|t-1}, P_{t|t-1}, \mu_{t|t}, P_{t|t}, \mu_{t|n}, P_{t|n}$, $t = 1, \dots, n$.
                         Forward filtering (Kalman filtering)
Perform forward filtering part in Algorithm 7.3 to calculate $\mu_{t|t}$, $P_{t|t}$, $\mu_{t|t-1}$, and
$P_{t|t-1}$, $t = 1, \dots, n$.
                         Backward sampling
for $t = n, \dots, 1$ do
    If $t = n$, sample $X_n \sim \mathcal{N}(\mu_{n|n}, P_{n|n})$; else, given $X_{t+1} = x_{t+1}$, calculate
    $\Gamma_{t|t+1} = P_{t|t} A^T P_{t+1|t}^{-1}$,
        $P_{t|t+1}^x = P_{t|t} - \Gamma_{t|t+1} A P_{t|t}$,
        $\mu_{t|t+1}^x = \mu_{t|t} + \Gamma_{t|t+1}(x_{t+1} - \mu_{t+1|t})$,
    and sample $X_t \sim \mathcal{N}(\mu_{t|t+1}^x, P_{t|t+1}^x)$.
```

7.3.4.1 Motivation for particle filters

One approximation to optimum filtering recursion is to use grid-based methods, where the continuous $\mathcal{X}$ is approximated by its finite discretised version and the update rules are used as in the case of finite state HMMs. Another approach is *extended Kalman filter* (Sorenson, 1985), which approximates a non-linear transition by a linear one and performs the Kalman filter afterwards. The method fails if the nonlinearity in the HMM is substantial An improved approach based on the Kalman filter is the *unscented Kalman filter* (Julier and Uhlmann, 1997), which is based on a deterministic selection of sigma-points from the support of the state distribution of interest such that the mean and the variance of the true distribution are preserved by the sample mean and covariance calculated at these selected sigma-points. All of these methods are deterministic and not capable of dealing with the most general state-space models; in particular they will fail when the dimensions or the nonlinearities increase.

Alternative to the deterministic approximation methods, Monte Carlo can provide a robust and efficient solution to the optimal filtering problem. SMC methods for optimal filtering, also known as *particle filters*, have been shown to produce more accurate estimates than the deterministic methods mentioned (Liu and Chen, 1998; Doucet et al., 2000b; Durbin and Koopman, 2000; Kitagawa, 1996). Some of the good tutorials on SMC methods for filtering as well as smoothing in HMMs are Doucet et al. (2000b); Arulampalam et al. (2002); Cappé et al. (2007); Fearnhead (2008); Doucet and Johansen (2009) from the earliest to the most recent. One can also see Doucet et al. (2001) as a reference book, although a bit outdated. Also, the book Del Moral (2004) contains a rigorous review of numerous theoretical aspects of the SMC methodology in a different framework where a SMC method is treated as an interacting particle system associated with the mean field interpretation of a Feynman-Kac flow.

7.3.4.2 Particle filtering for HMM

Equations (7.1) and (7.3) reveal that we can write $p(x_{1:t}|y_{1:t})$ in terms of $p(x_{1:t-1}|y_{1:t-1})$ as

$$p(x_{1:t}|y_{1:t}) = \frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|y_{1:t-1})}p(x_{1:t-1}|y_{1:t-1}). \tag{7.24}$$

The normalising constant $p(y_t|y_{1:t-1})$ can be written in terms of the known densities as

$$p(y_t|y_{1:t-1}) = \int p(x_{1:t-1}|y_{1:t-1})f(x_t|x_{t-1})g(y_t|x_t)dx_{1:t} \tag{7.25}$$

where by convention $p(y_1|y_0) := p(y_1) = \int g(y_1|x_1)\eta(x_1)dx_1$. The recursion in (7.24) is essential since it enables efficient sequential approximation of the distributions $p(x_{1:t}|y_{1:t})$ as we will see shortly.

With reference to the Monte Carlo methodology covered in Sections 6.2 and 6.3, the filtering problem in state space models can be considered as a sequential inference problem for the sequence of probability distributions

$$\pi_n(x_{1:n}) := p(x_{1:n}|y_{1:n}), \quad n \geq 1.$$

As we saw in Sections 6.2 and 6.3, we can perform SIS and SISR methods targeting $\{\pi_n(x_{1:n})\}_{n\geq 1}$. The SMC proposal density at time n, denoted as $q_n(x_{1:n}|y_{1:n})$, is designed conditional to the observations up to time n and state values up to time $n-1$; and in the most general case it can be written as

$$\begin{aligned} q_n(x_{1:n}|y_{1:n}) &:= q(x_1|y_1)\prod_{t=2}^{n} q(x_t|x_{1:t-1}, y_{1:t}) \\ &= q_{n-1}(x_{1:n-1}|y_{1:n-1})q(x_n|x_{1:n-1}, y_{1:n}) \end{aligned} \tag{7.26}$$

In fact, most of the time the transition densities $q(x_t|x_{1:t-1}, y_{1:t})$ only depends only on the current observation y_t and the previous state x_{t-1},

$$q(x_n|x_{1:n-1}, y_{1:n}) = q(x_n|x_{n-1}, y_n).$$

Therefore, we can write

$$q_n(x_{1:n}|y_{1:n}) = q(x_1|y_1)\prod_{t=2}^{n} q(x_t|x_{t-1}, y_t) \tag{7.27}$$

If we wanted to perform SMC using the target distribution π_n directly, then we would have to calculate the following incremental weight at time n

$$\begin{aligned}\frac{\pi_n(x_{1:n})}{\pi_{n-1}(x_{1:n-1})q(x_n|x_{n-1},y_n)} &= \frac{p(x_{1:n}|y_{1:n})}{p(x_{1:n-1}|y_{1:n-1})q(x_n|x_{n-1},y_n)}\\ &= \frac{p(x_{1:n},y_{1:n})}{p(x_{1:n-1},y_{1:n-1})q(x_n|x_{n-1},y_n)}\frac{p(y_{1:n-1})}{p(y_{1:n})}\\ &= \frac{f(x_n|x_{n-1})g(y_n|x_n)}{q(x_n|x_{n-1},y_n)}\frac{1}{p(y_n|y_{1:n-1})}\\ &\propto \frac{f(x_n|x_{n-1})g(y_n|x_n)}{q(x_n|x_{n-1},y_n)}. \end{aligned} \tag{7.28}$$

In most of the applications $p(y_n|y_{1:n-1})$ can not be calculated, hence the ratio above is not available. For this reason, instead of $\pi_n(x_{1:n})$ SMC methods use the joint density of $X_{1:n}$ and $Y_{1:n}$ as the unnormalised measure for importance sampling

$$\widehat{\pi}_n(x_{1:n}) = p(x_{1:n}, y_{1:n}),$$

where the normalising constant is $p(y_{1:n})$, the likelihood of observations up to time n. Define the incremental importance weight

$$w_{n|n-1}(x_{n-1},x_n) = \frac{f(x_n|x_{n-1})g(y_n|x_n)}{q(x_n|x_{n-1},y_n)}.$$

The importance weight for the whole path $X_{1:n}$ is given by

$$w_n(x_{1:n}) = w_{n-1}(x_{1:n-1})w_{n|n-1}(x_{n-1},x_n),$$

We present the SIS algorithm in Algorithm 7.5 and SISR algorithm (or the particle filter) for general state-space models in Algorithm 7.6, reminding that SIS is a special type of SISR where there is no resampling. As in the general SISR algorithm, we can use an optional resampling scheme, where we do resampling only when the estimated effective sampling size decreases below a threshold value. In the following we list some of the aspects of the particle filter.

7.3.4.3 Filtering, prediction, and smoothing densities

Although the particle filter we presented in Algorithm 7.6 targets the path filtering distributions $\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n})$; it can easily be modified, or used directly, to make inference on other distributions that might be of interest. For example, consider the one step path prediction distribution

$$\pi_n^p(x_{1:n}) = p(x_{1:n}|y_{1:n-1}).$$

There is the following relation between π_n and π_n^p.

$$\pi_n^p(x_{1:n}) = \pi_{n-1}(x_{1:n-1})f(x_n|x_{n-1}), \quad \pi_n(x_{1:n}) = \pi_n^p(x_{1:n})\frac{g(y_n|x_n)}{p(y_n|y_{1:n-1})}.$$

Algorithm 7.5: SIS for HMM

```
for $n = 1, 2, \ldots$ do
    for $i = 1, \ldots, N$ do
        if $n = 1$ then
            sample $X_1^{(i)} \sim q(\cdot)$, calculate $w_1(X_1^{(i)}) = \frac{\eta(X_1^{(i)})g(y_1|X_1^{(i)})}{q(X_1^{(i)}|y_1)}$.
        else
            if $n \geq 2$ sample $X_n^{(i)} \sim q(\cdot|X_{1:n-1}^{(i)})$, set $X_{1:n}^{(i)} = (X_{1:n-1}^{(i)}, X_n^{(i)})$, and calculate
                $$w_n(X_{1:n}^{(i)}) = w_{n-1}(X_{1:n-1}^{(i)}) \frac{f(X_n^{(i)}|X_{n-1}^{(i)})g(y_n|X_n^{(i)})}{q(X_n^{(i)}|X_{n-1}^{(i)}, y_n)}.$$
    for $i = 1, \ldots, N$ do
        Calculate
            $$W_n^{(i)} = \frac{w_n(X_{1:n}^{(i)})}{\sum_{i=1}^N w_n(X_{1:n}^{(i)})}.$$
```

Algorithm 7.6: SISR (Particle filter) for HMM

```
if $n = 1$ then
    for $i = 1, \ldots, N$ do
        sample $X_1^{(i)} \sim q_1(\cdot)$
    for $i = 1, \ldots, N$ do
        Calculate
            $$W_1^{(i)} \propto \frac{\eta(X_1^{(i)})g(y_1|X_1^{(i)})}{q(X_1^{(i)}|y_1)}.$$
else
    Resample from $\{X_{1:n-1}^{(i)}\}_{1\leq i\leq N}$ according to the weights $\{W_{n-1}^{(i)}\}_{1\leq i\leq N}$ to get
    resampled particles $\{\widetilde{X}_{1:n-1}^{(i)}\}_{1\leq i\leq N}$ with weight $1/N$.
    for $i = 1, \ldots, N$ do
        Sample $X_n^{(i)} \sim q(\cdot|\widetilde{X}_{1:n-1}^{(i)}, y_n)$, set $X_{1:n}^{(i)} = (\widetilde{X}_{1:n-1}^{(i)}, X_n^{(i)})$
    for $i = 1, \ldots, N$ do
        Calculate
            $$W_n^{(i)} \propto \frac{f(X_n^{(i)}|\widetilde{X}_{n-1}^{(i)})g(y_n|X_n^{(i)})}{q(X_n^{(i)}|\widetilde{X}_{n-1}^{(i)}, y_n)}.$$
```

Therefore, it is easy to derive approximations to these distributions from each other: obtaining $\pi_n^{p,N}$ from π_{n-1}^N requires a simple extension of the path $X_{1:n-1}$ to $X_{1:n}$ through f; this is done by sampling $X_n^{(i)}$ conditioned on the existing particles paths $X_{1:n-1}^{(i)}$, respectively for $i = 1, \dots, N$. Whereas; obtaining π_n^N from $\pi_n^{p,N}$ requires a simple re-weighting of the particles according to $g(y_n|x_n)$.

As a second example, the approximations to the marginal distributions $\pi_n^N(x_k)$ are simply obtained from the k'th components of the particles, e.g.

$$\pi_n^N(x_{1:n}) = \sum_{i=1}^N W_n^{(i)} \delta_{X_{1:n}^{(i)}}(x_{1:n}) \Rightarrow \pi_n^N(x_k) = \sum_{i=1}^N W_n^{(i)} \delta_{X_{1:n}^{(i)}(k)}(x_k).$$

Note that the optimal filtering problem corresponds to the case $k = n$. Therefore, it may be sufficient to have a good approximation for the marginal posterior distribution of the current state X_n rather than the whole path $X_{1:n}$. This justifies the resampling step of the particle filter in practice, since resampling trades off accuracy for states X_k with $k \ll n$ for a good approximation for the marginal posterior distribution of X_n.

7.3.4.4 Estimating the evidence

A by-product of the particle filter is that it can provide unbiased estimates for unknown normalising constants of the target distribution. For example, when SISR is used, an unbiased estimator of $p(y_{1:n})$ can be obtained as

$$p(y_{1:n}) \approx p^N(y_{1:n}) = \prod_{t=1}^n \frac{1}{N} \sum_{i=1}^N w_{t|t-1}(\widetilde{X}_{t-1}^{(i)}, X_t^{(i)}).$$

7.3.4.5 Choice of the importance density

The choice of the kernel q for the importnce distribution in the particle filter is important to ensure effective SMC approximation. The first genuine particle filter in the literature, proposed by Gordon et al. (1993), involved sampling from the transition density of $X_{1:n}$, hence taking

$$q(x_n|x_{n-1}, y_n) = f(x_n|x_{n-1})$$

and the resulting particle filter with this particular choice of q is called the *bootstrap filter*. With this choice, the particles are weighted by how they fit to the observation, i.e. by the observation density,

$$w_{n|n-1}(x_{n-1}, x_n) = \frac{f(x_n|x_{n-1})g(y_n|x_n)}{f(x_n|x_{n-1})} = g(y_n|x_n).$$

The optimal choice that minimises the variance of the incremental importance weights is, from equation (6.5),

$$q^{opt}(x_n|x_{n-1}, y_n) = p(x_n|x_{n-1}, y_n).$$

This results in the optimal incremental weights to be

$$w^{opt}_{n|n-1}(x_{n-1}, x_n) = p(y_n|x_{n-1}),$$

which is independent from the value of x_n. First works where q^{opt} was used include Kong et al. (1994); Liu and Chen (1995); Liu (1996).

Another interesting choice is to take $q(x_n|x_{n-1}, y_n) = q(x_n|y_n)$, which can be useful when observations provide significant information about the hidden state but the state dynamics are weak. This proposal was introduced in Lin et al. (2005) and the resulting particle filter was called *independent particle filter.*

Example 7.4 (**Linear Gaussian HMM**)**.** *This is an illustrative example that is designed to show both SIS and SISR (particle filter) algorithms applied to sequential Bayesian inference in the following linear Gaussian HMM*

$$\eta(x) = \phi(x; 0, \sigma_0^2), \quad f(x'|x) = \phi(x'; ax, \sigma_x^2), \quad g(y|x) = \phi(y; bx, \sigma_y^2).$$

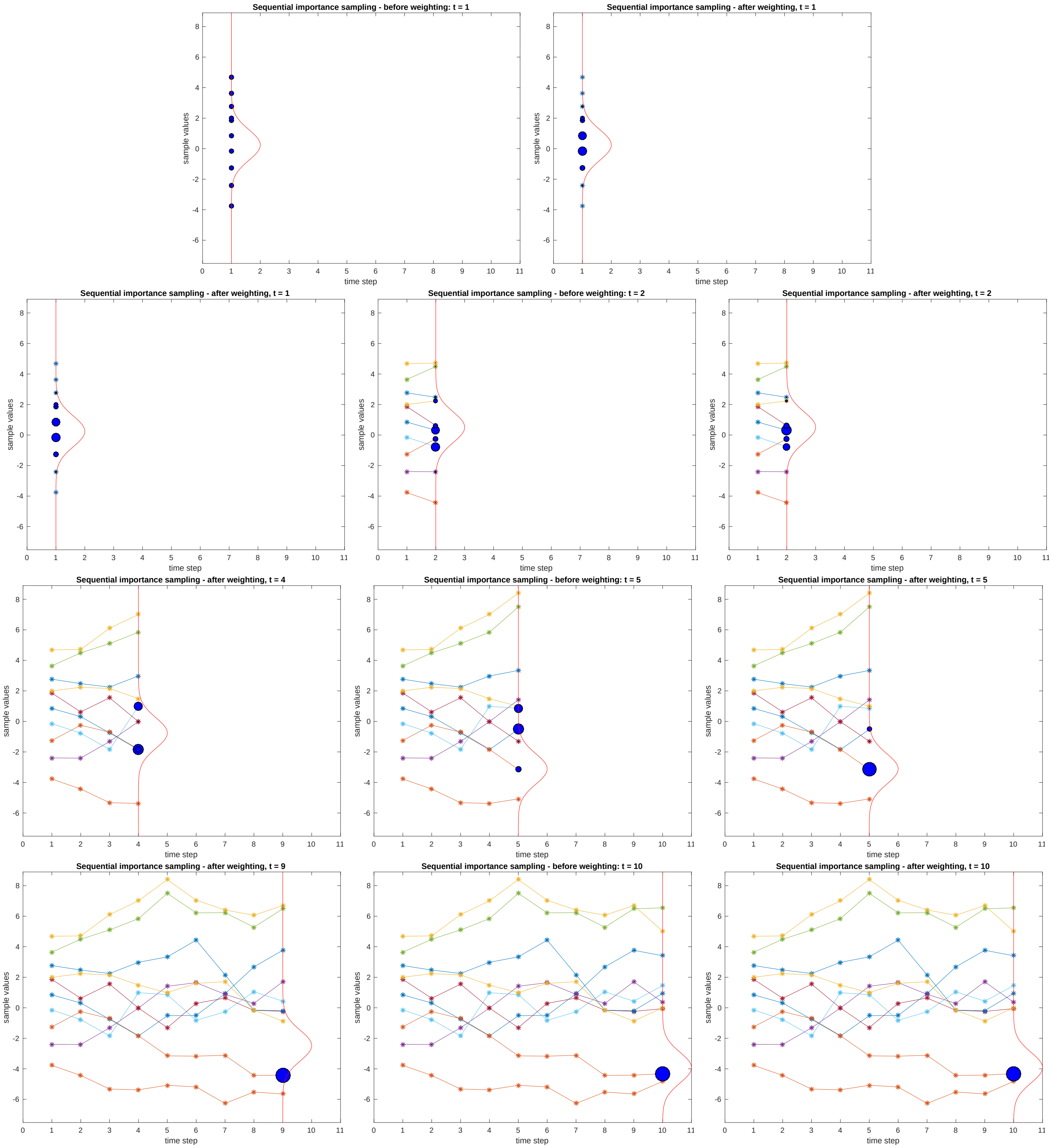
where $X_t \in \mathbb{R}$ and $Y_t \in \mathbb{R}$ and hence a, b, σ_0^2, σ_x^2, and σ_y^2 are all scalars.

In this example, we first generated $y_{1:n}$ with $n = 10$ using $a = 0.99$, $b = 1$, $\sigma_x^2 = 1$, $\sigma_y^2 = 1$ and $\sigma_0^2 = 4$. Our task is to run and compare and contrast the SMC algorithms, namely SIS and SISR, for sequential approximation of

$$\pi_1(x_1) = p(x_1|y_1), \dots, \pi_{10}(x_{1:10}) = p(x_{1:10}|y_{1:10})$$

Since the HMM here is linear and Gaussian, the problem is analytically tractable, π_t's are all Gaussian, and we can find those π_t's without any need to do Monte Carlo, for example using the Kalman filter. We use SIS and SISR merely for illustrative purposes.

We ran SIS in Algorithm 7.5 with $q_1(x_1) = \eta(x_1)$ and $q(x_n|x_{n-1}, y_n) = f(x_n|x_{n-1})$ for $n > 1$ so that $w_{n|n-1}(x_{n-1}, x_n) = g(y_n|x_n)$, which does not depend on x_{n-1}, and hence $W_n \propto g(y_n|X_n^{(i)})$ for all $n \geq 1$. Top row of Figure 7.2 shows the initialisation phase, both before and after weighting the initially generated particles $X_1^{(1)}, \dots, X_1^{(N)}$ whose locations are shown on the y-axis and weights are represented with the sizes of the balls centred around their values. The red curve represents the incremental weight function $w_1(x_1) = g(y_1|x_1)$ vs x_1 (located on the y-axis). Some of the later steps of SIS are shown Figure 7.2, from the second row on. Starting from the second row, each row shows (i) the particles and their weights from the previous time, (ii) The propagation and extension of the particles for the new time step, and (iii) Update of the particle weights. Note that, due to out particular choice of the importance density $q(x_t|x_{t-1}, y_t)$, the incremental weights $w_{t|t-1}(x_{t-1}, x_t) = g(y_t|x_t)$ depend only on the current value x_t, so for this example it is actually possible to show it as a function of x_t, which we have done by the red curve in the plots. Note that the size of the ball around the value of a particle represents the weight of the whole *path $X_{1:t}^{(i)}$. Also, notice the weight degeneracy problem in the SIS algorithm, since there is no resampling procedure. At time $t = 10$, we have effectively only one useful particle to approximate $\pi_{10}(x_{1:10}) = p(x_{1:10}|y_{1:10})$, which is not a good sign.*


Sequential importance sampling - before weighting: t = 1

Sequential importance sampling - after weighting, t = 1

Sequential importance sampling - after weighting, t = 1

Sequential importance sampling - before weighting: t = 2

Sequential importance sampling - after weighting, t = 2

Sequential importance sampling - after weighting, t = 4

Sequential importance sampling - before weighting: t = 5

Sequential importance sampling - after weighting, t = 5

Sequential importance sampling - after weighting, t = 9

Sequential importance sampling - before weighting: t = 10

Sequential importance sampling - after weighting, t = 10

Figure 7.2: SIS: propagation and weighting of particles for several time steps. Each row shows (i) the particles and their weights from the previous time, (ii) The propagation and extension of the particles for the new time step, and (iii) Update of the particle weights. Notice the weight degeneracy problem.

Next, the SISR algorithm, or the particle filter, in Algorithm 7.6 for the same problem. The initialisation is just the same as in SIS, see the top row of Figure 7.3. The remaining plots in Figure 7.3 shows some later steps of SISR. Starting from the second row, each row shows (i) the particles and their weights from the previous time, (ii) the resampling and propagation and extension of the particles for the new time step, and (iii) Update of the particle weights. Notice how the weight degeneracy problem is alleviated by resampling: there are distinct particles having closer weights to each other than in SIS. But several resampling steps in succession lead the path degeneracy problem: $\pi_{10}^N(x_{1:10})$ *has only one support point for* x_1*.*

Example 7.5 (**Tracking a moving target**). *We consider the HMM for a moving target in Example 7.3. Our objective is to estimate the position of the target at times* $t = 1, 2, \ldots$ *given the observations up to time* t*. With the particle filter, we can approximate* $\pi_t(x_{1:t}) = p(x_{1:t}|y_{1:t})$ *as*

$$\pi_t^N(x_{1:t}) = \sum_{i=1}^{N} W_t^{(i)} \delta_{X_{1:t}^{(i)}}(x_{1:t})$$

As discussed already, this approximation can be used to approximate the filtering distribution

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{N} W_t^{(i)} \delta_{X_t^{(i)}}(x_t)$$

We can use the position components of $X_t^{(i)} = (V_t^{(i)}, P_t^{(i)})$*'s in order to estimate the current position from the observations.*

$$\mathbb{E}[P_t(j)|Y_{1:t} = y_{1:t}] \approx \hat{P}_t^N(j) = \sum_{i=1}^{N} W_t^{(i)} P_t^{(i)}, \quad j = 1, 2.$$

Figure 7.4 illustrates a target tracking scenario depicted above for 500 *time steps. On the top left plot we see the position (in red) and its filtering estimate* $\hat{P}_t^N$ *(in black) given the sensor measurements on the right plot up to the current time* t*, with* $N = 1000$ *particles. The lower plots show the performance of the particle filter at each direction separately.*

7.3.5 Extensions to HMMs

Although HMMs are the most common class of time series models in the literature, there are also many time series models which are not a HMM and are still of great importance. These models differ from HMMs mostly because they do not possess the conditional independency of observations. Here, we give two examples.

- In the first example of such models, the process $\{X_n\}_{n \geq 1}$ is still a Markov chain; however the conditional distribution of Y_n, given all past variables $X_{1:n}$ and $Y_{1:n-1}$, depends not only on the value of X_n but also on the values of past observations

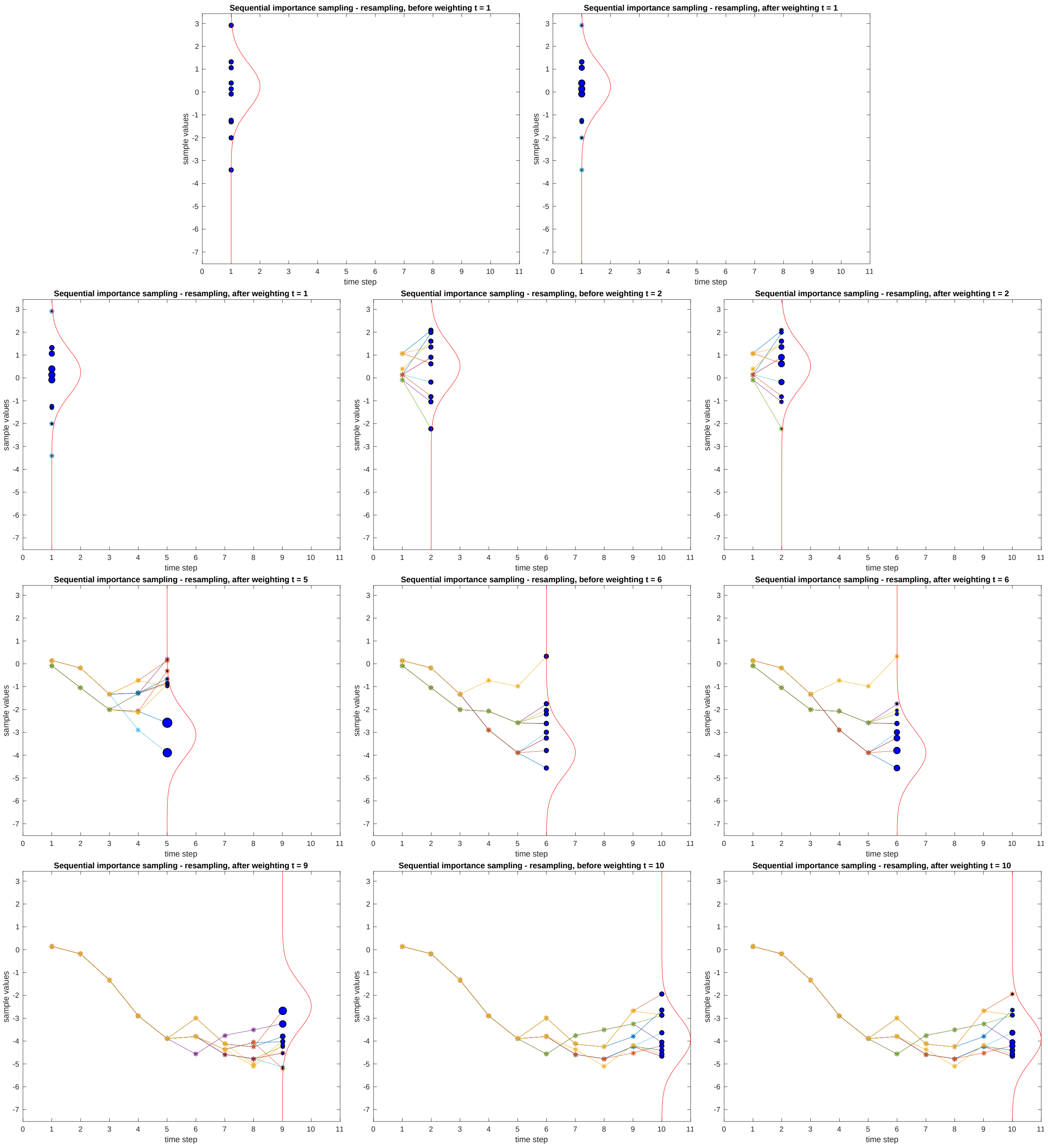


Figure 7.3: SIRS: resampling, propagation and weighting of particles for several time steps. Each row shows (i) the particles and their weights from the previous time, (ii) the resampling and propagation and extension of the particles for the new time step, and (iii) Update of the particle weights. Notice the path degeneracy problem: $\pi_{10}^N(x_{1:10})$ has only one support point for x_1.

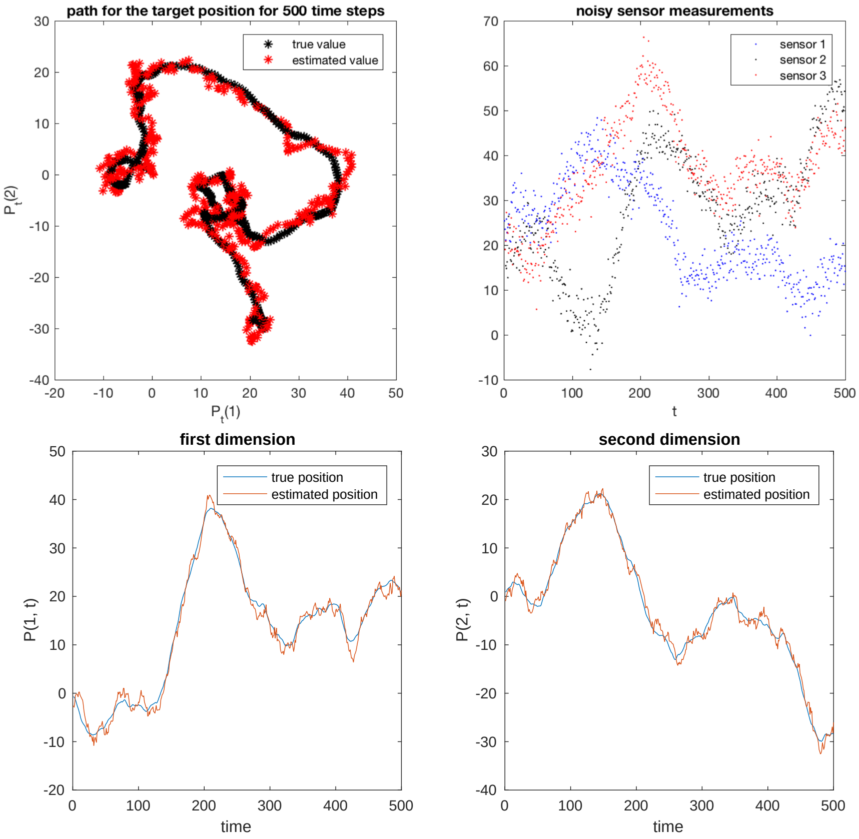


Figure 7.4: Particle filter for target tracking

i.e. $Y_{1:n-1}$. If we denote the probability density of this conditional distribution $g_n(y_n|x_n, y_{1:n-1})$, the joint probability density of $(X_{1:n}, Y_{1:n})$ is

$$p(x_{1:n}, y_{1:n}) = \eta(x_1)g(y_1|x_1)\prod_{t=2}^{n} f(x_t|x_{t-1})g_t(y_t|x_t, y_{1:t-1}).$$

If Y_n given X_n is independent of the past values of the observations prior to time $n-k$, then we can define $g_n(y_n|x_n, y_{1:n-1}) = g(y_n|x_n, y_{n-k:n-1})$ for all n.

These models have much in common with basic HMMs in the sense that virtually identical computational tools may be used for both models. In the particular context of SMC, the similarity between these two types of models is more clearly exposed in Del Moral (2004) via the Feynman-Kac representation of SMC methods, where the conditional density of observation at time n is treated generally as a *potential function* of x_n.

- In another type of time series models that are not HMM the latent process $\{X_n\}_{n\geq 1}$ is, again, still a Markov chain; however observation at current time depends on all the past values, i.e. Y_n conditional on $(X_{1:n}, Y_{1:n-1})$ depends on all of these conditioned random variables. Actually, these models are usually the result of marginalising an extended HMM. Consider the HMM $\{(X_n, Z_n), Y_n\}_{n\geq 1}$, where the latent joint process $\{X_n, Z_n\}_{n\geq 1}$ is a Markov chain such that its transitional density can be factorised as

$$f(x_n, z_n|x_{n-1}, z_{n-1}) = f_1(x_n|x_{n-1})f_2(z_n|x_n, z_{n-1}).$$

and the observation Y_n depends only on X_n and Z_n given all the past random variables and admits the probability density $g(y_n|x_n, z_n)$. Now, the reduced bivariate process $\{X_n, Y_n\}_{n\geq 1}$ is not a HMM and we express the joint density of $(X_{1:n}, Y_{1:n})$ as

$$p(x_{1:n}, y_{1:n}) = \eta(x_1)p_1(y_1|x_1)\prod_{t=2}^{n} f_1(x_t|x_{t-1})p_t(y_t|x_{1:t}, y_{1:t-1})$$

where the density $p_t(y_t|x_{1:t}, y_{1:t-1})$ is given by

$$p_t(y_t|x_{1:t}, y_{1:t-1}) = \int p(z_{1:t-1}|x_{1:t-1}, y_{1:t-1})f_2(z_t|x_t, z_{t-1})g(y_t|x_t, z_t)dz_{1:t}. \tag{7.29}$$

The reason $\{X_n, Y_n\}_{n\geq 1}$ might be of interest is that the conditional laws of $Z_{1:n}$ may be available in close form and exact evaluation of the integral in (7.29) is available. In that case, it can be more effective to perform Monte Carlo approximation for the law of $X_{1:n}$ given observations $Y_{1:n}$, which leads to the so called *Rao-Blackwellised particle filters* in the literature (Doucet et al., 2000a).

The integration is indeed available in close form for some time series models. One example is the *linear Gaussian switching state space models* (Chen and Liu, 2000; Doucet et al., 2000a; Fearnhead and Clifford, 2003), where X_n takes values on a finite set whose elements are often called 'labels', and conditioned on $\{X_n\}_{n\geq 1}$, $\{Z_n, Y_n\}_{n\geq 1}$ is a linear Gaussian state-space model.

We note that the computational tools developed for HMMs are generally applicable to a more general class of time series models with some suitable modifications.

7.3.6 The Rao-Blackwellised particle filter

Assume we are given a HMM $\{(X_n, Z_n), Y_n\}_{n\geq 1}$ where this time the hidden state at time n is composed of two components X_n and Z_n. Suppose that the initial and transition distributions of the Markov chain $\{X_n, Z_n\}_{n\geq 1}$ have densities η and f and they can be factorised as follows

$$\eta(x_1, z_1) = \eta_1(x_1)\eta_2(z_1|x_1), \quad f(x_n, z_n|x_{n-1}, z_{n-1}) = f_1(x_n|x_{n-1})f_2(z_n|x_n, z_{n-1}).$$

Also, conditioned on (x_n, z_n) the distribution of observation Y_n admit a density $g(y_n|x_n, z_n)$ with respect to ν. We are interested in the case where the posterior distribution

$$\pi_n(x_{1:n}, z_{1:n}) = p(x_{1:n}, z_{1:n}|y_{1:n})$$

is analytically intractable and we are interested in approximating the expectations

$$\pi_n(\varphi_n) = \mathbb{E}\left[\varphi_n(X_{1:n}, Z_{1:n}) | Y_{1:n} = y_{1:n}\right]$$

for functions $\varphi_n : \mathcal{X}^n \times \mathcal{Z}^n \to \mathbb{R}$. Obviously, one way to do this is to run an SMC filter for $\{\pi_n\}_{n\geq 1}$ which obtains the approximation π_n^N at time n as

$$\pi_n^N(x_{1:n}, z_{1:n}) = \sum_{i=1}^N W_n^{(i)} \delta_{(X_{1:n}^{(i)}, Z_{1:n}^{(i)})}(x_{1:n}, z_{1:n}), \quad \sum_{i=1}^N W_n^{(i)} = 1.$$

However, if the conditional posterior probability distribution

$$\pi_{2,n}(z_{1:n}|x_{1:n}) = p(z_{1:n}|x_{1:n}, y_{1:n})$$

is analytically tractable, there is a better SMC scheme for approximating π_n and estimating $\pi_n(\varphi_n)$. This SMC scheme is called the *Rao Blackwellised particle filter* (RBPF) (Doucet et al., 2000a). Consider the following decomposition which follows from the chain rule

$$p(x_{1:n}, z_{1:n}|y_{1:n}) = p(x_{1:n}|y_{1:n})p(z_{1:n}|x_{1:n}, y_{1:n})$$

and define the marginal posterior distribution of $X_{1:n}$ conditioned on $y_{1:n}$ as

$$\pi_{1,n}(x_{1:n}) = p_1(x_{1:n}|y_{1:n}).$$

The RBPF is a particle filter for the sequence of marginal distributions $\{\pi_{1,n}\}_{n\geq 1}$ which produces at time n the approximation

$$\pi_{1,n}^N(x_{1:n}) = \sum_{i=1}^N W_{1,n}^{(i)} \delta_{X_{1:n}^{(i)}}(x_{1:n}), \quad \sum_{i=1}^N W_{1,n}^{(i)} = 1.$$

and the Rao-Blackwellised approximation the full posterior distribution involves the particle filter estimate $\pi_{1,n}^N$ and the exact distribution $\pi_{2,n}$

$$\pi_n^{\mathrm{RB},N}(x_{1:n}, z_{1:n}) = \pi_{1,n}^N(x_{1:n})\pi_{2,n}(z_{1:n}|x_{1:n}).$$

Then, the estimator of the the RBPF for $\pi_n(\varphi_n)$ becomes

$$\begin{aligned}\pi_n^{\mathrm{RB},N}(\varphi_n) &= \mathbb{E}_{\pi_{1,n}^N}\left[\mathbb{E}_{\pi_{2,n}(\cdot|X_{1:n}^{(i)})}\left[\varphi_n(X_{1:n}^{(i)}, Z_{1:n})\right]\right] \\ &= \sum_{i=1}^N W_{1,n}^{(i)}\left[\int \pi_{2,n}(z_{1:n}|X_{1:n}^{(i)})\varphi_n(X_{1:n}^{(i)}, z_{1:n})dz_{1:n}\right]. \end{aligned} \tag{7.30}$$

Assuming $q(x_{1:n}|y_{1:n}) = q(x_{1:n-1}|y_{1:n-1})q(x_n|x_{1:n-1}, y_{1:n})$ is used as the proposal distribution, the incremental importance weight for the RBPF is given by

$$w_{1,n|n-1}(x_{1:n}) = \frac{f_1(x_n|x_{n-1})p(y_n|x_{1:n}, y_{1:n-1})}{q(x_n|x_{1:n-1}, y_{1:n})}$$

where the density $p(y_n|x_{1:n}, y_{1:n-1})$ is given by

$$p_n(y_n|x_{1:n}, y_{1:n-1}) = \int p(z_{1:n-1}|x_{1:n-1}, y_{1:n-1}) f_2(z_n|x_n, z_{n-1}) g(y_n|x_n, z_n) dz_{1:n}.$$

Also, the optimum importance density which reduces the variance of $w_{1,n|n-1}$ is when the incremental importance density $q(x_n|x_{1:n-1}, y_{1:n})$ is taken to be $p(x_n|x_{1:n-1}, y_{1:n})$ which results in $w_{1,n|n-1}(x_{1:n})$ being equal to $p(y_n|x_{1:n-1}, y_{1:n-1})$.

The use of the RBPF whenever it is possible is intuitively justified by the fact that we substitute particle approximation of some expectations with their exact values. Indeed, the theoretical analysis in Doucet et al. (2000a) and Chopin (2004, Proposition 3) revealed that the RBPF has better precision than the regular particle filter: the estimates of the RBPF never have larger variances. The favouring results for the RBPF are basically due to the Rao-Blackwell theorem (see e.g. Blackwell (1947)), after which the proposed particle filter gets its name.

The RBPF was formulated by Doucet et al. (2000a) and have been implemented in various settings by Chen and Liu (2000); Andrieu and Doucet (2002); Särkkä et al. (2004) among many.

Exercises

1. Write your own `log_sum_exp` function that takes an array of numbers $a = [a_1, \ldots, a_m]$ and returns
$$\log\left(e^{a_1} + \ldots + e^{a_m}\right).$$
in a numerically safe way. Try your code with
$$a = [100, 500, 1000], \quad a = [-1200, -1500, -1200], a = [-1000, 1000]$$
Compare each answer with the naive solution which we would have if we typed directly `log(sum(exp(a)))`.

   [Hint: $\log(e^a - e^b) = \log(e^{a-\max\{a,b\}} - e^{b-\max\{a,b\}}) + \max\{a, b\}$.]

2. Consider the linear Gaussian HMM in Example 7.4.

   - Generate hidden states $x_{1:n}$ and observations $y_{1:n}$ for $n = 2000$, $a = 0.99$, $b = 1$, $\sigma_0^2 = 5$, $\sigma_x^2 = 1$, $\sigma_y^2 = 4$.
   - You already have the code for this model that performs the particle filter i.e. the SISR algorithm with the importance (proposal) density being equal to the transition density, more explicitly
$$q_1(x_1) = \eta(x_1), \quad q_n(x_{1:n}) = q_{n-1}(x_{1:n-1}) f(x_n|x_{n-1})$$
Run the SISR algorithm for $y_{1:n}$ with $N = 100$ this choice of importance density, and at each time step estimate the mean posterior value $\mathbb{E}[X_t|Y_{1:t} = y_{1:t}]$ from the particles.
$$\hat{X}_t^N = \sum_{i=1}^{N} X_t^{(i)} W_t^{(i)}.$$
Calculate the mean squared error (MSE) for $\hat{X}_t$, that is $\frac{1}{n}\sum_{t=1}^{n}(\hat{X}_t - x_t)^2$.
   - Now, set $N = 1000$ and calculate the MSE again. Compare it with the previous one.
   - This time take $N = 100$ and run the SISR algorithm with the optimum choice for the importance density which is obtained by
$$q(x_1) = p(x_1|y_1), \quad q(x_n|x_{n-1}, y_n) = p(x_n|x_{n-1}, y_n),$$
Show that this leads to the incremental weight $w_{n|n-1}(x_{n-1}, x_n) = p(y_n|x_{n-1})$. (Since this is a linear Gaussian model, both $p(x_n|x_{n-1}, y_n)$ and $p(y_n|x_{n-1})$ are available to sample from and calculate, respectively.) Calculate the MSE for $\hat{X}_t$ and compare it with the previous ones that you found.

3. Consider the target tracking problem in Example 7.5. It may not be a good modelling practice to take the noise in the measurements additive Gaussian for two reasons: (i) One may expect to have a bigger error when a longer distance is measured. (ii) When the noise is Gaussian, the noisy measurement is allowed to be negative, which does not make sense. That is why, instead of the existing observation model, consider the following one with multiplicative nonnegative noise:

$$Y_{t,i} = R_{t,i} E_{t,i}, \quad E_{t,i} \overset{\text{i.i.d.}}{\sim} \ln \mathcal{N}(0, \sigma_y^2), \quad i = 1, 2, 3. \tag{7.31}$$

where $\ln \mathcal{N}(\mu, \sigma^2)$ denotes the lognormal distribution with location and scale parameters μ and σ^2. It can be shown that if $E_{t,i} \sim \ln \mathcal{N}(\mu, \sigma^2)$, $\log E_{t,i} \sim \mathcal{N}(\mu, \sigma^2)$, so we effectively have

$$\log Y_{t,i} \sim \mathcal{N}(\log R_{t,i}, \sigma_y^2).$$

- Generate data according to the new model for $n = 500$ using $\sigma_y^2 = 0.1$, $a = 0.99$, $\sigma_p^2 = 0.001$, $\sigma_v^2 = 0.01$, $\sigma_{bv}^2 = 0.01$, $\sigma_{bp}^2 = 4$, and the sensor locations being the same as in the previous examples.
- Write down the new observation density $g(y_t|x_t)$ according to (7.31).
- Run a particle filter for the data you have generated with $N = 1000$ particles, using $q(x_1|y_1) = \eta(x_1)$ and $q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$. Calculate the posterior mean estimates for the position versus t, $\mathbb{E}[P_t(i)|Y_{1:t} = y_{1:t}]$, $i = 1, 2$. Generate results similar to the ones in Example 7.5.
- Remove one of the sensors and repeat your experiments. Comment on the results.

Appendix A

Some Basics of Probability

Summary: *This chapter provides some basics of probability which is related to the content of this course. The covered concepts are probability, random variables, cumulative distribution function, discrete and continuous distributions, probability mass function, probability density function, expectation, independence, correlation and covariance, Bayes' Theorem, and posterior distribution*

A.1 Axioms and properties of probability

Let Ω be the *sample space* and $\mathcal{F}$ be the *event space.* (In a non-rigorous way, you can think of $\mathcal{F}$ as the set of all subsets of Ω as an example.) A *probability measure* on $(\Omega, \mathcal{F})$ is a function $\mathbb{P} : \mathcal{F} \to \mathbb{R}$ that satisfies the following *axioms of probability.*

(A1) The probability of an event is a non-negative and real number:

$$\mathbb{P}(E) \in \mathbb{R}, \quad \mathbb{P}(E) \geq 0, \quad \forall E \in \mathcal{F}.$$

(A2) Unitarity: The probability that at least one of the elementary events in the entire sample space will occur is 1

$$\mathbb{P}(\Omega) = 1.$$

(A3) σ-additivity: A countable sequence of disjoint sets (or *mutually exclusive* sets) $E_1, E_2, \ldots$ ($E_i \cap E_j = \emptyset$ for all $i \neq j$) satisfies

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(E_i)$$

Any function that satisfies those three axioms can be a probability measure. These axioms lead to some useful properties of probability that we are familiar with.

(P1) The probability of the empty set:

$$\mathbb{P}(\emptyset) = 0.$$

(P2) Monotonicity:

$$\mathbb{P}(A) \leq \mathbb{P}(B), \quad \forall A, B \in \mathcal{F} : A \subseteq B.$$

(P3) The numeric bound:

$$0 \leq \mathbb{P}(E) \leq 1, \quad \forall E \in \mathcal{F}.$$

(P4) Union of two sets:

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B), \quad \forall A, B \in \mathcal{F}.$$

(P5) Completion of a set:

$$\mathbb{P}(A^c) = 1 - \mathbb{P}(A), \quad \forall A \in \mathcal{F}.$$

A.2 Random variables

Suppose we are given the triple $(\Omega, \mathcal{F}, \mathbb{P})$. A *real-valued* random variable is a function

$$X : \Omega \to \mathbb{R}$$

such that $\{\omega \in \Omega : X(w) \leq x\} \in \mathcal{F}$ for all $x \in \mathbb{R}$. We need this condition since we need the probability of this set in order to construct our cumulative distribution function.

Cumulative distribution function The probability distribution of X is mainly characterised by its cumulative distribution function (cdf) denoted as F, which is defined as

$$F(x) := \mathbb{P}(X \leq x) = \mathbb{P}(\{\omega \in \Omega : X(\omega) \leq x\}), \quad x \in \mathbb{R}.$$

There are three points to note here:

- The probability distribution of X is *induced* by $\mathbb{P}$: There is always an implicit reference to $(\Omega, \mathcal{F}, \mathbb{P})$ when one calculates $P(X \leq x)$, but we tend to forget it once we have out cumulative distribution function F for X. This is because once we know F, we know everything about the probability distribution of X and usually we do not need to go back to the lower level and work with $(\Omega, \mathcal{F}, \mathbb{P})$ in practice. However, it may be useful to know what a random variable is in general.
- The use of $\leq$ (and not $<$) is important. Especially for discrete random variables, this matters a lot.
- Note that X, written in capital letter, represents the randomness in the probability statement while x is a given certain value in $\mathbb{R}$.

By definition, F has the following properties:

(P1) F is a non-decreasing function: For any $a, b \in \mathbb{R}$, if $a < b$, then $F(a) \leq F(b)$.

(P2) F is right continuous (no jumps occur when the limit point is approached from the right).

(P3) $\lim_{x\to-\infty} F(x) = 0$.

(P4) $\lim_{x\to\infty} F(x) = 1$.

Any function that satisfies those four properties can be a cdf. Therefore, the definition and the properties have an if and only if relation.

All the probability questions about X can be answered in terms of F. Examples:

- $\mathbb{P}(X \in (a,b]) = \mathbb{P}(X \le b) - \mathbb{P}(X \le a) = F(b) - F(a)$
- $\mathbb{P}(X = a) = F(a) - \lim_{x\to a^-} F(x)$. (the second term is a limit from the left)
- $\mathbb{P}(X \in [a,b]) = \mathbb{P}(X \in (a,b]) + \mathbb{P}(X = a) = F(b) - F(a) + [F(a) - \lim_{x\to a^-} F(x)]$
- $\mathbb{P}(X \in (a,b)) = \mathbb{P}(X \in (a,b]) - \mathbb{P}(X = b) = F(b) - F(a) - [F(b) - \lim_{x\to b^-} F(x)]$

Depending on the nature of set of values X takes, it can be called a discrete or a continuous random variable (sometimes neither of them!).

A.2.1 Discrete random variables

If X takes finite or countably infinite number of possible values in $\mathbb{R}$, then X is called a discrete random variable. The possible values of X may be listed as $x_1, x_2, \ldots$, where the sequence terminates in the finite case but continues indefinitely in the countably infinite case.

Let $p(x_i) := \mathbb{P}(X = x_i)$, $i = 1, 2, \ldots$ The function $p(\cdot)$ is called the *probability mass function (pmf)* of X and has the following properties: $p(x_i) \ge 0, \quad i = 1, 2, \ldots$ and $\sum_i p(x_i) = 1$.

It can be shown that, for any $x \in \mathbb{R}$,

$$F(x) = \sum_{i:x_i \le x} p(x_i).$$

Hence, the cdf F of X is a step function where jumps occur at points x_i with jump height being $p(x_i) = \mathbb{P}(X = x_i) = F(x_i) - F(x_{i-1})$.

Some discrete distributions: Some well known distributions with a pmf (hence the cdf is a step function): Bernoulli $\mathcal{B}(\rho)$, Geometric distribution $\text{Geo}(\rho)$, Binomial distribution $\text{Binom}(n,\rho)$ Negative binomial $\text{NB}(r,\rho)$, Poisson distribution $\mathcal{PO}(\lambda)$.

A.2.2 Continuous random variables

If X takes values on a continuous subset R_X of $\mathbb{R}$ (such as $\mathbb{R}$ itself, an interval $[a,b]$ or union of such intervals), then X is said to be a continuous random variable. Furthermore, if F for X is continuous (i.e. no jumps), we have

$$\mathbb{P}(X \in (a,b)) = \mathbb{P}(X \in (a,b]) = \mathbb{P}(X \in [a,b)) = \mathbb{P}(X \in [a,b]) = F(b) - F(a).$$

Also, if F is right differentiable, we can define the *probability density function (pdf)* for X

$$p(x) := \lim_{h\to 0} \frac{F(x+h)-F(x)}{h} = \frac{\partial_+ F(x)}{\partial x}, \quad x \in \mathbb{R}.$$

Since F is monotonic, we have $p(x) \geq 0$ for all $x \in \mathbb{R}$. Also, p integrates to 1 i.e. $\int_{-\infty}^{\infty} p(x)dx = \int_{R_X} p(x)dx = 1$. All probability statements for X can be calculated using f, such as

$$\mathbb{P}(X \in [a,b]) = F(b) - F(a) = \int_a^b p(x)dx,$$
$$\mathbb{P}(X \leq a) = F(a) = \int_{-\infty}^a p(x)dx.$$

From the above equation, we can conclude that $\mathbb{P}(X = x) = 0$ for any $x \in \mathbb{R}$, because

$$\int_x^x p(x)dx = F(x) - F(x) = 0.$$

A.2.2.1 Some continuous distributions

The following are some well known distributions with a continuous cdf (hence admitting a pdf): Uniform distribution $\text{Unif}(a,b)$, exponential distribution $\text{Exp}(\mu)$, gamma distribution $\Gamma(\alpha,\beta)$, inverse gamma distribution $\mathcal{IG}(\alpha,k)$, normal (Gaussian) distribution $\mathcal{N}(\mu,\sigma^2)$, Beta distribution $\text{Beta}(\alpha,\beta)$.

A.2.3 Moments, expectation and variance

If X is a random variable, the n'th *moment* of X, $n \geq 1$, denoted by $\mathbb{E}(X^n)$, is defined for discrete and continuous random variables as follows:

$$\mathbb{E}(X^n) := \begin{cases} \sum_i x_i^n p(x_i), & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^{\infty} x^n p(x)dx, & \text{if } X \text{ is continuous.} \end{cases} \tag{A.1}$$

The first moment ($n = 1$) is called the *expectation* of X, also sometimes referred to as the mean of X.

If $|\mathbb{E}(X)| < \infty$, the n'th central moments of X, $n \geq 1$, is defined for discrete and continuous random variables as follows:

$$\mathbb{E}([X - \mathbb{E}(X)]^n) := \begin{cases} \sum_i [x_i - \mathbb{E}(X)]^n p(x_i), & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^{\infty} [x - \mathbb{E}(X)]^n p(x)dx, & \text{if } X \text{ is continuous.} \end{cases} \tag{A.2}$$

The second central moment is the most notable of them and is called the variance of X and denoted by $V(X)$:

$$\mathbb{V}(X) := \mathbb{E}([X - \mathbb{E}(X)]^2).$$

A useful identity relating $\mathbb{V}(X)$ to the expectation and the second moment of X is

$$\mathbb{V}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2.$$

Finally, the standard deviation of X is

$$\sigma_X := \sqrt{\mathbb{V}(X)}.$$

A.2.4 More than one random variables

Suppose we have two real valued random variables, $X : \Omega \to \mathbb{R}$ and $Y : \Omega \to \mathbb{R}$, both defined on the same proabability space $(\Omega, \mathcal{F}, \mathbb{P})$.[1] The joint distribution of X and Y is characterised by the joint cdf $F_{X,Y}$ which is defined as

$$F_{X,Y}(x, y) := \mathbb{P}(X \leq x, Y \leq y) = \mathbb{P}(\{\omega \in \Omega : X(\omega) \leq x, Y(\omega) \leq y\}).$$

The marginal cdf's for X and Y can be deduced from $F_{X,Y}(x, y)$:

$$F_X(x) = \lim_{y\to\infty} F_{X,Y}(x, y), \quad F_Y(y) = \lim_{x\to\infty} F_{X,Y}(x, y).$$

Discrete variables: For discrete X and Y taking values x_i, $i = 1, 2, \ldots$ and y_j, $j = 1, 2, \ldots$, we can define a joint pmf $p_{X,Y}$ for X and Y such that

$$p_{X,Y}(x_i, y_j) := \mathbb{P}(X = x_i, Y = y_j)$$

so that for any $x, y \in \mathbb{R}$, we have

$$F_{X,Y}(x, y) = \sum_{i,j : x_i \leq x, y_i \leq y} p_{X,Y}(x_i, y_j).$$

Expectation of any function g of X, Y can be evaluated using the joint pmf, for example

$$\mathbb{E}(g(X, Y)) = \sum_{i,j} p_{X,Y}(x_i, y_j) g(x_i, y_j).$$

The *marginal pmf*'s for X and Y are given as follows:

$$p_X(x_i) = \sum_j p_{X,Y}(x_i, y_j), \quad p_Y(y_j) = \sum_i p_{X,Y}(x_i, y_j),$$

[1] (X, Y) together can be called a bivariate random variable. A generalisation of this is a multivariate random variable of dimension m, such as $(X_1, X_2, \ldots, X_m)$.

Continuous variables: Similar to the joint pmf defined for discrete X and Y, one can define the joint pdf for continuous X and Y, assuming F is right-differentiable,

$$p_{X,Y}(x,y) := \frac{\partial_+^2 F(x,y)}{\partial x \partial y}$$

so that for any a, b, we have

$$F_{X,Y}(a,b) = \int_{-\infty}^{b} \int_{-\infty}^{a} p_{X,Y}(x,y) dx dy$$

Expectation of any function g of X, Y can be evaluated using the joint pdf,

$$\mathbb{E}(g(X,Y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_{X,Y}(x,y) g(x,y) dx dy.$$

The *marginal pdf*'s for X and Y can be obtained

$$p_X(x) = \int_{-\infty}^{\infty} p_{X,Y}(x,y) dy, \quad p_Y(y) = \int_{-\infty}^{\infty} p_{X,Y}(x,y) dx,$$

Independence: We say random variables X and Y are independent if for all pairs of sets $A \subseteq \mathbb{R}$, $B \subseteq \mathbb{R}$ we have

$$\mathbb{P}(X \in A, Y \in B) = \mathbb{P}(X \in A)\mathbb{P}(Y \in B).$$

If X and Y are discrete variables taking x_i, $i = 1, 2, \ldots$ and y_j, $j = 1, 2, \ldots$, then independence between X and Y can be expressed as

$$p_{X,Y}(x_i, y_j) = \mathbb{P}(X = x_i, Y = y_j) = p_X(x_i) p_Y(y_j), \quad \forall i, j$$

If X and Y are continuous variables, then independence between X and Y can be expressed as

$$p_{X,Y}(x,y) = p_X(x) p_Y(y), \quad \forall x, y \in \mathbb{R}.$$

Covariance and Correlation: Covariance between two random variables X and Y, $\mathrm{Cov}(X,Y)$ is given as

$$\begin{aligned} \mathrm{Cov}(X,Y) &:= \mathbb{E}([X - \mathbb{E}(X)][Y - \mathbb{E}(Y)]) \\ &= \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y) \end{aligned}$$

A normalised version of covariance is *correlation* $\rho(X,Y)$. Provided that $\mathbb{V}(X) \geq 0$ and $\mathbb{V}(Y) \geq 0$,

$$\rho(X,Y) := \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y};$$

When one of $\mathbb{V}(X)$ and $\mathbb{V}(Y)$ is 0, we set $\rho(X,Y) = 1$ if $X = Y$ and $\rho(X,Y) = 0$ if $X \neq Y$. One can show that

$$-1 \leq \rho(X,Y) \leq 1.$$

Absolute value of $\rho(X,Y)$ indicates the level of correlation. We say two random variables X, Y are uncorrelated if $\mathrm{Cov}(X,Y) = 0$ (hence $\rho(X,Y) = 0$).

Note: Independence implies uncorrelatedness, but the reverse is not always true.

A.3 Conditional probability and Bayes' rule

Consider the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ again. Given two sets $A, B \in \mathcal{F}$, the conditional distribution of A given B is denoted by $\mathbb{P}(A|B)$ and is defined as

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

The Bayes' rule is derived from this definition and it relates the two conditional probabilities $\mathbb{P}(A|B)$ and $\mathbb{P}(B|A)$:

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A)\mathbb{P}(B|A)}{\mathbb{P}(B)} \tag{A.3}$$

This relation can be written in terms of two random variables. Suppose X, Y are discrete random variables with joint pmf $p_{X,Y}(x_i, y_j)$, where $x \in \mathcal{X} = \{x_1, x_2, \ldots\}$ and $y \in \mathcal{Y} = \{y_1, y_2, \ldots\}$ so that the marginal pmf's are

$$p_X(x) = \sum_{y \in \mathcal{Y}} p_{X,Y}(x, y), \quad p_Y(y) = \sum_{x} p_{X,Y}(x, y), \quad x \in \mathcal{X}, y \in \mathcal{Y}.$$

Then the conditional pmf's $p_{X|Y}(x|y)$ and $p_{Y|X}(y|x)$ are defined as

$$p_{X|Y}(x|y) = \frac{p_{X,Y}(x, y)}{p_Y(y)}, \quad p_{Y|X}(y|x) = \frac{p_{X,Y}(x, y)}{p_X(x)} \tag{A.4}$$

and Bayes' rule relating them together is

$$p_{X|Y}(x|y) = \frac{p_X(x)p_{Y|X}(y|x)}{p_Y(y)} \tag{A.5}$$

When X, Y are continuous random variables taking values from $\mathcal{X}$ and $\mathcal{Y}$, respectively, with a joint pdf $p_{X,Y}(x, y)$, similar definitions follow: The marginal pdf's are

$$p_X(x) = \int_{-\infty}^{\infty} p_{X,Y}(x, y)dy, \quad p_Y(y) = \int_{-\infty}^{\infty} p_{X,Y}(x, y)dx.$$

The conditional pdf's are defined exactly the same way as in (A.4) and (A.5).

Appendix B

Solutions

B.1 Exercises in Chapter 2

1. We have $g : (0,1) \to (a,b)$ with $x = g(u) := (b-a)u + a$, hence $g^{-1} : (a,b) \to (0,1)$ with
$$u = g^{-1}(x) = (x-a)/(b-a), \quad x \in (a,b)$$
The Jacobian is $J(x) = \frac{\partial g^{-1}(x)}{\partial x} = \frac{1}{b-a}$ for all x. We apply the formula in (2.5) for transformation from U to X to have
$$\begin{aligned} p_X(x) &= p_U(g^{-1}(x))\,|J(x)| \\ &= p_U\left(\frac{x-a}{b-a}\right)\frac{1}{b-a} \\ &= \frac{1}{b-a} \end{aligned}$$
for $x \in (a,b)$. So, we conclude that $X \sim \text{Unif}(a,b)$.

2. Probably the best method for generating from $\mathcal{PO}(\lambda)$ is by the method of inversion. The cdf at integer values is $F(k) = e^{-\lambda}\sum_{i=0}^{k}\frac{\lambda^i}{i!}$, so we can generate $U \sim \text{Unif}(0,1)$ and find the smallest k such that $F(k) > U$ is distributed from $\mathcal{PO}(\lambda)$.

   One alternative to it is based on the Poisson process: When the interarrival times of a process are i.i.d. and distributed from Exp(1), the number of arrivals in an interval of λ is $\mathcal{PO}(\lambda)$. From this, we can produce $E_i \overset{\text{i.i.d.}}{\sim} \text{Exp}(1)$ and $N := \max\{n : \sum_{i=1}^{n} E_i \leq \lambda\} \sim \mathcal{PO}(\lambda)$ (keep generating E_i's until the sum exceeds λ). Equivalently, $N = \max\{n : \prod_{i=1}^{n} U_i \geq e^{-\lambda}\}$ with $U_i \sim \text{Unif}(0,1)$ (why?). If λ is large this requires about λ uniform variables to generate one point.

3. The pdf of the Laplace distribution Laplace(a,b) is $p_X(x) = \frac{1}{2b}\exp\left(-\frac{|x-a|}{b}\right)$. Notice that $Y = X - a$ is centred with
$$p_Y(y) = \frac{1}{2b}\exp\left(-\frac{|y|}{b}\right).$$
This pdf is a two sided version of the pdf of the exponential distribution, where each side is multiplied by a half so that the integral is 1. Let $Z \sim \text{Exp}(b)$ and

$Y = Z$ with probability $1/2$ and $Y = -Z$ with probability $1/2$. One can find out using composition that Y has pdf $p_Y(y)$ given above. Therefore, we can generate $X \sim \text{Laplace}(a, b)$ as follows

- Generate $Z \sim \text{Exp}(b)$,
- Set $Y = Z$ or $Y = -Z$ with probability $1/2$.
- Set $X = Y + a$.

4. We have X' is $p_{X'}(x) = q(x)$, $\widehat{p}(x) = p(x)Z_p$ and $\widehat{q}(x) = q(x)Z_q$, and $\mathbb{P}(\text{Accept}|X' = x) = \frac{\widehat{p}(x)}{M\widehat{q}(x)} = \frac{1}{M}\frac{Z_p p(x)}{Z_q q(x)}$. So, the acceptance probability can be derived as

$$\begin{aligned} \mathbb{P}(\text{Accept}) &= \int \mathbb{P}(\text{Accept}|X' = x)p_{X'}(x)dx \\ &= \int \frac{1}{M}\frac{Z_p p(x)}{Z_q q(x)} q(x)dx \\ &= \frac{1}{M}\frac{Z_p}{Z_q}\int p(x)dx \\ &= \frac{1}{M}\frac{Z_p}{Z_q} \end{aligned}$$

The validity can be verified by considering the distribution of the accepted samples. Using Bayes' theorem,

$$p_X(x) = p_{X'}(x|\text{Accept}) = \frac{p_{X'}(x)\mathbb{P}(\text{Accept}|X' = x)}{\mathbb{P}(\text{Accept})} = \frac{q(x)\frac{1}{M}\frac{Z_p p(x)}{Z_q q(x)}}{\frac{Z_p}{Z_q}(1/M)} = p(x).$$

8. The pdf of $\text{Beta}(a, b)$ is

$$p(x) = \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)} \propto x^{a-1}(1-x)^{b-1} =: \widehat{p}(x), \quad x \in (0,1).$$

We have $Q = \text{Unif}(0, 1)$ so $q(x) = 1$ and the ratio $\widehat{p}(x)/q(x) = \widehat{p}(x)$.

- First, it can be seen that the ratio is unbounded for $a < 1$ or $b < 1$, so $Q = \text{Unif}(0, 1)$ cannot be used.
- When $a = b = 1$, we have the uniform distribution for X so it is trivial.
- For $a \geq 1$ and $b \geq 1$ and at least one of them is strictly greater than 1, the first derivative of $\widehat{p}(x)$ is equal to 0 at $x = \frac{a-1}{a+b-2}$, and the second derivative at that value of x is $-(a + b - 2)^2 \left(\frac{1}{a-1} + \frac{1}{b-1}\right)$, which is negative, so $x^* = \frac{a-1}{a+b-2}$ is a maximum point, yielding

$$\widehat{p}(x)/q(x) \leq \widehat{p}(x^*) = \left(\frac{a-1}{a+b-2}\right)^{a-1}\left(\frac{b-1}{a+b-2}\right)^{b-1}$$

so the smallest (hence the best) M we can choose is $M^* = \left(\frac{a-1}{a+b-2}\right)^{a-1}\left(\frac{b-1}{a+b-2}\right)^{b-1}$. Hence the rejection sampling algorithm can be applied as follows:

(a) Sample $X' \sim \text{Unif}(0,1)$ and $U \sim \text{Unif}(0,1)$

(b) If $U \leq \frac{(X')^{a-1}(1-X')^{b-1}}{\left(\frac{a-1}{a+b-2}\right)^{a-1}\left(\frac{b-1}{a+b-2}\right)^{b-1}}$, accept $X = X'$, else restart.

B.2 Exercises in Chapter 3

1. For $\mu = 0$ and $\varphi(x) = x^2$, . Find the optimum k for this ϕ and calculate the gain due to variance reduction compared to the plug-in estimator $P^N_{\text{MC}}(\varphi)$.

   When $\varphi(x) = x^2$ and $\mu = 0$, $Q_{2-k}(\varphi^2) = \mathbb{E}_{Q_{2-k}}(X^4) = \frac{3\sigma^4}{(2-k)^2}$. Therefore, we need to minimise

   $$\mathbb{V}_{Q_k}(P^N_{\text{IS}}(\varphi)) = \frac{1}{\sqrt{k(2-k)}}\frac{3\sigma^4}{(2-k)^2} = 3\sigma^4(2-k)^{-5/2}k^{-1/2}.$$

   The minimum is attained at $k = 1/3$ and is

   $$\mathbb{V}_{Q_{1/3}}(P^N_{\text{IS}}(\varphi)) = \frac{1}{N}\left\{3\sigma^4(2-k)^{-5/2}k^{-1/2} - \underbrace{\mathbb{E}(X^2)^2}_{\sigma^4}\right\}\Bigg|_{k=1/3} = 0.4490\frac{\sigma^4}{N}$$

   The variance of the plug-in estimator $P^N_{\text{MC}}(\varphi) = \frac{1}{N}\sum_{i=1}^N X_i^2$, $X_i \sim P$, is

   $$\mathbb{V}(P^N_{\text{MC}}(\varphi)) = \frac{\mathbb{E}(X^4) - \mathbb{E}(X^2)^2}{N} = \frac{3\sigma^4 - \sigma^4}{N} = \frac{2\sigma^4}{N}.$$

   Therefore the IS estimator provides a variance-reduction factor of ≈ 0.22.

2. - For an acyclic directed graph, see for example `http://statweb.stanford.edu/~owen/mc/Ch-var-is.pdf`.
   - Importance sampling part: The probability $\mathbb{P}(E_{10} > 70) = \mathbb{E}(\varphi(X))$ where $X = (T_1, \ldots, T_{10})$ and $\varphi(X) = \mathbb{I}(E_{10} > 70)$ can be estimated via importance sampling by sampling independent $X^{(i)}$'s where

     $$X^{(i)} = (T_1^{(i)}, \ldots, T_{10}^{(i)}), \quad T_j^{(i)} \sim \text{Exp}(1/\lambda_j)$$

     The proposal density for this choice at $x = (t_1, \ldots, t_{10})$ is $q(x) = \prod_{j=1}^{10}\frac{1}{\lambda_j}e^{-t_j/\lambda_j}$, so the weights will be.

     $$w(X^{(i)}) = \frac{p(X^{(i)})}{q(X^{(i)})} = \prod_{j=1}^{10}\frac{\lambda_j}{\theta_j}e^{\left(\frac{1}{\lambda_j}-\frac{1}{\theta_j}\right)T_j^{(i)}}$$

Therefore, the overall importance sampling estimate is

$$P_{\text{IS}}^{N}(\varphi) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(E_{10}^{(i)} > 70) \prod_{j=1}^{10} \frac{\lambda_j}{\theta_j} e^{\left(\frac{1}{\lambda_j}-\frac{1}{\theta_j}\right)T_j^{(i)}}$$

where $E_{10}^{(i)}$ is the computed completion time for the i'th sample $X^{(i)}$.

B.3 Exercises of Chapter 4

1. The full tables are given below.

$p_{X,Y}(x,y)$	$y=1$	$y=2$	$y=3$	$y=4$	$p_X(x)$
$x=1$	$1/40$	$3/40$	$4/40$	$2/40$	$10/40$
$x=2$	$5/40$	$7/40$	$6/40$	$5/40$	$23/40$
$x=3$	$1/40$	$2/40$	$2/40$	$2/40$	$7/40$
$p_Y(y)$	$7/40$	$12/40$	$12/40$	$9/40$	

$p_{X\|Y}(x\|y)$	$y=1$	$y=2$	$y=3$	$y=4$
$x=1$	$1/7$	$3/12$	$4/12$	$2/9$
$x=2$	$5/7$	$7/12$	$6/12$	$5/9$
$x=3$	$1/7$	$2/12$	$2/12$	$2/9$

$p_{Y\|X}(y\|x)$	$y=1$	$y=2$	$y=3$	$y=4$
$x=1$	$1/10$	$3/10$	$4/10$	$2/10$
$x=2$	$5/23$	$7/23$	$6/23$	$5/23$
$x=3$	$1/7$	$2/7$	$2/7$	$2/7$

2. We have $p_X(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}e^{-\beta x}$, and $p_{Y|X}(y|x) = xe^{-xy}$, so

$$\begin{aligned} p_{X|Y}(x|y) &\propto p_X(x)p_{Y|X}(y|x) \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}e^{-\beta x} x e^{-xy} \\ &\propto x^{\alpha} e^{-(\beta+y)x}, \end{aligned}$$

which is in the same form of the pdf of $\Gamma(\alpha+1, \beta+y)$. Therefore, $\alpha_{x|y} = \alpha + 1$ and $\beta_{x|y} = \beta + y$.

3. Let us define

$$\mu(y) = \mathbb{E}(X|Y=y) = \int xp(x|y)dx$$

for brevity (we can define such μ since $\mathbb{E}(X|Y=y)$ is a function of y only). We can write any estimator as $\hat{X}(Y) = \mu(Y) + \hat{X}(Y) - \mu(Y)$. Now, the expected MSE is

$$\begin{aligned}\mathbb{E}((\hat{X}(Y) - X)^2) &= \int (\hat{X}(y) - x)^2 p(x,y) dx dy \\ &= \int \left[\int (\hat{X}(y) - x)^2 p(x|y) dx \right] p(y) dy \\ &= \int \left[\int [\mu(y) - x + \hat{X}(y) - \mu(y)]^2 p(x|y) dx \right] p(y) dy\end{aligned}$$

Let us focus on the inner integral first.

$$\begin{aligned}&\int [\mu(y) - x + \hat{X}(y) - \mu(y)]^2 p(x|y) dx \\ &= \int \left[(\mu(y) - x)^2 + (\hat{X}(y) - \mu(y))^2 + 2(\mu(y) - x)(\hat{X}(y) - \mu(y)) \right] p(x|y) dx \\ &= \int (\mu(y) - x)^2 p(x|y) dx + (\hat{X}(y) - \mu(y))^2 + 2(\hat{X}(y) - \mu(y)) \int (\mu(y) - x) p(x|y) dx \\ &= \int (\mu(y) - x)^2 p(x|y) dx + (\hat{X}(y) - \mu(y))^2 \qquad \text{(B.1)}\end{aligned}$$

where the last equation follows since the last term is zero:

$$\int (\mu(y) - x) p(x|y) dx = \mu(y) - \int x p(x|y) dx = 0.$$

The first term in (B.1) does not depend on the estimator $\hat{X}(y)$ so we have control only on the second term $(\hat{X}(y) - \mu(y))^2$ which is always nonnegative and therefore minimum when $\hat{X}(y) - \mu(y) = 0$, i.e. $\hat{X}(y) = \mu(y) = \mathbb{E}(X|Y=y)$. Since this is true for all y, we conclude that the estimator $\hat{X}(Y) = \mathbb{E}(X|Y)$, as a random variable of Y, minimises the expected MSE.

4. - We can use the result in Example 4.7 which states

$$Y|X = x \sim \mathcal{N}(Ax, R), \quad X \sim \mathcal{N}(m, S) \Rightarrow X|Y = y \sim \mathcal{N}(m_{x|y}, S_{x|y})$$

where $S_{x|y} = (S^{-1} + A^T R^{-1} A)^{-1}$ and $m_{x|y} = S_{x|y}(S^{-1}m + A^T R^{-1} y)$. By observation, we can see that X is a univariate number with $m = 0$ and $S = \sigma_x^2$, A is an $n \times 1$ vector with $A(t) = \sin(2\pi t/T)$, $t = 1, \ldots, n$, and $R = \sigma_y^2 I_n$. Therefore $p(x|y_{1:n}) = \phi(x; m_{x|y}, S_{x|y})$ with

$$S_{x|y} = \left(\frac{1}{\sigma_x^2} + \frac{1}{\sigma_y^2} \sum_{t=1}^{n} \sin^2(2\pi t/T) \right)^{-1}, \qquad \text{(B.2)}$$

and

$$m_{x|y} = S_{x|y} \frac{1}{\sigma_y^2} \sum_{t=1}^{n} \sin(2\pi t/T) y_t.$$

It is possible to derive $p(y_{1:n})$ from $p(y_{1:n}) = p(x, y_{1:n})/p(x|y_{1:n})$, but there is an easier way when the prior and the likelihood is gaussian and the relation between X and Y is linear. One can view

$$Y = AX + V$$

where X and A are as before and $V \sim \mathcal{N}(0_n, \sigma_y^2 I_n)$. As covered already in Section 2.2.2, since X and V are gaussian, $Y_{1:n} \sim \mathcal{N}(m_n, \Sigma_n)$ with

$$m_n = \mathbb{E}(AX + V) = A0_n + 0_n = 0_n, \quad \Sigma_n = \text{Cov}(AX + V) = \sigma_x^2 AA^T + \sigma_y^2 I_n.$$

- $f(n+1, X) = \sin(2\pi(n+1)/T)X$ and we can view $\sin(2\pi(n+1)/T)$ as a scalar constant. Since $X|Y_{1:n} = y_{1:n} \sim \mathcal{N}(m_{x|y}, S_{x|y})$, we have

$$f(n+1, X)|Y_{1:n} = y_{1:n} \sim \mathcal{N}(\sin(2\pi(n+1)/T)m_{x|y}, \sin^2(2\pi(n+1)/T)S_{x|y})$$

- First, let us write

$$Y_{n+1}|X \sim \mathcal{N}(f(n+1, X), \sigma_y^2) = \mathcal{N}(\sin(2\pi(n+1)/T)X, \sigma_y^2).$$

The unconditional (marginal) distribution of Y_{n+1} can be calculated in a similar way as done for $Y_{1:n}$. Since the unconditional distribution of X is $X \sim \mathcal{N}(0, \sigma_x^2)$, we have

$$Y_{n+1} \sim \mathcal{N}(0, \sin^2(2\pi(n+1)/T)\sigma_x^2 + \sigma_y^2). \tag{B.3}$$

Conditional on $Y_{1:n} = y_{1:n}$, we have

$$X|Y_{1:n} = y_{1:n} \sim \mathcal{N}(m_{x|y}, S_{x|y})$$

and

$$Y_{n+1}|X = x, Y_{1:n} = y_{1:n} \sim \mathcal{N}(x \sin(2\pi(n+1)/T), \sigma_y^2)$$

as before, since Y_{n+1} is conditionally independent from $Y_{1:n}$ given X. We can use the same mechanism as before and derive

$$Y_{n+1}|Y_{1:n} = y_{1:n} \sim \mathcal{N}(\sin(2\pi(n+1)/T)m_{x|y}, \sin^2(2\pi(n+1)/T)S_{x|y} + \sigma_y^2). \tag{B.4}$$

When compare the variances (B.3) and (B.4), we see that the second one is smaller since $S_{x|y} < \sigma_x^2$, see (B.2). The decrease in the variance, hence uncertainty, is due to information that comes from $Y_{1:n} = y_{1:n}$.

B.4 Exercises of Chapter 7

2. For $t > 1$, the optimum proposal is

$$q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1}, y_{t-1}) = \frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|x_{t-1})}$$

Conditional on x_{t-1}, X_t has prior $\mathcal{N}(ax_{t-1}, \sigma_x^2)$ and $Y|X_t = x_t \sim \mathcal{N}(bx_t, \sigma_y^2)$. Therefore, using conjugacy, we have

$$p(x_t|x_{t-1}, y_{t-1}) = \phi(x_t; \mu_q, \sigma_q^2), \quad \sigma_q^2 = \left(1/\sigma_x^2 + b^2/\sigma_y^2\right)^{-1}, \quad \mu_q = \sigma_q^2(ax_{t-1}/\sigma_x^2 + by_t/\sigma_y^2).$$

The resulting incremental weight is

$$w_{t|t-1}(x_{t-1}, x_t) = \frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(x_t|x_{t-1}, y_t)} = p(y_t|x_{t-1}),$$

which only depends on x_{n-1}. It can be checked that $Y_t = bX_t + V_t$, $V_t \sim \mathcal{N}(0, \sigma_y^2)$ given $X_{t-1} = x_{t-1}$ is Gaussian with mean bax_{t-1} and variance $b^2\sigma_x^2 + \sigma_y^2$, i.e.

$$w_{t|t-1}(x_{t-1}, x_t) = p(y_t|x_{t-1}) = \phi(y_t; bax_{t-1}, b^2\sigma_x^2 + \sigma_y^2).$$

For $t = 1$, we get similar results by replacing $f(x_t|x_{t-1})$ with $\eta(x_1)$ and considering $Y_1 = bX_1 + V_1$ with $V_1 \sim \mathcal{N}(0, \sigma_y^2)$. This results in

$$q(x_1|y_1) = p(x_1|y_1) = \phi(x_1; \mu_q, \sigma_q^2), \quad \sigma_q^2 = \left(1/\sigma_0^2 + b^2/\sigma_y^2\right)^{-1}, \quad \mu_q = \sigma_q^2 by_t/\sigma_y^2.$$

and

$$w_1(x_1) = p(y_1) = \phi(y_1; 0, b^2\sigma_0^2 + \sigma_y^2).$$

3. Since each $Y_{t,i}$ is lognormal distributed with parameters $\log R_i$ and σ_y^2, we have

$$g(y_t|x_t) = \prod_{i=1}^{3} \frac{1}{\sqrt{2\pi\sigma_y^2}y_{t,i}} e^{-\frac{1}{2\sigma_y^2}(\log y_{t,i} - \log R_{t,i})^2}$$

This result can be reached by transformation of random variables, considering that the $\log Y_{t,i}$ is Gaussian with mean $\log R_{t,i}$ and variance σ_y^2.

References

Anderson, B. and Moore, J. (1979). *Optimal Filtering.* Prentice-Hall, New York. 7.2

Andrieu, C., Davy, M., and Doucet, A. (2001). Improved auxiliary particle filtering: applications to time-varying spectral analysis. In *Statistical Signal Processing, 2001. Proceedings of the 11th IEEE Signal Processing Workshop on*, pages 309–312. 7.1

Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64:827–836. 7.3.6

Andrieu, C., Doucet, A., and Tadić, V. B. (2005). On-line parameter estimation in general state-space models. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 332–337. 12

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188. 7.3.4.1

Blackwell, D. (1947). Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, 18(1):105–110. 7.3.6

Cappé, O., Godsill, S., and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924. 7.3.4.1

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models.* Springer. 5.2, 6.1

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). An improved particle filter for nonlinear problems. *Radar Sonar & Navigation, IEE Proceedings*, 146:2–7. 12

Chen, R. and Liu, J. (1996). Predictive updating methods with application to Bayesian classification. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58:397–415. 8

Chen, R. and Liu, J. S. (2000). Mixture kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(3):493–508. 7.3.5, 7.3.6

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411. 7.3.6

Crisan, D., Moral, P. D., and Lyons, T. (1999). Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields*, 5(3):293–318. 12

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer-Verlag, New York. 6.1, 7.3.4.1, 7.3.5

Del Moral, P. and Doucet, A. (2003). On a class of genealogical and interacting metropolis models. In Azéma, J., Émery, M., Ledoux, M., and Yor, M., editors, *Séminaire de Probabilités XXXVII*, volume 1832 of *Lecture Notes in Mathematics*, pages 415–446. Springer Berlin Heidelberg. 12

Doucet, A. (1997). *Monte Carlo methods for Bayesian estimation of hidden Markov models. Application to radiation signals (in French)*. PhD thesis, University Paris-Sud Orsay, France. 8, 8

Doucet, A., Briers, M., and Sénécal, S. (2006). Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711. 12

Doucet, A., De Freitas, J., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York. 6.1, 7.1, 7.3.4.1

Doucet, A., de Freitas, N., Murphy, K., and Russell, S. (2000a). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 176–183, San Francisco, CA. Morgan Kaufmann. 7.3.5, 7.3.6, 7.3.6

Doucet, A., Godsill, S., and Andrieu, C. (2000b). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208. 6.1, 6.2, 6.3, 7.3.4.1

Doucet, A. and Johansen, A. M. (2009). A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. In Crisan, D. and Rozovsky, B., editors, *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press. 7.3.4.1

Durbin, J. and Koopman, S. J. (2000). Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62:3–56. 7.3.4.1

Eckhardt, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo method. *Los Alamos Science, Special Issue*, pages 131–137. 1.2, 2.2.1, 2.2.4

Fearnhead, P. (2008). Computational methods for complex stochastic systems: a review of some alternatives to MCMC. *Statistics and Computing*, 18(2):151–171. 7.3.4.1

Fearnhead, P. and Clifford, P. (2003). On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65:887–889. 12, 7.3.5

Fearnhead, P. and Liu, Z. (2007). On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605. 12

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409. 5.4

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6(6):721–741. 5.4

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339. 5, 6.2

Gilks, W. R. (1992). Derivative-free adaptive rejection sampling for Gibbs sampling. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4*, pages 641–649. Oxford University Press, Oxford, UK. 2.2.4.2

Gilks, W. R. and Berzuini, C. (2001). Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146. 12

Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995). Adaptive rejection Metropolis sampling within Gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 44(4):455–472. 2.2.4.2

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC. 5.1, 5.2

Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(2):337–348. 2.2.4.2

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F*, 140(6):107–113. 12, 7.1, 7.3.4.5

Handschin, J. E. (1970). Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6:555–563. 6.2

Handschin, J. E. and Mayne, D. (1969). Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9:547–559. 6.2

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 52(1):97–109. 5.1, 5.3, 4

Hitchcock, D. B. (2003). A history of the Metropolis-Hastings algorithm. *The American Statistician*, 57:254–257. 4

Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls 3*, pages 182–193. 7.3.4.1

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME; Series D: Journal of Basic Engineering*, 82:35–45. 7.3.1.1

Kitagawa, G. (1996). Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 1:1–25. 12, 7.3.4.1

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288. 3.1.1, 8, 6.3, 7.3.4.5

Lin, M. T., Zhang, J. L., Cheng, Q., and Chen, R. (2005). Independent particle filters. *Journal of the American Statistical Association*, 100(472):1412–1421. 7.3.4.5

Liu, J. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119. 3.1.1, 7.3.4.5

Liu, J. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer Verlag, New York, NY, USA. 12

Liu, J. and Chen, R. (1995). Blind deconvolution via sequential imputation. *Journal of the American Statistical Association*, 90:567–576. 8, 7.3.4.5

Liu, J. and Chen, R. (1998). Sequential Monte-Carlo methods for dynamic systems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 93:1032–1044. 12, 7.3.4.1

Marsaglia, G. (1977). The squeeze method for generating gamma variates. *Computers and Mathematics with Applications*, 3(4):321–325. 2.2.4.2

Mayne, D. (1966). A solution of the smoothing problem for linear dynamic systems. *Automatica*, 4:73–92. 6.2

Mengersen, K. and Tweedie, R. L. (1996). Rates of convergence of the Hastings and Metropolis algorithms. *Annals of Statistics*, 24:101–121. 4

Metropolis, N. (1987). The beginning of the Monte Carlo method. *Los Alamos Science, Special Issue*, pages 125–130. 1.2

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092. 5.1, 5.3, 4

Metropolis, N. and Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):pp. 335–341. 1.2

Meyn, S. and Tweedie, R. L. (2009). *Markov Chains and Stochastic Stability*. Cambridge University Press, New York, NY, USA, 2nd edition. 5.2

Newman, M. E. J. and Barkema, G. T. (1999). *Monte Carlo Methods in Statistical Physics*. Oxford University Press, USA. 1.2.1

Olsson, J., Cappé, O., Douc, R., and Moulines, E. (2008). Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14:155–179. 12

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599. 7.1

Press, W. H. (2007). *Numerical Recipes : The Art of Scientific Computing*. Cambridge University Press, 3rd edition. 1.2.1

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286. 7.1, 7.2, 7.3.1.1

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. New York: Springer, 2 edition. 2.2.1, 3.1, 3.1.1, 5.1, 5.2, 4, 6.1

Roberts, G. and Smith, A. (1994). Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and their Applications*, 49(2):207–216. 4, 4

Roberts, G. and Tweedie, R. (1996). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83:95–110. 4, 4

Rubin, D. B. (1987). A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when the fraction of missing information is modest: the SIR algorithm (discussion of Tanner and Wong). *Journal of the American Statistical Association*, 82:543–546. 12

Särkkä, S., Vehtari, A., and Lampinen, J. (2004). Rao-Blackwellized Monte Carlo data association for multiple target tracking. In *In Proceedings of the Seventh International Conference on Information Fusion*, pages 583–590. 7.3.6

Shiryaev, A. N. (1995). *Probability*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 edition. 1.2.1, 5.2

Sorenson, H. W. (1985). *Kalman Filtering: Theory and Application*. IEEE Press, reprint edition. 7.3.4.1

Tierney, L. (1994). Markov chains for exploring posterior distributions. *Annals of Statistics*, 22:1701–1762. 5.2, 4, 4, 5.4.1

von Neumann, J. (1951). Various techniques used in connection with random digits. *Journal of Research of the National Bureau of Standards*, 12:36–38. 2.2.4

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85. 12