CHAPTER

MONTE CARLO METHODS

14

CHAPTER OUTLINE

14.1 INTRODUCTION

In Chapters 12 and 13, the Bayesian inference task was considered. A large part of the latter chapter was dedicated to dealing with approximation techniques, which offered escape routes when the involved pdfs were complex enough to render integral computations intractable. All these techniques were of a deterministic nature; that is, the goal was to approximate the mathematical expression of the corresponding pdf by another one that could ease the associated calculations. Such methods include the Laplacian approximation, as well as the variational methods based on the mean field theory or the convex duality concept. Deterministic approximation methods will also be used for approximate inference in Chapter 15, to deal with graphical models.

In this chapter, we turn our attention to approximation methods with a much stronger statistical flavor, which are based on randomly generating samples using numerical techniques; these samples

Machine Learning. http://dx.doi.org/10.1016/B978-0-12-801522-3.00014-8

are typical of an underlying distribution, which may be of either continuous or discrete nature. This is an old field, with origins tracing back to the late forties and early fifties in the pioneering work of Stanislav Ulam, John Von-Neumann, and Nicholas Metropolis in Los Alamos, when the term *Monte Carlo* was coined as an umbrella name of such techniques, inspired by the famous casino in Monaco (see, e.g., [28] for a historical note). The first application of such techniques, which coincided with the development of the first computers, was in the context of the Manhattan project for developing the hydrogen bomb; soon after, Monte Carlo methods were embraced by almost every scientific area where statistical computations are involved.

As is often the case with pioneering ideas, when they are looked at a posteriori, that is, once they have been stated, the basic idea seems simple. Our current task of interest is the computation of an integral, which involves a pdf; this can alternatively be interpreted as the computation of an "expectation." Such a view provides the permit to approximate the integral as the sample mean of the involved quantities, given a sufficient number of samples and exploiting the law of large numbers.

To condense a field with a history of a number of decades in a single chapter is obviously impossible. Our goal is to present the basic concepts, definitions, and directions, with the aim of serving the needs associated with typical machine learning tasks rather than looking at it as an entity on its own.

We start with the more classical methods using transformations and then move on to the rejection and importance sampling techniques. In the sequel, the more powerful methods based on arguments from the theory of Markov chains are reviewed. The Metropolis-Hastings and the Gibbs sampling methods are presented and discussed. Finally, a case study concerning the change-point detection task is considered.

14.2 MONTE CARLO METHODS: THE MAIN CONCEPT

Our starting point is the evaluation of integrals of the form

$$\mathbb{E}\left[f(\mathbf{x})\right] := \int_{-\infty}^{\infty} f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{14.1}$$

where $\mathbf{x} \in \mathbb{R}^l$ is a random vector and $p(\boldsymbol{x})$ is the corresponding distribution.[1] Our interest lies in cases where the forms of $f(\boldsymbol{x})$ and/or $p(\boldsymbol{x})$ are such that the evaluation of such integrals is intractable. For example, such integrations occur in the evaluation of the evidence in Eq. (12.14), in the prediction task (Eq. (12.18)), and in the E-step of the EM algorithm (Eq. (12.61)). In Eq. (12.14), the random variable is the parameter vector θ and $f(\boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{\theta})$.

Coming back to Eq. (14.1), assume that one has at her/his disposal a number of i.i.d. samples, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, drawn from $p(\boldsymbol{x})$. Then, the following approximation

$$\mathbb{E}\left[f(\mathbf{x})\right] \simeq \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}_i) := \bar{\mathbb{E}}_{f,N}, \tag{14.2}$$

[1] In the case of discrete variables, $p(\boldsymbol{x})$ becomes the probability mass function, $P(\boldsymbol{x})$, and integrations are replaced by summations.

is justified by (a) the *law of large numbers* and (b) the *central limit theorem* [32]. Let us denote as $\mathbb{E}[f(\mathbf{x})] = \mu_f$ and the respective variance as $\text{var}[f(\mathbf{x})] := \mathbb{E}\left[(f(\mathbf{x}) - \mathbb{E}[f(\mathbf{x})])^2\right] = \sigma_f^2$. Then, the previously referred two theorems guarantee that

$$\lim_{N\to\infty} \bar{\mathbb{E}}_{f,N} = \mu_f, \tag{14.3}$$

and

$$p(\bar{\mathbb{E}}_{f,N}) \simeq \mathcal{N}\left(\bar{\mathbb{E}}_{f,N} | \mu_f, \frac{\sigma_f^2}{N}\right). \tag{14.4}$$

The limit in Eq. (14.3) refers to the notion of *almost sure convergence*, that is,

$$\text{Prob}\left\{\lim_{N\to\infty} |\mu_f - \bar{\mathbb{E}}_{f,N}| = 0\right\} = 1.$$

The approximate Gaussian distribution in Eq. (14.4) guarantees that the variance (as one changes the set of N samples) of the obtained estimate, $\bar{\mathbb{E}}_{f,N}$, around the true value, μ_f, decreases with N.

Thus, if one generates the samples $\boldsymbol{x}_n,\ n = 1, 2, \ldots, N$, from the distribution $p(\boldsymbol{x})$, the use of Monte Carlo techniques offers the means for an approximation of the integral in Eq. (14.1) with the following nice properties: (a) the approximation error is decreasing as $\frac{1}{\sqrt{N}}$; (b) the obtained estimate using N samples is an unbiased estimate of the true value; and (c) the convergence rate is *independent* on the dimensionality, l. The latter property is in contrast to methods based on the deterministic numerical integration, which, in general, have a rate of convergence that slows down as the dimensionality increases. In Monte Carlo techniques, if one is not satisfied with the obtained accuracy, all he/she has to do is generate more samples.

The crucial point now becomes that of developing techniques to generate i.i.d. samples from $p(\boldsymbol{x})$. This is not an easy task, especially for high-dimensional spaces. Note that achieving a certain accuracy for the estimator in Eq. (14.2) is independent on the dimensionality, once i.i.d. samples drawn from $p(\boldsymbol{x})$ are available. On the other hand, drawing i.i.d. samples typical of $p(\boldsymbol{x})$ becomes harder as the dimensionality increases. We will return to this point soon. In the sequel, we will focus on some basic directions on how to achieve the aforementioned goal.

14.2.1 RANDOM NUMBER GENERATION

Random number generation can be achieved either as the result of an experiment or via the use of computers. For example, the tosses of a fair coin can generate a random sequence of 0's (heads) or 1's (tails). Another example is the sequence of numbers corresponding to the distance between radioactive emissions; such an experiment generates a sequence of exponentially distributed samples. However, such approaches are not of much practical value and the emphasis has been on techniques that generate samples via a computer, using a *pseudorandom number generator*. At the heart of such methods lie algorithms that guarantee the generation of a sequence of *integers*, z_i, which approximately follow a *uniform distribution* in an interval in the real axis. In the sequel, the generation of random numbers/vectors, which follow an arbitrary distribution, is obtained *indirectly* via a variety of methods, each with its pros and cons. The path for generating integers in an interval, $(0, M)$, follows the general recursion,

$$z_i = g(z_{i-1}, \ldots, z_{i-m}) \bmod M,$$

where g is a function depending on the m previously generated samples and mod denotes the modulus operation; that is, z_i is the remainder of the division of $g(z_{i-1}, \ldots, z_{i-m})$ by M. The simpler form is the linear version,

$$z_i = \alpha z_{i-1} \bmod M, \quad z_0 = 1, \; i \geq 1, \tag{14.5}$$

where M is a large prime number and α is an integer. Recursion (14.5) generates a sequence of numbers between 1 and $M - 1$. The method is known as *linear congruential generator* or Lehmer's algorithm [20].

If α is properly chosen, then the resulting sequence of numbers turns out to be periodic with period $M - 1$. This is the reason we call these generators pseudorandom, because a periodic sequence can never be claimed to be random. However, for large values of M, the obtained sequence can be sufficiently random with uniform distribution, provided, of course, that $N \leq M - 1$. For example, a value of M of the order of 10^9 is sufficient for most applications. Note that not all possible choices of the parameter α guarantee a good generator. In practice, a sequence is accepted as being random only if it meets a number of related tests of randomness and is subsequently used successfully in a variety of applications (see, e.g., [32]). A common choice of parameters that leads to a reasonably good uniformly distributed random sequence is $\alpha = 7^5$ and $M = 2^{31} - 1$ (see, e.g., [34]). More on this topic can be found in Knuth's classical text and the references therein [18]. Once a sequence of integers is available, a sequence of uniformly distributed real random numbers is obtained as the ratio $x_i = \frac{z_i}{M} \in (0, 1)$ (as a matter of fact, this is the sequence on which the randomness tests are applied).

Remarks 14.1.

- Note that even the generation of a sequence of (pseudo) random numbers with uniform distribution in $(0, 1)$, is not an easy task, in spite of the fact that the uniform distribution is an "easy" one; that is, all values are equally probable. Moreover, often in practice, a pdf is known up to its normalizing constant, that is,

  $$p(\boldsymbol{x}) = \frac{\phi(\boldsymbol{x})}{Z},$$

  where

  $$Z = \int_{-\infty}^{+\infty} \phi(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}.$$

  However, if $\phi(\boldsymbol{x})$ has a complicated form, the previous integration may be intractable. This is often met when computing posterior pdfs. The previous points make the process of sampling from a general $p(\boldsymbol{x})$ much harder than for the case of a uniform one. The task becomes even harder in high dimensions, even if Z is available. The required number of points, in order to cover sufficiently a region in a high-dimensional space, exhibits an exponential dependence on the respective dimensionality (curse of dimensionality). Thus, a huge number of points is needed in order to get a good representation of $p(\boldsymbol{x})$ in high-dimensional spaces. In practice, one would be more content to generate samples from the regions where $p(\boldsymbol{x})$ gets relatively high values. However, the higher the dimensionality, the more difficult the task of locating the high-probability regions. Similar arguments hold for random variables of a discrete nature, where the number of states that the variable can take is very large. Ideally, in order to have a representative sequence of samples, all states have to be visited.

14.3 RANDOM SAMPLING BASED ON FUNCTION TRANSFORMATION

In this section, we deal with some of the most basic techniques for drawing samples from a pdf, $p(\boldsymbol{x})$.

Function Inversion. Let x be a real random variable with a pdf, $p(x)$, and a corresponding cumulative distribution function

$$F_{\mathrm{x}}(x) = \int_{-\infty}^{x} p(\tau)\, \mathrm{d}\tau.$$

It is known from probability theory that the random variable, u, defined as

$$\mathrm{u} := F_{\mathrm{x}}(\mathrm{x}), \tag{14.6}$$

is *uniformly* distributed in the interval $0 \leq u \leq 1$ *irrespective of the nature* of $p(x)$ [32] (Problem 14.1). If, in addition, we assume that the function F_{x} has an inverse, F_{x}^{-1}, then we can write that

$$\mathrm{x} = F_{\mathrm{x}}^{-1}(\mathrm{u}). \tag{14.7}$$

Thus, following the reverse arguments, samples from $p(x)$ can be generated by first generating samples from the uniform distribution, $\mathcal{U}(u|0,1)$, and then applying on them the inverse function, F_{x}^{-1} (Problem 14.2).

This method works well provided that F_{x} has an inverse that can be easily computed. However, only a few pdfs can be "proud" of having inverses that can be expressed in an analytical form.

Example 14.1. Generate samples, x_n, that follow the exponential distribution,

$$p(x) = \lambda \exp(-\lambda x), \quad x \geq 0,\ \lambda > 0, \tag{14.8}$$

using a pseudorandom generator that generates samples, u_n, from the uniform distribution $\mathcal{U}(u|0,1)$.

We have that

$$F_{\mathrm{x}}(x) = \int_{0}^{x} \lambda \exp(-\lambda\tau)\, \mathrm{d}\tau = 1 - \exp(-\lambda x).$$

By letting

$$u := F_{\mathrm{x}}(x),$$

and solving for x, we get

$$x = -\frac{1}{\lambda}\ln(1-u) := F_{\mathrm{x}}^{-1}(u).$$

Hence, if u_n are samples drawn from a uniform distribution, the sequence

$$x_n = -\frac{1}{\lambda}\ln(1-u_n), \quad n = 1, 2, \ldots, N,$$

are samples drawn from the exponential pdf in Eq. (14.8). Figure 14.1 shows the histogram of the generated samples for $N = 1000$, alongside $p(x)$ for $\lambda = 1$.

Example 14.2. *Generating samples from discrete distributions*. Here, an intuitive method for generating samples from discrete distributions is presented. We will use such distributions in Section 17.2.

Let, $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_K$, denote discrete random events occurring with probabilities, $P_1, P_2, \ldots, P_K$, respectively, such that $\sum_{k=1}^{K} P_k = 1$. Then, the following simple algorithm draws samples from this distribution.

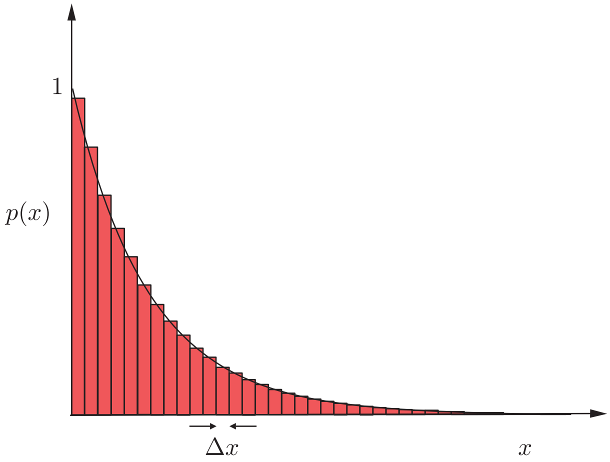


FIGURE 14.1

The histogram of the samples generated from the uniform distribution, and using the inverse of F_{x}, which describes the exponential pdf, whose curve is shown in black. The length of the bin interval, Δx, was chosen equal to 0.02.

Algorithm 14.1 (Sampling discrete distributions).

```
• Define $a_k = \sum_{i=1}^{k-1} P_i$, $b_k = \sum_{i=1}^{k} P_i$, $k = 1, 2, \ldots, K$, $a_1 = 0$.
• For $i = 1, 2, \ldots,$ Do
    • $u \sim \mathcal{U}(0, 1)$
    • Select
          $x_k \text{ if } u \in [a_k, b_k), \quad k = 1, 2, \ldots, K$
• End For
```

Figure 14.2 provides an illustration of the algorithm. Note that the probability jumps at the beginning of each interval and the corresponding cumulative distribution function (CDF) is constructed; the algorithm basically computes the inverse of u, according to this CDF (see, e.g., [4]).

Function Transformation. We will demonstrate the method via an example involving the transformation of two random variables, say r and ϕ, to two new ones, x and y. Let

$$\mathrm{x} = g_{\mathrm{x}}(\mathrm{r}, \phi),$$

and

$$\mathrm{y} = g_{\mathrm{y}}(\mathrm{r}, \phi).$$

Let us now assume that there is a unique solution for the inverses and that they can be expressed in an analytic form (which is not the case in general), that is,

$$\mathrm{r} = g_{\mathrm{r}}(\mathrm{x}, \mathrm{y}),$$
$$\phi = g_{\phi}(\mathrm{x}, \mathrm{y}).$$

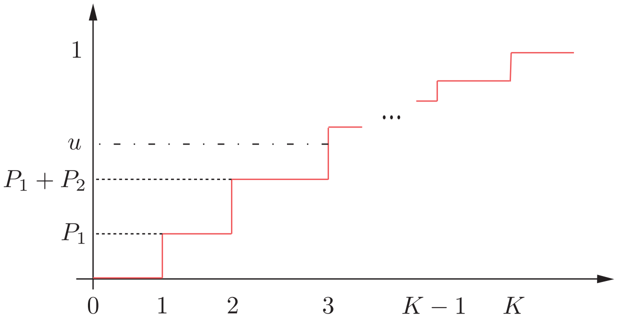


FIGURE 14.2

The CDF for a discrete distribution of K discrete random events. If $P_1 + P_2 \le u < P_1 + P_2 + P_3$, the event x_3 is drawn. Note that the larger the probability of an event, the larger the corresponding interval jump in the CDF is, hence the higher the probability of this event being drawn.

We know from Section 2.2.5 that if $p_{\mathrm{r},\phi}(r,\phi)$ is the joint distribution of r and ϕ, then the joint distribution of x and y is given by

$$p_{\mathrm{x,y}}(x,y) = \frac{p_{\mathrm{r},\phi}\big(g_{\mathrm{r}}(x,y), g_{\phi}(x,y)\big)}{\big|\det\big(J(\mathrm{x},\mathrm{y};\mathrm{r},\phi)\big)\big|}$$
$$= p_{\mathrm{r},\phi}\big(g_{\mathrm{r}}(x,y), g_{\phi}(x,y)\big)\big|\det\big(J(\mathrm{r},\phi;\mathrm{x},\mathrm{y})\big)\big|, \tag{14.9}$$

where $\big|\det\big(J(\mathrm{x},\mathrm{y};\mathrm{r},\phi)\big)\big|$ is the absolute value of the determinant of the Jacobian matrix,

$$J(\mathrm{x},\mathrm{y};\mathrm{r},\phi) = \begin{bmatrix} \frac{\partial g_{\mathrm{x}}}{\partial r} & \frac{\partial g_{\mathrm{x}}}{\partial \phi} \\ \frac{\partial g_{\mathrm{y}}}{\partial r} & \frac{\partial g_{\mathrm{y}}}{\partial \phi} \end{bmatrix}, \tag{14.10}$$

$J(\mathrm{r},\phi;\mathrm{x},\mathrm{y})$ is analogously defined), and we have assumed, for simplicity, that to each value of (r,ϕ) there corresponds one value of (x,y). Let us now see how one can generate samples from a Gaussian $p(x) = \mathcal{N}(x|0,1)$ by using samples drawn from a uniform and an exponential distribution, respectively, for ϕ and r; recall that in Example 14.1, we described a technique for generating samples from an exponential distribution.

The Box-Müller method. Let r be distributed according to an exponential distribution,

$$p_{\mathrm{r}}(r) = \frac{1}{2}\exp\left(-\frac{r}{2}\right), \quad r \ge 0, \tag{14.11}$$

and ϕ to a uniform distribution, $\mathcal{U}(\phi|0,1)$,

$$p_{\phi}(\phi) = \begin{cases} \frac{1}{2\pi} & 0 \le \phi \le 2\pi, \\ 0 & \text{otherwise}, \end{cases} \tag{14.12}$$

and also assume that they are independent, that is,

$$p_{\mathrm{r},\phi}(r,\phi) = p_{\mathrm{r}}(r)p_{\phi}(\phi). \tag{14.13}$$

Generate two new random variables as

$$\mathrm{x} = \sqrt{\mathrm{r}}\cos\phi, \tag{14.14}$$

$$y = \sqrt{r} \sin \phi. \tag{14.15}$$

The physical interpretation of the previous transformation is that x, y correspond to the cartesian coordinates of a point and r, ϕ are its polar ones, Figure 14.3. From Eqs. (14.14) and (14.15), we can write that

$$r = x^2 + y^2, \tag{14.16}$$

$$\phi = \arctan\left(\frac{y}{x}\right). \tag{14.17}$$

Adjusting Eq. (14.9) to our current needs, using Eqs. (14.11) to (14.13), we obtain

$$\begin{aligned} p_{x,y}(x, y) &= \frac{1}{2\pi}\frac{1}{2}\exp\left(-\frac{x^2+y^2}{2}\right)2 \\ &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y^2}{2}\right), \end{aligned} \tag{14.18}$$

where we have used that (Problem 14.3)

$$|J(x, y; r, \phi)| = \frac{1}{2}.$$

Thus, we have shown that using the transformation given in Eqs. (14.14) and (14.15), we can generate samples from normalized Gaussians.

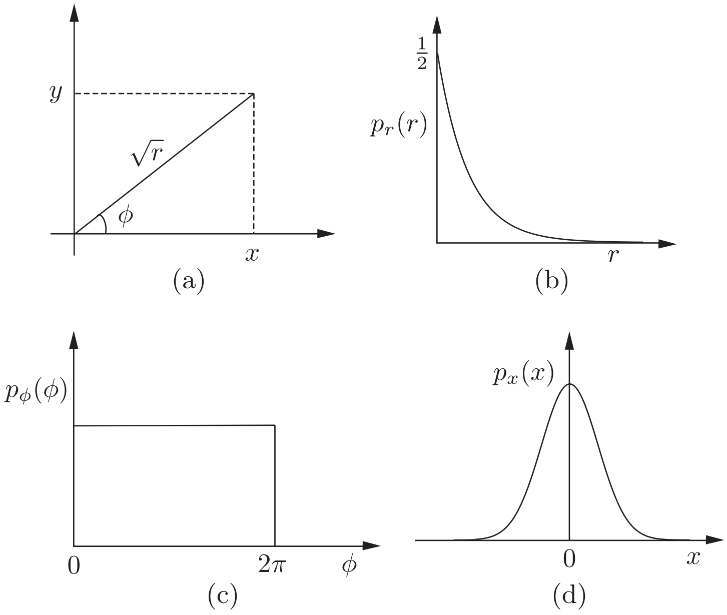


FIGURE 14.3

(a) Relation of the cartesian (x, y) to the polar coordinates (r, ϕ). (b) and (c) If r and ϕ are random variables following an exponential and a uniform in $[0, 2\pi]$ distributions, respectively, then x and y are independent and they both follow a normalized Gaussian, as shown in (d) for the x variable.

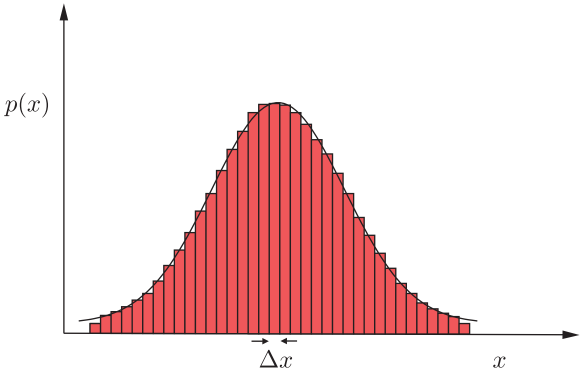


FIGURE 14.4

The histogram of $N = 100$ points generated in Example 14.3 together with the graph of $p(x) = \mathcal{N}(x|1, 0.5)$. The bin length was chosen to be equal to $\Delta x = 0.05$.

Once samples from a normalized Gaussian, $\mathcal{N}(x|0, 1)$, are available, samples from a general Gaussian, $\mathcal{N}(y|\mu, \sigma^2)$, are obtained via the obvious transformation,

$$\mathrm{y} = \sigma \mathrm{x} + \mu. \tag{14.19}$$

The previous approach is also generalized to random vectors in $\mathbb{R}^l$. One can first draw samples from $\mathbf{x} \sim \mathcal{N}(\boldsymbol{x}|\mathbf{0}, I)$ by stacking together l i.i.d. samples drawn from a normalized Gaussian, $\mathcal{N}(x|0, 1)$, and then apply the transformation

$$\mathbf{y} = L\mathbf{x} + \boldsymbol{\mu},$$

which is equivalent with drawing samples from

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}, \Sigma), \tag{14.20}$$

where $\Sigma = LL^T$ (Cholesky factorization, Problem 14.4).

Example 14.3. Generate $N = 100$ samples, r_n, $n = 1, 2, \ldots, 100$, from the exponential distribution in Eq. (14.11) (following Example 14.1) and $N = 100$ samples, ϕ_n, $n = 1, 2, \ldots, 100$, from the uniform in Eq. (14.12). Then use the transformations in Eqs. (14.14), (14.15) and (14.19) to obtain samples, x_n, $n = 1, 2, \ldots, 100$, from $p(x) = \mathcal{N}(x|1, 0.5)$. The histogram of the obtained samples is shown in Figure 14.4.

14.4 REJECTION SAMPLING

Applying the previously reported transformation techniques relies on having the involved transform functions available in a convenient (analytic) form, which in general is the exception instead of the rule. From now on, we turn our attention to alternative methods.

Rejection Sampling (e.g., [7, 37]) is conceptually a simple technique; in order to generate independent samples from a desired pdf, $p(\boldsymbol{x})$, one draws samples from another one, say, $q(\boldsymbol{x})$, that is easier to handle, and then, instead of applying a transformation, some of the points are *rejected* according to an appropriate criterion.

Given two random variables, x and u, recall that the marginal, $p(x)$, is obtained by integrating the joint pdf, $p_{\mathrm{x,u}}(x,u)$, that is,

$$p(x) = \int_{-\infty}^{+\infty} p_{\mathrm{x,u}}(x,u)\, \mathrm{d}u. \tag{14.21}$$

Let us now consider the following identity,

$$p(x) \equiv \int_0^{p(x)} 1\, \mathrm{d}x = \int_{-\infty}^{+\infty} \chi_{[0,p(x)]}(u)\, \mathrm{d}u, \tag{14.22}$$

where $\chi_{[0,p(x)]}(\cdot)$ is our familiar characteristic function in the interval $[0, p(x)]$, that is,

$$\chi_{[0,p(x)]}(u) = \begin{cases} 1 & 0 \leq u \leq p(x), \\ 0 & \text{otherwise.} \end{cases}$$

Comparing Eqs. (14.21) and (14.22), it turns out that $\chi_{[0,p(x)]}(u)$ can be interpreted as the joint pdf of the couple (x, u) defined over the set

$$\mathcal{A} = \Big\{(x,u): \ x \in \mathbb{R}, \ 0 \leq u \leq p(x)\Big\}. \tag{14.23}$$

Looking more carefully at $p_{\mathrm{x,u}}(x,u) = \chi_{[0,p(x)]}(u)$, it does not take long to realize that this is the *uniform density* under the area of the graph $u = p(x)$, as seen in Figure 14.5a. In other words, if one fills in the shaded area in Figure 14.5a uniformly at random with points (x,u) and then neglects the u dimension, then the obtained points are samples drawn from $p(x)$. We can now go one step further and assume that $p(x)$ is not exactly known, that is,

$$p(x) = \frac{1}{Z}\phi(x),$$

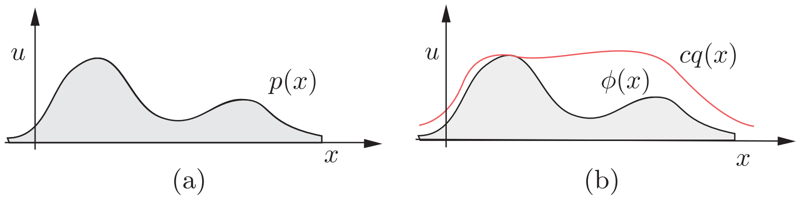


FIGURE 14.5

(a) Filling in the shaded area uniformly at random with points (x_n, u_n), and after neglecting the coordinate u_n, is equivalent with drawing points, x_n, from $p(x)$. (b) The proposal distribution, $cq(x)$, is everywhere larger or equal to $\phi(x)$.

and that the normalizing constant is not available (as we know, often, the computation of the normalizing constant is not easy). Then, we have that

$$p(x) = \frac{1}{Z}\phi(x) = \frac{1}{Z}\int_0^{\phi(x)} \mathrm{d}u = \frac{1}{Z}\int_{-\infty}^{+\infty} \chi_{[0,\phi(x)]}(u)\,\mathrm{d}u,$$

and Z is given by

$$Z = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \chi_{[0,\phi(x)]}(u)\,\mathrm{d}u\,\mathrm{d}x.$$

Hence,

$$p(x) = \frac{\int_{-\infty}^{+\infty} \chi_{[0,\phi(x)]}(u)\,\mathrm{d}u}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \chi_{[0,\phi(x)]}(u)\,\mathrm{d}u\,\mathrm{d}x}. \tag{14.24}$$

In other words, even if $p(x)$ is not exactly known, $p(x)$ can still be obtained, this time in terms of the uniform distribution, $\chi_{[0,\phi(x)]}(x,u)$, normalized appropriately. However, rescaling the uniform does not affect the marginal. It suffices to sample uniformly at random the region $\mathcal{A}$, which now should be defined in terms of $\phi(x)$ instead of $p(x)$. What we have said so far applies also to random vectors, $\mathbf{x} \in \mathbb{R}^l$, by considering the extended space $(\boldsymbol{x}, u)$, and we talk about the volume under the surface $\phi(\boldsymbol{x})$ (or $p(\boldsymbol{x})$).

We now turn our attention to see how one can fill in the volume under the surface formed by $u = \phi(\boldsymbol{x})$ (or $u = p(\boldsymbol{x})$ if it is fully available), with points uniformly at random. Let $q(\boldsymbol{x})$ be a distribution from which we know how to draw samples, and we refer to it as the *proposal distribution*. We select a constant c, such that[2]

$$\phi(\boldsymbol{x}) \le cq(\boldsymbol{x}), \ \forall \boldsymbol{x} \in \mathbb{R}^l.$$

The respective geometry is shown in Figure 14.5b. The goal is to draw points in the interval $[0, cq(\boldsymbol{x})]$ and then keep only those that lie in the region under the surface $u = \phi(\boldsymbol{x})$. The following algorithm does the job.

Algorithm 14.2 (Rejection sampling).

```
• For $i = 1, 2, \ldots, N$, Do
    • Draw $x_i \sim q(\boldsymbol{x})$
    • Draw $u_i \sim \mathcal{U}(0, cq(x_i))$
    • Retain the sample if
          $u_i \le \phi(x_i).$
• End For
```

The probability of accepting a point, $\boldsymbol{x}$, is given by

$$\text{Prob}\,\{\text{u} \le \phi(\boldsymbol{x})\} = \frac{1}{cq(\boldsymbol{x})}\phi(\boldsymbol{x}),$$

[2] If $q(\boldsymbol{x}) = 0$ in an interval, then $\phi(\boldsymbol{x})$ should be zero there.

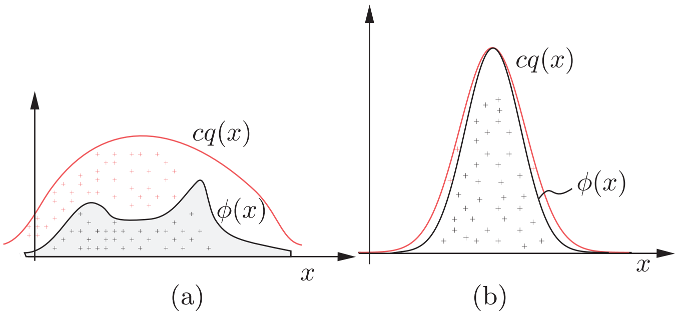


FIGURE 14.6

(a) If $cq(x)$ is much larger than $\phi(x)$ most of the samples are rejected (red points) which is inefficient. (b) If $cq(x)$ and $\phi(x)$ have a good match, most of the samples are retained.

and the total probability, over all the possible values of $\boldsymbol{x}$, for accepting samples is equal to

$$\text{Prob}\{\text{acceptance}\} = \frac{1}{c}\int \frac{\phi(\boldsymbol{x})}{q(\boldsymbol{x})} q(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \frac{1}{c}\int \phi(\boldsymbol{x}).$$

Hence, if c has a large value, only a small percentage of points is finally retained. In order to have a practical algorithm, $cq(\boldsymbol{x})$ must be chosen in order to be a good fit of $\phi(\boldsymbol{x})$. Figure 14.6a is an example of a bad choice, while Figure 14.6b corresponds to a good example. This is a reason that rejection sampling *does not scale* well with *dimensionality*. In high dimensions, guaranteeing that $cq(\boldsymbol{x}) \geq \phi(\boldsymbol{x})$ may oblige us to select a c with an excessively large value (Problem 14.5).

Besides the basic rejection scheme, a number of variants have also been proposed, in order to overcome the difficulty of selecting a proposal distribution that "looks like" the desired one. *Adaptive rejection sampling* is such a technique (see, e.g., [11] and the references therein). According to this method, the proposal distribution is adaptively constructed, based on the derivatives of $\ln p(x)$. For log-concave functions, this is a nondecreasing function and can be used to construct an envelope function of $p(x)$.

Although rejection sampling is not appropriate for difficult tasks, still it has been used, sometimes in its more refined forms, to generate samples from a number of standard distributions, such as the Gaussian, gamma, and student's-t (see, e.g., [22]).

14.5 IMPORTANCE SAMPLING

Importance sampling (IS) is a method for estimating expectations. Let $f(\mathbf{x})$ be a known function of a random vector variable, $\mathbf{x}$, which is distributed according to $p(\boldsymbol{x})$. If one could draw samples from $p(\boldsymbol{x})$, then the expectation in Eq. (14.1) could be approximated as in Eq. (14.2). We will now assume that we are not able to draw samples from $p(\boldsymbol{x})$, and to go one step further, assume that $p(\boldsymbol{x})$ is only known up to a normalizing constant, that is,

$$p(\boldsymbol{x}) = \frac{1}{Z}\phi(\boldsymbol{x}).$$

Let $q(\boldsymbol{x})$ be another distribution from which samples can be drawn. Then we can write

$$\mathbb{E}\left[f(\mathbf{x})\right] = \frac{1}{Z}\int_{-\infty}^{\infty} f(\boldsymbol{x})\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \frac{1}{Z}\int_{-\infty}^{\infty} f(\boldsymbol{x})\frac{\phi(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

$$\simeq \frac{1}{NZ}\sum_{i=1}^{N} f(\boldsymbol{x}_i)w(\boldsymbol{x}_i), \tag{14.25}$$

where $\boldsymbol{x}_i,\ i = 1, 2, \ldots, N$, are samples drawn from $q(\boldsymbol{x})$ and

$$w(\boldsymbol{x}) := \frac{\phi(\boldsymbol{x})}{q(\boldsymbol{x})}. \tag{14.26}$$

The normalizing constant can readily be obtained as

$$Z = \int_{-\infty}^{\infty} \phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{-\infty}^{\infty}\left(\frac{\phi(\boldsymbol{x})}{q(\boldsymbol{x})}\right)q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \simeq \frac{1}{N}\sum_{i=1}^{N} w(\boldsymbol{x}_i). \tag{14.27}$$

Combining Eqs. (14.25) and (14.27), we finally obtain

$$\mathbb{E}\left[f(\mathbf{x})\right] \simeq \frac{\sum_{i=1}^{N} w(\boldsymbol{x}_i)f(\boldsymbol{x}_i)}{\sum_{i=1}^{N} w(\boldsymbol{x}_i)}, \tag{14.28}$$

or

$$\boxed{\mathbb{E}\left[f(\mathbf{x})\right] \simeq \sum_{i=1}^{N} W(\boldsymbol{x}_i)f(\boldsymbol{x}_i) : \quad \text{Importance Sampling Approximation,}}$$

where $W(\boldsymbol{x}_i) = \frac{w(\boldsymbol{x}_i)}{\sum_{i=1}^{N} w(\boldsymbol{x}_i)}$ are the normalized weights. It is not difficult to show (Problem 14.6) that the estimate

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N} w(\boldsymbol{x}_i) \tag{14.29}$$

corresponds to an unbiased estimator of the normalizing constant. This is very interesting, because computing the normalizing constant is particularly useful information in a number of tasks. Recall that the evidence function, discussed in Chapter 12, is a normalizing constant; see also [26] for related comments.

In contrast, the estimator associated with Eq. (14.28), being the result of a ratio, is unbiased only asymptotically and it is a *biased* one for finite values of N (Problem 14.6). Hence, if one would have the luxury of a very large number N of samples, Eq. (14.28) would be a good enough estimate. However, in practice, N cannot be made arbitrarily large and the resulting estimate may not be satisfactory.

If $q(\boldsymbol{x}) \simeq p(\boldsymbol{x})$, or at least $q(\boldsymbol{x})$ is a fairly good approximation of $\phi(\boldsymbol{x})$, then Eq. (14.28) would approximate Eq. (14.2). However, for most practical cases, this is not easy to obtain, especially in high-dimensional spaces. If $q(\boldsymbol{x})$ is not a good match to $\phi(\boldsymbol{x})$, it is very likely that there will be regions where $\phi(\boldsymbol{x})$ is large while $q(\boldsymbol{x})$ is much smaller. The corresponding weights will have large values, relative to those from other regions, and they will be the dominant ones in the summation (Eq. (14.28)).

The effect of it is equivalent to reducing the number, N, of samples. Moreover, it is also possible that $q(\boldsymbol{x})$ takes very small values in some regions, which makes it very likely that samples from such regions

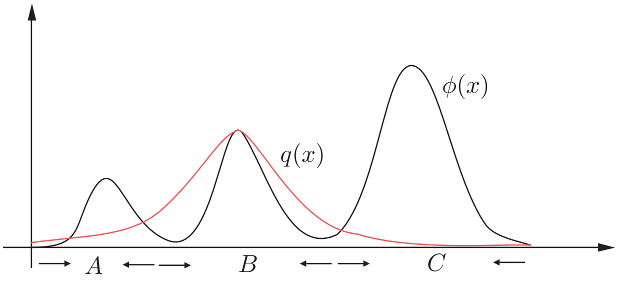


FIGURE 14.7

If $q(\boldsymbol{x})$ is not a good match of $\phi(\boldsymbol{x})$, a number of undesired effects appear. Samples from region A will give rise to weights of much larger values compared to these in region B. Due to the extremely low values of $q(\boldsymbol{x})$ in C, it is highly likely that given the finite size of the number of the samples, N, no samples will be drawn from this region, in spite of the fact that this is the most dominant region for $\phi(\boldsymbol{x})(p(\boldsymbol{x}))$.

are completely absent in Eq. (14.28); see Figure 14.7. In such cases, not only may the resulting estimate be wrong, but we will not be aware of it, and the variance of the weights, $w(\boldsymbol{x}_n)$ and $w(\boldsymbol{x}_n)f(\boldsymbol{x}_n)$, may exhibit low values. *These phenomena are accentuated in high-dimensional spaces* (see, e.g., [26] and Problem 14.7).

To alleviate the previous shortcomings, a number of variants have been proposed to search for high-probability regions and make local approximations around the modes and use them in order to generate samples (see, e.g., [30] and the references therein).

14.6 MONTE CARLO METHODS AND THE EM ALGORITHM

In Section 12.5.1, the EM algorithm was introduced for maximizing the log-likelihood function when some of the variables are hidden or missing. During the E-step (Eq. (12.61)), the function $\mathcal{Q}(\cdot,\cdot)$ is computed, which at the $(j+1)$th iteration step of the algorithms is written as

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\xi},\boldsymbol{\xi}^{(j)}) &= \mathbb{E}\left[\ln p(\mathcal{X}^l,\mathcal{X};\boldsymbol{\xi})\right] \\ &= \int p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi}^{(j)})\ln p(\mathcal{X}^l,\mathcal{X};\boldsymbol{\xi})d\mathcal{X}^l, \end{aligned} \tag{14.30}$$

where $\mathcal{X}^l$ is the set of hidden variables, $\mathcal{X}$ the set of observed values, and $\boldsymbol{\xi}$ the unknown set of parameters. In case the computation of the integral is not tractable, Monte Carlo techniques can be mobilized to generate L samples for the hidden variables, $\mathcal{X}_1^l,\dots,\mathcal{X}_L^l$, from the posterior $p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi}^{(j)})$ and obtain an approximation

$$\hat{\mathcal{Q}}(\boldsymbol{\xi},\boldsymbol{\xi}^{(j)}) \approx \frac{1}{L}\sum_{i=1}^{L}\ln p(\mathcal{X}_i^l,\mathcal{X};\boldsymbol{\xi}). \tag{14.31}$$

Maximization with respect to $\boldsymbol{\xi}$ is now carried out via $\hat{\mathcal{Q}}(\cdot,\cdot)$.

A specific form of Monte Carlo EM results in the context of mixture modeling and it is known as *stochastic EM*. The idea is to generate a *single* sample from the posterior (which now refers to the labels of the mixtures) and assign corresponding observations in the respective mixtures. That is, a hard assignment takes place. The M-step is then applied based on this approximation [3].

14.7 MARKOV CHAIN MONTE CARLO METHODS

As we have already discussed, a major drawback associated with rejection as well as importance sampling methods is that they cannot tackle tasks in high-dimensional spaces very well.

In this section, we will deal with methods that scale well with the dimensionality of the sample space. Such techniques build upon arguments that come from the theory of Markov chains; we start by presenting some definitions and basics related to this important theory. Hidden Markov models, treated in Section 16.5, are instances of Markov chains. Here we will shed more light on such models from a different perspective.

Markov chains/processes are named after the Russian mathematician Andrey Andreyevish Markov (1856–1922), who contributed seminal papers in the field of stochastic processes. As a professor at Saint Petersburg University during the students' riots in 1908, he refused the government's order to monitor and spy on his students, and he retired from the university.

Definition 14.1. A *Markov chain* is a sequence of random (vector) variables, $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2 \ldots$ with conditional distributions that obey the rule

$$p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}, \{\boldsymbol{x}_t : t \in \mathcal{I}\}) = p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}), \tag{14.32}$$

where $\mathcal{I} = \{0, 1, \ldots, n-2\}$. The index n is usually interpreted as time.

In words, Eq. (14.32) says that $\mathbf{x}_n$ is independent of the variables with indices in $\mathcal{I}$, given the values of the variables in $\mathbf{x}_{n-1}$. The distribution, p, can be either a density function or a probability distribution corresponding to discrete variables, taking values in a discrete set, known as *states*. We will assume that all variables share a common range, known as *state space*. Most of our discussion will evolve along finite state spaces, where states take values in a finite discrete set, say, $\{1, 2, \ldots, K\}$. A Markov chain is specified in terms of (a) the distribution (vector of probabilities), $\boldsymbol{p}_0$, associated with the first vector in the sequence, $\mathbf{x}_0$; and (b) the $K \times K$ matrices of the *transition probabilities*, that is,

$$P_n(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) = [P_n(i|j)],$$

where

$$P_n(i|j) := P_n(\boldsymbol{x}_n = i|\boldsymbol{x}_{n-1} = j), \quad i, j = 1, 2, \ldots, K,$$

denotes the probability of the variable at time $n-1$ to be at state j *and* the variable at time n to be at state i.[3] Given the matrix of the transition probabilities, we write

$$\boldsymbol{p}_n = P_n(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})\boldsymbol{p}_{n-1}, \tag{14.33}$$

where

$$\boldsymbol{p}_n := [P(\boldsymbol{x}_n = 1), P(\boldsymbol{x}_n = 2), \ldots, P(\boldsymbol{x}_n = K)]^T \tag{14.34}$$

$$:= [P_n(1), \ldots, P_n(K)]^T \tag{14.35}$$

[3] Note that equality here, $\boldsymbol{x}_n = i$, means that the (vector) variable $\mathbf{x}_n$ is at state i.

is the vector of the respective probabilities at time n. The Markov chain is said to be *homogeneous* or *stationary* if the transition matrix is independent on time, that is,

$$P_n(\boldsymbol{x}_n = i|\boldsymbol{x}_{n-1} = j) = P(i|j) := P_{ij}, \quad i, j = 1, 2, \ldots, K,$$

and

$$P_n(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) = P = \left[P_{ij}\right].$$

In this case, we can write

$$\boldsymbol{p}_n = P\boldsymbol{p}_{n-1} = P^2\boldsymbol{p}_{n-2} = \cdots = P^n\boldsymbol{p}_0, \tag{14.36}$$

or equivalently,

$$P_n(i) = \sum_{j=1}^{K} P_{ij}P_{n-1}(j). \tag{14.37}$$

In the sequel, we will focus on stationary Markov chains.

Properties of the Transition Probabilities Matrix. The transition matrix has a special structure leading to certain properties, which will be used later on.

- The matrix P is a *stochastic* matrix. That is, all its entries are nonnegative, and the entries across each column add to one, that is,

  $$\sum_{i=1}^{K} P_{ij} = 1,$$

  which is a direct consequence of the definition of probabilities.
- The value $\lambda = 1$ is always an eigenvalue of P (Problem 14.8). Moreover, there is no eigenvalue with magnitude larger than one (Problem 14.9).
- The eigenvectors corresponding to eigenvalues $\lambda \neq 1$ comprise components that add to zero (Problem 14.10).
- The left eigenvector corresponding to $\lambda = 1$,

  $$\boldsymbol{b}_1^T P = \boldsymbol{b}_1^T, \tag{14.38}$$

  has all its elements equal. This is easily verified by plugging in $\boldsymbol{b}_1 = [1, 1, \ldots, 1]^T$ and checking that this is indeed an eigenvector.
- *Invariant distribution.* A distribution is said to be invariant over the states of a Markov chain if

  $$\boldsymbol{p} = P\boldsymbol{p}.$$

  Note that $\boldsymbol{p}$ is necessarily an eigenvector corresponding to the eigenvalue $\lambda = 1$. Moreover, because $\boldsymbol{p}$ consists of probabilities, its elements must add to one. Depending on the multiplicity of $\lambda = 1$, there may be more than one invariant distribution. For example, if $P = I$ any probability distribution is an invariant for the respective Markov chain. It turns out that any Markov chain with a finite number of states has at least one invariant distribution. However, if the elements of P are strictly positive, then there is a unique invariant distribution that coincides with the unique eigenvector corresponding to the maximum eigenvalue, $\lambda = 1$, which in this case has a multiplicity of one. Furthermore, the eigenvector corresponding to $\lambda = 1$ comprises positive elements, which after scaling can always be made to add to one, and it is a by-product of the celebrated

Perron-Frobenius theorem, which is assured if P has strictly positive entries[4] (see, e.g., [32]). We will cover invariant distributions in more detail soon.

- *Detailed Balanced Condition*. Let P be the transition probability matrix of a stationary Markov chain. Let also $\boldsymbol{p} = [P_1, \ldots, P_K]^T$ be the set of probabilities describing a discrete distribution. We say that the *detailed balanced condition* is satisfied if

$$P(i|j)P_j = P(j|i)P_i. \tag{14.39}$$

That is, there exists a type of *symmetry*. If this condition holds, then the respective distribution is invariant for the Markov chain. Indeed,

$$\sum_{j=1}^{K} P(i|j)P_j = \sum_{j=1}^{K} P(j|i)P_i = P_i, \tag{14.40}$$

or

$$\boldsymbol{p} = P\boldsymbol{p}. \tag{14.41}$$

Although this is not a necessary condition for distribution invariance, it is very useful in practice; it helps us construct Markov chains with a desired invariant distribution. As we will soon see, this will be the type of distributions from which we want to draw samples.

14.7.1 ERGODIC MARKOV CHAINS

We now turn our attention to a specific type of Markov chains, which are known as *ergodic*. Such chains have a *unique* invariant distribution, which can be obtained as the limit

$$\lim_{n\to\infty} \boldsymbol{p}_n = \lim_{n\to\infty} P^n \boldsymbol{p}_0,$$

which is *independent* of the choice of the initial values in $\boldsymbol{p}_0$. We will now focus on a class of ergodic processes and elaborate on their convergence.

Let us consider a stationary Markov chain, with a transition matrix P, with eigenvalues $1 = \lambda_1 > |\lambda_2| \geq \cdots \geq |\lambda_K|$. That is, only one eigenvalue has the maximum value and the rest have magnitude strictly less than one. Moreover, we assume that one can find a complete set of *linearly independent* eigenvectors. Such assumptions are not restrictive and hold true for a wide class of stochastic matrices. Then, we can write that

$$P = A\Lambda A^{-1}, \tag{14.42}$$

where Λ is the diagonal matrix $\Lambda = \text{diag}\{1, \lambda_2, \ldots, \lambda_K\}$ and A has as columns the respective eigenvectors. Hence, from Eq. (14.36) we get

$$\boldsymbol{p}_n = A\Lambda^n A^{-1}\boldsymbol{p}_0 = A \begin{bmatrix} 1 & & & \\ & \lambda_2^n & & \text{O} \\ \text{O} & & \ddots & \\ & & & \lambda_K^n \end{bmatrix} A^{-1}\boldsymbol{p}_0,$$

with $\lambda_k^n \longrightarrow 0, k = 2, \ldots, K$.

[4] This is also true for a class of matrices with nonnegative elements, known as *primitive* matrices. That is, there exists an n such that P^n has positive elements.

Hence,

$$\boldsymbol{p}_\infty := \lim_{n\to\infty} \boldsymbol{p}_n = P_\infty \boldsymbol{p}_0, \tag{14.43}$$

where

$$P_\infty = A \begin{bmatrix} 1 & & \\ & 0 & \mathrm{O} \\ \mathrm{O} & \ddots & \\ & & 0 \end{bmatrix} A^{-1} = \boldsymbol{a}_1 \boldsymbol{b}_1^T, \tag{14.44}$$

with $\boldsymbol{a}_1$ being the first eigenvector (first column of A), corresponding to $\lambda = 1$ and $\boldsymbol{b}_1^T$ the first row of A^{-1}. In other words, P_∞ is a rank-one matrix. However, it is straightforward to see from Eq. (14.42) $\left(A^{-1}P = \Lambda A^{-1}\right)$ that $\boldsymbol{b}_1^T$ is a left eigenvector of P, that is,

$$\boldsymbol{b}_1^T P = \boldsymbol{b}_1^T,$$

and recalling the properties (Eq. (14.38) and the comments just after it) of P, $\boldsymbol{b}_1^T = [1, 1, \dots, 1]$ (within a proportionality constant, c). Thus,

$$P_\infty = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_1],$$

and from Eq. (14.43), because the components of $\boldsymbol{p}_0$ add to one, we finally obtain that

$$\boldsymbol{p}_\infty = \boldsymbol{a}_1.$$

That is, the limiting distribution is equal (after scaling) to the unique eigenvector of P corresponding to $\lambda_1 = 1$; moreover, this is true irrespective of the values of $\boldsymbol{p}_0$. In other words, the limiting distribution is the invariant distribution of P, that is,

$$P\boldsymbol{p} = \boldsymbol{p}. \tag{14.45}$$

Note that the convergence rate is controlled by the magnitude of $|\lambda_2|$. Other, more theoretically refined convergence results and bounds can be found in, for example, [24, 39, 40].

Remarks 14.2.

- Needless to say, not all Markov chains are ergodic. For example, if the eigenvalue $\lambda_1 = 1$ of the transition matrix has multiplicity higher than one, then the limiting distribution depends on the values of the initial choice in $\boldsymbol{p}_0$. On the other hand, if the transition matrix has more than one eigenvalue with magnitude equal to one (e.g., $\lambda_1 = 1, \lambda_2 = -1$) then, again, it does not have a limiting distribution but instead exhibits a *periodic* limit cycle (e.g., [32]).
- *Building Markov Chains*. In practice, one can construct transition probability matrices for ergodic chains using a set of simpler transition matrices, $B_1, B_2, \dots, B_M$, which are known as *base transition* matrices. Each one of them may not be ergodic, but it is required to accept the desired distribution as its invariant. Then, the transition matrix is built as

$$P = \sum_{m=1}^{M} \alpha_m B_m, \quad \alpha_m > 0, \ \sum_{m=1}^{M} \alpha_m = 1.$$

If a distribution is invariant with respect to each B_m, $m = 1, 2, \dots, M$, it will be invariant for P. The same applies with the detailed balance condition.

Another way is to combine individual transition matrices sequentially, that is,

$$P = B_1 B_2 \cdots B_M.$$

For example, each B_m, $m = 1, 2, \ldots, M$, may act and change a subset of the random entries comprising the random vector $\mathbf{x}$. We will see that this is the case with the Gibbs sampling method, to be reviewed soon. It is easy to see that if $\boldsymbol{p}$ is invariant for each individual B_m, $m = 1, 2, \ldots, M$, will also be invariant for P.

- In this section, we focused our discussion on Markov chains with finite state space. Everything we have said can be generalized to Markov chains with countably infinite or continuous state spaces. In the latter case, the place of the probability transition matrix is taken by the transition density or kernel, $p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$, and the probability density of $\mathbf{x}_n$, at time n, is given by

$$p_n(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{y}) p_{n-1}(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}. \qquad (14.46)$$

The analysis in this case is more difficult and care has to be taken because not all results obtained for the finite discrete case are readily valid for the continuous one. The reason we focused on the discrete finite state space is that one can get the feeling of the theory of Markov chains by spending less "budget" on the required mathematical effort.

Example 14.4. Consider the Markov chain with the transition probability matrix

$$P = \begin{bmatrix} 0.2 & 0.4 & 0.6 \\ 0.5 & 0.1 & 0.3 \\ 0.3 & 0.5 & 0.1 \end{bmatrix}.$$

Its eigenvalues are $\lambda_1 = 1$, $\lambda_2 = -0.3 + 0.1732j$, $\lambda_3 = -0.3 - 0.1732j$, and the respective eigenvectors are

$$\boldsymbol{a}_1 = [0.6608,\ 0.5406,\ 0.5206]^T,$$
$$\boldsymbol{a}_2 = [0.5774,\ -0.2887 - 0.5j,\ -0.2887 + 0.5j]^T,$$
$$\boldsymbol{a}_3 = [0.5774,\ -0.2887 + 0.5j,\ -0.2887 - 0.5j]^T.$$

Observe that all the elements of the eigenvector corresponding to $\lambda = 1$ are positive. Also, the elements of the other two eigenvectors add to zero.

We can now write that

$$P = \begin{bmatrix} 0.6608 & 0.5774 & 0.5774 \\ 0.5406 & -0.2887 - 0.5j & -0.2887 + 0.5j \\ 0.5206 & -0.2887 + 0.5j & -0.2887 - 0.5j \end{bmatrix}$$
$$\times \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.3 + 0.1732j & 0 \\ 0 & 0 & -0.3 - 0.1732j \end{bmatrix}$$
$$\times \begin{bmatrix} 0.5807 & 0.5807 & 0.5807 \\ 0.5337 - 0.0058i & -0.3323 + 0.4942j & -0.3323 - 0.5058j \\ 0.5337 + 0.0058i & -0.3323 - 0.4942j & -0.3323 + 0.5058j \end{bmatrix}.$$

Observe that the first row of the last matrix (A^{-1}) has all its elements equal. Having written P in a product form, it is easily obtained that

$$P^2 = \begin{bmatrix} 0.42 & 0.42 & 0.30 \\ 0.24 & 0.36 & 0.36 \\ 0.34 & 0.22 & 0.34 \end{bmatrix},$$

$$P^{10} = \begin{bmatrix} 0.3837 & 0.3837 & 0.3837 \\ 0.3140 & 0.3140 & 0.3139 \\ 0.3023 & 0.3023 & 0.3029 \end{bmatrix}.$$

The sequence has converged for $n = 10$. Note that after convergence, P^n has all its column vectors equal, and the elements add to one. Moreover, observe that

$$P_\infty \propto [\boldsymbol{a}_1, \boldsymbol{a}_1, \boldsymbol{a}_1].$$

Example 14.5. *Random walk with finite states*. Random walks are popular models that can model faithfully a number of real-world phenomena, such as thermal noise, the motion of gas molecules, and stock value variations. Moreover, such chains can help us understand the behavior of more complex Markov chains, to be discussed soon. There are various random walk models, depending on the choice of the transition probabilities (see, e.g., [32]). Here, we assume the variables to be discrete and take integer values in a finite set, $[0, N]$. Hence, the total number of states is $N + 1$. At every time instant, the value of the variable can either increase or decrease by one with probability p, respectively, or stay unchanged, with probability q, provided that the current state is in the interval $[1, N - 1]$. That is, if $0 < x_{n-1} < N$,

$$P(x_n = x_{n-1} + 1) = P(x_n = x_{n-1} - 1) = p,$$
$$P(x_n = x_{n-1}) = q.$$

If $x_{n-1} = 0$, then x_n can either stay in the same state with probability q_e or increase by one with probability p. If $x_{n-1} = N$, then x_n can either stay in the same state with probability q_e or decrease by one with probability p. Obviously,

$$2p + q = 1, \quad p + q_e = 1.$$

The transition probability matrix, for the case of $N = 4$, $p = \frac{1}{4}$, $q = \frac{1}{2}$, $q_e = \frac{3}{4}$, is

$$P = \begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 0 & 1/4 & 1/2 & 1/4 \\ 0 & 0 & 0 & 1/4 & 3/4 \end{bmatrix}.$$

The respective eigenvalues are $\lambda_1 = 1$, $\lambda_2 = 0.904$, $\lambda_3 = 0.654$, $\lambda_4 = 0.345$, $\lambda_5 = 0.095$. Observe that all eigenvalues, except $\lambda_1 = 1$, have magnitude less than one. The corresponding eigenvectors are

$$\boldsymbol{a}_1 = [0.447,\ 0.447,\ 0.447,\ 0.447,\ 0.447]^T$$
$$\boldsymbol{a}_2 = [-0.601,\ -0.371,\ 0,\ 0.371,\ 0.601]^T$$

$$\boldsymbol{a}_3 = [-0.511,\ 0.195,\ 0.632,\ 0.195,\ -0.511]^T$$
$$\boldsymbol{a}_4 = [-0.371,\ 0.6015,\ 0,\ -0.601,\ 0.371]^T$$
$$\boldsymbol{a}_5 = [0.195,\ -0.511,\ 0.632,\ -0.511,\ 0.195]^T.$$

The eigenvector corresponding to λ_1 has all its components equal and positive. Hence, the invariant distribution ($\boldsymbol{p}$:$P\boldsymbol{p} = \boldsymbol{p}$), after the required scaling, becomes the uniform one, $\boldsymbol{p} = [1/5,\ 1/5,\ 1/5,\ 1/5,\ 1/5]^T$. Similar arguments apply for any value of N. Observe that the components of all the other eigenvectors add to zero.

Figure 14.8 shows the probability distribution $\boldsymbol{p}_n$ for the case of $N = 4$, at times $n = 10,\ 50$, and 100. The components of $\boldsymbol{p}_0$ were randomly chosen. Figure 14.9 corresponds to the case of $N = 9$. Observe that the larger the value of N, the slower the convergence.

Example 14.6. In this example, we consider a random walk with (countable) infinite many states; at every time instant, the value of the random variable can either increase or decrease by one, with probability p, respectively, or stay in the same state with probability q, that is,

$$P(x_n = x_{n-1} + 1) = P(x_n = x_{n-1} - 1) = p,$$
$$P(x_n = x_{n-1}) = q,$$

and

$$2p + q = 1.$$

The difference with the previous example is that now there are not "barrier" points and the random variable can take any integer value. Our goal is to compute the mean and variance as functions of time n, when the starting point is deterministically chosen to be $x_0 = 0$.

It is readily seen that $\mathbb{E}[\mathrm{x}_n] = 0$, because the variable is equally likely to increase or decrease and hence it is equally likely to assume any positive or negative value.

For the variance, we obtain (Problem 14.11)

$$\begin{aligned}\mathbb{E}[\mathrm{x}_n^2] &= \mathbb{E}[\mathrm{x}_{n-1}^2] + 2p \\ &= 2pn + \mathbb{E}[\mathrm{x}_0^2] = 2pn,\end{aligned} \qquad (14.47)$$

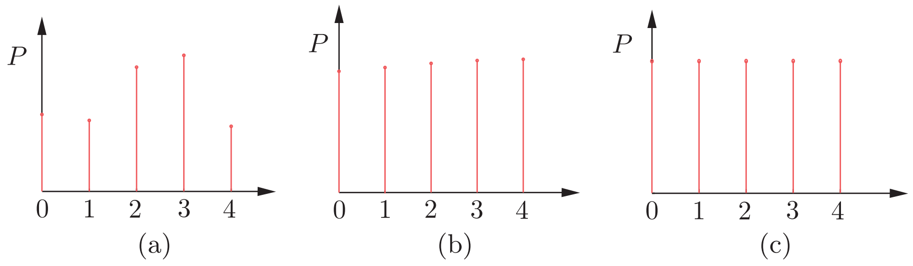


FIGURE 14.8

The probability distribution for the random walk chain of Example 14.5 for $N = 4$ and at time instants (a) $n = 10$, (b) $n = 50$, and (c) $n = 100$.

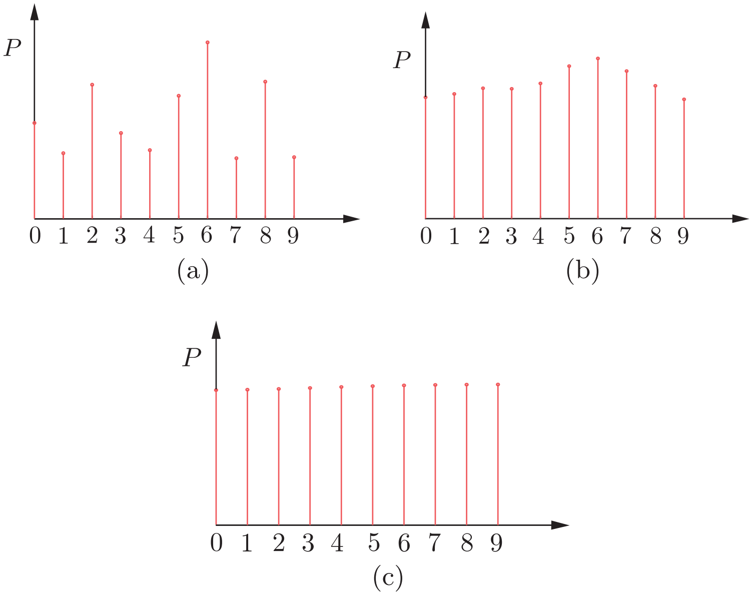


FIGURE 14.9

The probability distribution for the random walk chain of Example 14.5 for $N = 9$ and at time instants (a) $n = 10$, (b) $n = 50$, and (c) $n = 100$. Compared to Figure 14.8, observe that the higher the number N the slower the convergence.

because $\mathbb{E}[x_0^2] = 0$. Note that the variance tends to infinity with time, hence the infinite state-space random walk *does not* have a limiting distribution. This verifies what we said before; results that hold true for finite state-spaces do not necessarily carry on to the case where the number of states becomes infinite.

Looking at Eq. (14.47) more carefully reveals that, on average, after n time instants, x_n would be within $\pm\sqrt{2pn}$. If x_n denotes the distance of a point from the origin from where it starts and moves backward or forward, then the distance it travels is proportional only to the square root of the time it has spent traveling. Although this result has been derived for the infinite state-space case, still it can shed light on the slow convergence to the invariant distribution that we saw in the previous example, for the case of finite state space. As stated in [30], convergence to the invariant distribution can be achieved once all points in the state space have been visited, and this has a square root dependence on time. In order to get a good enough approximation of the limited distribution, one must be patient enough to compute $\mathcal{O}(N^2)$ iterations.

14.8 THE METROPOLIS METHOD

The Metropolis method or algorithm, as it is sometimes called, builds upon a surprisingly simple idea, and it is the first method that exploited the Markov chain theory for sampling. It appeared in the classical paper [27] and it may be the most popular and widely known sampling technique, which has inspired a wealth of variants. In contrast to rejection and importance sampling, the proposal distribution is now time varying, following the evolution of a Markov chain; the latter is constructed such as its transition probability matrix/density kernel to have the desired distribution $p(\boldsymbol{x})$ as its invariant. Moreover, in contrast to the rejection and importance sampling techniques, it is not required that the proposal distribution "look like" the desired one in order for the method to be useful in practice. The proposal distribution depends on the value of previous state, $\boldsymbol{x}_{n-1}$, that is, $q(\cdot|\boldsymbol{x}_{n-1})$. In words, drawing a new sample (generating a new state) depends on the value of the previous one. In its original version, the proposal distribution was chosen to be symmetric, that is,

$$q(\boldsymbol{x}|\boldsymbol{y}) = q(\boldsymbol{y}|\boldsymbol{x}).$$

Later on, it was generalized by Hastings [14] to include nonsymmetric ones. The general scheme is known as the Metropolis-Hastings algorithm, which is summarized next.

Algorithm 14.3 (Metropolis-Hastings algorithm).

```
• Let the desired distribution be $p(\cdot) = \frac{1}{Z}\phi(\cdot)$.
• Choose the proposal distribution to be $q(\cdot|\cdot)$.
• Choose the value of the initial state $\boldsymbol{x}_0$.
• For $n = 1, 2, \ldots, N$, Do
    • Draw $\boldsymbol{x} \sim q(\cdot|\boldsymbol{x}_{n-1})$
    • Compute the acceptance ratio
          $\alpha(\boldsymbol{x}|\boldsymbol{x}_{n-1}) = \min\left\{1, \frac{q(\boldsymbol{x}_{n-1}|\boldsymbol{x})\phi(\boldsymbol{x})}{q(\boldsymbol{x}|\boldsymbol{x}_{n-1})\phi(\boldsymbol{x}_{n-1})}\right\}$
    • Draw
          $u \sim \mathcal{U}(0, 1)$
    • If   $u \leq \alpha(\boldsymbol{x}|\boldsymbol{x}_{n-1})$
        - $\boldsymbol{x}_n = \boldsymbol{x}$
    • Else
        - $\boldsymbol{x}_n = \boldsymbol{x}_{n-1}$
• End For
```

The following points are readily deduced from the algorithm:

- The algorithm does not need the exact form of p. It suffices to know it up to its normalizing constant Z. This is due to the fact that p enters into the algorithm only in the ratio for computing the acceptance ratio.
- If the proposal distribution is symmetric, the acceptance ratio becomes

$$\alpha(\boldsymbol{x}|\boldsymbol{x}_{n-1}) = \min\left\{1, \frac{\phi(\boldsymbol{x})}{\phi(\boldsymbol{x}_{n-1})}\right\}, \tag{14.48}$$

  and in this case, we sometimes refer to it as the Metropolis algorithm.
- Note that if a sample is not accepted, we retain the value of the previous state.
- Observe that a sample is accepted or rejected depending on the value of $\alpha(\boldsymbol{x}|\boldsymbol{x}_{n-1})$. This is easier understood by looking at the original form of the algorithm based on Eq. (14.48). If the probability $p(\boldsymbol{x})$ is larger than $p(\boldsymbol{x}_{n-1})$, then the new sample is accepted. If not, it is accepted/rejected based on its relative value.
- Successive samples are *not* independent.

There are variants of the previous basic scheme, concerning the choice of the function for the acceptance ratio. In [33], an argument in support of the rationale behind the Metropolis-Hastings scheme is based on an optimality proof concerning the variance of the obtained approximations.

Let us now turn our focus to understanding how the previously stated algorithm relates to the Markov chain theory. We will work with the more general continuous state-space models, and define

$$p(\boldsymbol{x}|\boldsymbol{y}) = q(\boldsymbol{x}|\boldsymbol{y})\alpha(\boldsymbol{x}|\boldsymbol{y}) + \delta(\boldsymbol{x}-\boldsymbol{y})r(\boldsymbol{x}), \tag{14.49}$$

where $r(\boldsymbol{x})$ is the rejection probability,

$$r(\boldsymbol{x}) = \int (1-\alpha(\boldsymbol{x}|\boldsymbol{y}))q(\boldsymbol{x}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}, \tag{14.50}$$

and $\delta(\cdot)$ is Dirac's delta function. A little thought reveals that $p(\cdot|\cdot)$, as defined above, is the transition density kernel (transition matrix for finite discrete spaces), $p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$, for an equivalent Markov chain. Moreover, this Markov chain has the desired distribution $p(\boldsymbol{x})$ as its invariant, that is,

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y},$$

which, as already pointed out in Section 14.7, is a direct outcome of the fact that the following detailed balance condition is satisfied (Problem 14.12):

$$p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}).$$

It turns out that the equivalent Markov chain is ergodic, hence converging to the invariant (desired) distribution, provided that $p(\boldsymbol{x}|\boldsymbol{y})$ as well as $p(\boldsymbol{x})$ are *strictly positive*. This guarantees that any state has a nonzero probability to be reached starting from any state.

Hence, the Metropolis-Hastings algorithm equivalently draws samples from the Markov chain defined by the transition density given in Eq. (14.49), albeit the samples are drawn from the chosen (easily sampled) proposal distribution. Typical distributions used to play the role of the proposal distribution are the Gaussian and Cauchy distributions. The latter, due to its heavy-tail property, allows large changes to occur from time to time. Sometimes, the uniform distribution is also used. For the discrete case, the uniform distribution seems to be a popular choice.

Burn-in phase: After convergence, the process becomes equivalent with drawing samples from the desired $p(\boldsymbol{x})$! However, nothing is perfect in this world; a major weakness of the Markov chain Monte Carlo techniques is that it is difficult to assess whether the Markov chain *has converged*, and hence to be sure that the samples one generates are indeed effectively independent and truly representative of $p(\boldsymbol{x})$. Samples generated before the chain has converged are not representative of the desired distribution and have to be rejected; this is known as the *burn-in phase*. The interval that a Markov chain takes to converge is known as the *mixing time* (e.g., [21]).

To this end, a number of diagnostics have been proposed, though none of them can be considered a panacea (see, e.g., [6, 8, 35]) for a discussion. A theoretical justification concerning the difficulty of accessing convergence of such techniques is provided in [2], where it is shown that this is a computationally intractable task.

In practice, after the rejection of the samples during the burn-in phase, one runs a long chain and discards one out of, say, M samples. For large enough values of M, one expects to obtain independent samples. This process is also known as *thinning*. An alternative path is to run a few, say three to four, different (starting from different initial points) chains of medium size (e.g., 100,000) and take samples from each of them, having discarded the samples in the respective burn-in phases (e.g., the first half of them).

14.8.1 CONVERGENCE ISSUES

When dealing with the rejection and importance sampling, we discussed that these methods do not scale well with dimensionality. In contrast, the Metropolis approach shows much better behavior, and it is an algorithm that lends itself to applications in large spaces. Having said that, the method is not without shortcomings. To elaborate, we will use our experience gained from the random walk examples and employ similar arguments as those given in [30].

Consider a two-dimensional task and adopt as the proposal distribution, $q(\boldsymbol{x}|\boldsymbol{x}_{n-1})$, the Gaussian one with covariance matrix $\sigma^2 I$ and each time-centered at $\boldsymbol{x}_{n-1}$. The desired distribution, from which samples are to be drawn, is another elongated Gaussian $\mathcal{N}(\boldsymbol{x}|\boldsymbol{0}, \Sigma)$, as shown in Figure 14.10a. The values $\sigma_{\max}$, $\sigma_{\min}$ denote the scales (standard deviations) associated with the two axes of the ellipse (recall from Chapter 2 that this is defined by the eigenstructure of Σ), which corresponds to the one standard deviation contour (the exponent in the Gaussian is equal to $-1/2$) of $p(\boldsymbol{x})$.

Every time a sample is drawn from $\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_{n-1}, \sigma^2 I)$, the new sample will be within the circle of radius σ around $\boldsymbol{x}_{n-1}$, with high probability. In order for the new sample to have a large chance to lie within the high-probability elliptical region, σ must be of the order of $\sigma_{\min}$ or smaller. If σ is chosen to have a large value, there is high probability for the sample to end up outside the ellipse and be rejected. Hence, once sampling starts inside the ellipse, small values of σ guarantee, with high probability, that samples remain within the ellipse, and hence are accepted. On the other hand, if σ is small, a large number of iterations will be required in order to cover sufficiently with points the interior of the ellipse. If one looks at the process of sampling as a random walk, with approximate step size σ, then the number of iterations needed to cover a scale of the order of $\sigma_{\max}$ will be $\left(\frac{\sigma_{\max}}{\sigma}\right)^2$; if $\sigma \simeq \sigma_{\min}$, this becomes $\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^2$. In high dimensions, where there is high probability for one of the dimensions to be of relatively small scale compared to the maximum one, this square-dependence rule of thumb can slow convergence substantially.

Figures 14.10b, c, d, show the case where the desired two-dimensional Gaussian has zero mean and covariance matrix given by

$$\Sigma = \begin{bmatrix} 1.00 & 0.99 \\ 0.99 & 1.00 \end{bmatrix}.$$

The proposal distribution is a Gaussian with covariance matrix $0.1I$. The figures show three snapshots in the sequence of point generation, corresponding to 50, 100, and 3000 points. The rejected ones are denoted in red. The number of rejected points in Figure 14.10d is equal to 5%. This percentage increases to 13% if the proposal distribution has covariance matrix equal to $0.3I$.

Another problem that may arise with the Metropolis method is that of *local trapping*. This may occur when the desired distribution is multimodal, which is common in high-dimensional complex

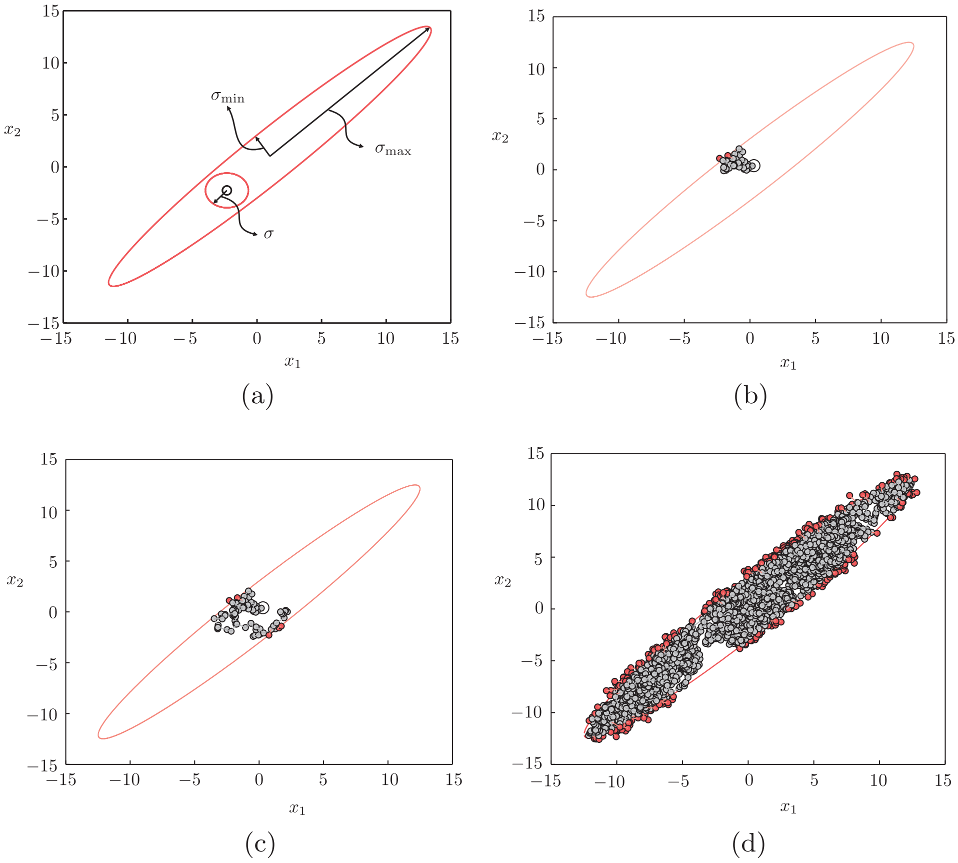


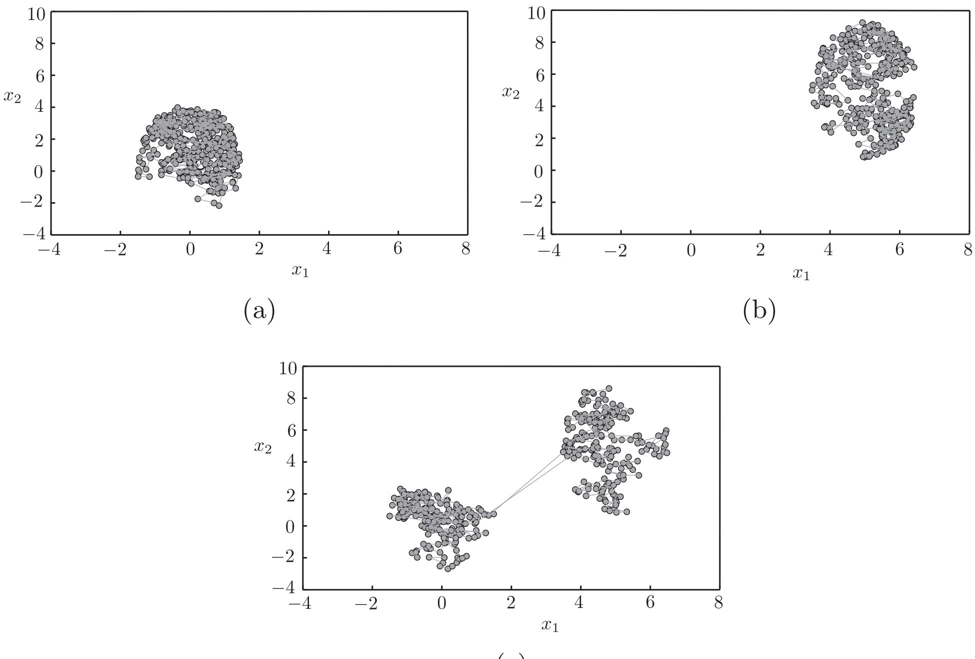
FIGURE 14.10

(a) The region of significant probability mass is enclosed by the ellipse, with scales $\sigma_{\max}$ and $\sigma_{\min}$, respectively, as defined by the major and minor axes. The region of significant probability mass of the proposal distribution is spherical of scale equal to σ, which is of the same order as $\sigma_{\min}$. (b) Starting from the point shown as a circle, 50 generated points are shown using a proposal distribution of covariance equal to $\sigma^2 = 0.1I$; rejected points are shown in red. (c) The snapshot with 100 points and d) with 3000 points. Observe that even in the latter case, still there are parts in the high-probability region of the desired distribution that have not been covered.

problems. We will demonstrate the case via a simple example in the two-dimensional space. Let the desired distribution comprise a mixture of two Gaussians,

$$p(\boldsymbol{x}) = \frac{1}{2}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_1, \Sigma_1) + \frac{1}{2}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_2, \Sigma_2),$$

where $\boldsymbol{\mu}_1 = [0,0]^T$, $\boldsymbol{\mu}_2 = [5,5]^T$, $\Sigma_1 = \Sigma_2 = \text{diag}\{0.25, 2\}$, and the proposal distribution is $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, I)$, where $\boldsymbol{\mu} = [2.5, 2.5]^T$. Figure 14.11 shows the paths traveled by the drawn and accepted points over three different runs. In Figure 14.11a, b after 400 iterations the points drawn cover only one of the two mixtures. Both mixtures are visited in the run corresponding to Figure 14.11c.



(a)

(b)

(c)

FIGURE 14.11

The desired distribution comprises the mixture of two Gaussians. In each figure, a different initialization point is selected. In all cases, 400 points have been generated. Note that in two of the cases, the process seems to be trapped in either one of the two Gaussians.

14.9 GIBBS SAMPLING

Gibbs sampling is among the most popular and widely used sampling methods. It is also known as the *heat bath* algorithm. Although Gibbs sampling was already known and used in statistical physics, two papers [9, 10] were catalytic for its widespread use in the Bayesian and machine learning communities.

Josiah Willard Gibbs (1839–1903) was an American scientist whose work on thermodynamics laid the foundations of physical chemistry. Together with James Clark Maxwell (1831–1879) and Ludwig Eduard Boltzmann (1844–1906), Gibbs pioneered the field of statistical mechanics. He is also the father, together with Oliver Heaviside (1850–1925), of what is today known as vector calculus.

Gibbs sampling is appropriate for drawing samples from multidimensional distributions, and it can be considered a special instance of the more general Metropolis method.

Let the random vector in the Markov chain at time n be given as

$$\mathbf{x}_n = [\mathrm{x}_n(1), \ldots, \mathrm{x}_n(l)]^T.$$

The basic assumption underlying Gibbs sampling is that the *conditional* distributions of each one of the variables, $x_n(d),\ d = 1, 2, \ldots, l$, given the rest, that is,

$$p\big(x_n(d)|\{x_n(i) : i \neq d\}\big), \tag{14.51}$$

are known and they can be easily sampled. At each iteration, a sample is drawn for only one of the variables, based on Eq. (14.51), by freezing the values of the rest to those already available from the previous iteration. The scheme is summarized next.

Algorithm 14.4 (Gibbs sampler).

```
• Initialize, $x_0(1), \dots, x_0(l)$, arbitrarily.
• For $n = 1, 2, \dots, N$, Do
    • For $d = 1, 2, \dots, l$, Do
        - Draw
              $$x_n(d) \sim p\left(x \mid \{x_n(i), i < d \neq 1\}, \{x_{n-1}(i),\ i > d \neq l\}\right)$$
    • End For
• End For
```

Note that in the previous scheme, all dimensions are visited in sequence. Another version is to visit them in a random order.

The Gibbs scheme can be viewed as a realization of a Markov chain, where the transition matrix/pdf is sequentially constructed from l base transitions, that is,

$$T = B_1 \cdots B_l,$$

where each individual base transition acts on the corresponding dimension, that is, coordinate-wise. For continuous variables, it can be readily checked out that

$$B_d(\boldsymbol{x}|\boldsymbol{y}) = p\left(x(d)|\{y(i)\} : i \neq d\right) \prod_{i \neq d} \delta(y(i) - x(i)), \quad d = 1, 2, \dots, l.$$

In words, only the component $x(d)$ changes while the rest are left unchanged. It is not difficult to see that the desired joint distribution $p(\boldsymbol{x}) = p(x(1), \dots, x(l))$ is invariant with respect to each one of B_d, $d = 1, 2, \dots, l$ (Problem 14.13). Hence, it will be invariant under their product

$$T = B_1 \cdots B_l.$$

Ergodicity is ensured by requiring that all conditional probabilities are *strictly* positive, which guarantees the convergence of the chain to the desired $p(\boldsymbol{x})$.

Remarks 14.3.

- Gibbs sampling, being an instance of the Metropolis method, inherits its random walk-like convergence performance.
- Gibbs sampling is appropriate for many graphical models (Chapter 16) that are described in terms of conditional distributions. Often, these distributions can be sampled in an easy way, using techniques such as rejection sampling and its variants, as discussed in Section 14.4.
- Note that in Gibbs sampling, no samples are rejected. This can also be shown if Gibbs sampling is considered as an instance of the Metropolis method, via the specific choice of Eq. (14.51) as the proposal distribution (Problem 14.14).
- *Blocking Gibbs sampling*: Gibbs sampling samples one variable at a time. This can make the algorithm move very slowly through the state-space in case the variables are highly correlated. In

such cases, it is preferable to sample *groups* of variables, not necessarily disjoint, and sample from the variables in the block, conditioned on the remaining. This is known as blocking Gibbs sampling [16], and it improves performance by achieving much bigger moves through the state-space.

- *Collapsed Gibbs Sampling*: In collapsed Gibbs sampling, one integrates out (marginalizes over) one or more variables and samples from the remaining ones. For example, in the case of three variables, Gibbs sampling samples from $p(x_1|x_2, x_3)$, then $p(x_2|x_1, x_3)$, and finally $p(x_3|x_1, x_2)$ to complete the iteration step. In collapsed Gibbs sampling, we can integrate out, for example, x_3 (which is *collapsed*), and sample sequentially from $p(x_1|x_2)$ and $p(x_2|x_1)$. Then sampling is performed in a lower dimensional space, hence it is more efficient. Collapsing one variable is tractable if it is a conjugate prior of another involved variable; for example, they are both members of the exponential family. Thus, x_3 does not participate in Gibbs sampling. In the sequel, we can sample $p(x_3|x_1, x_2)$. This can be justified by the *Rao-Blackwell* theorem, which states that the variance of the estimate created by analytically integrating out x_3 will always be lower than (or equal to) the variance of direct Gibbs sampling, [23].

14.10 IN SEARCH OF MORE EFFICIENT METHODS: A DISCUSSION

In order to sidestep the drawbacks associated with the described basic Markov chain-based schemes, namely the slow random walk-like convergence and the local-trap problem, a number of more advanced methods have been suggested. It is beyond the scope of this chapter to present such schemes in more detail and the interested reader may consult more specialized books and articles, that is [5, 22, 24, 30, 38]. Below, we provide a short discussion on some of the most popular directions that have been proposed.

A family of algorithms known as *auxiliary variable Markov chain Monte Carlo* methods is a popular one. Such methods augment with auxiliary variables either the desired or the proposal distribution in the Metropolis-Hastings algorithm. The presence of the auxiliary variable is intended to either help the algorithm to escape from possible local-traps, or to cancel out the normalizing constant, if this is intractable. Such methods include algorithms like the *simulating annealing*, [17]; the *simulated tempering*, [25]; and the *slice sampler*, [15]. The rationale behind the slice-sampling techniques builds around our discussion in Section 14.4; recall that sampling from $p(\boldsymbol{x})$ is equivalent with sampling uniformly from the region in

$$\mathcal{A} = \left\{(\boldsymbol{x}, u) : \boldsymbol{x} \in \mathbb{R}^l, \ 0 \leq u \leq p(\boldsymbol{x})\right\}.$$

In [31], a Gibbs-type implementation of the slice sampler is suggested; each component of $\boldsymbol{x}$ is updated sequentially using a single-variable slicing sampling strategy. It turns out that the slice sampler improves upon the convergence speed of the standard Metropolis-Hastings algorithm.

In [29], an auxiliary variable is used so that the computation of the normalizing constant is bypassed. This is important in cases where its computation is intractable.

Another sampling philosophy spans the *population-based* methods. In order to overcome the local-trap problem, a number (population) of Markov chains are run in parallel under an information exchange strategy, which improves convergence. Typical examples of such techniques include *adaptive direction* *sampling*, [12]; and the *evolutionary* Monte Carlo method, which builds upon arguments used in genetic algorithms [22].

Another direction is that of the *Hamiltonian* Monte Carlo methods, which exploit arguments from classical mechanics around the elegant Hamiltonian equations [26, 30]. For pdfs of the form

$$p(\boldsymbol{x}) = \frac{1}{Z_E} \exp\left(-E(\boldsymbol{x})\right),$$

$E(\boldsymbol{x})$ can be given the interpretation of the system's potential energy (Section 15.4.2). Once such a bridge has been established, an auxiliary random vector, $\boldsymbol{q}$, is introduced and interpreted as the momentum of the system; hence, the corresponding kinetic energy is expressed as

$$K(\boldsymbol{q}) = \frac{1}{2} \sum_{i=1}^{l} q_i^2.$$

The Hamiltonian function is then given by

$$H(\boldsymbol{x}, \boldsymbol{q}) = E(\boldsymbol{x}) + K(\boldsymbol{q}),$$

and it defines the distribution

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{q}) &= \frac{1}{Z_H} \exp\left(-H(\boldsymbol{x}, \boldsymbol{q})\right) \\ &= \frac{1}{Z_E} \exp\left(-E\right) \frac{1}{Z_K} \exp\left(-K(\boldsymbol{q})\right) \\ :&= p(\boldsymbol{x}) p(\boldsymbol{q}), \end{aligned}$$

where Z_K is the normalizing constant of the respective Gaussian term associated with the kinetic energy. The desired distribution, $p(\boldsymbol{x})$, is obtained as the marginal of $p(\boldsymbol{x}, \boldsymbol{q})$. Hence, if sampling from $p(\boldsymbol{x}, \boldsymbol{q})$ is possible, then discarding $\boldsymbol{q}$ results in samples drawn from the desired distribution. The evolution of the variables in time is given by the associated Hamiltonian dynamics of the equivalent system.

Such methods may lead to a substantial improvement in convergence speed; the reason is that via the Hamiltonian interpretation, information hidden in the derivatives of $E(\boldsymbol{x})$ (i.e., $\dot{\boldsymbol{q}} = -\frac{\partial E(\boldsymbol{x})}{\partial \boldsymbol{x}}$) is exploited in order for the system to detect directions toward high-probability mass.

In the *reversible jump Markov chain Monte Carlo* algorithms, the Metropolis-Hastings algorithm is extended to account for state-spaces of varying dimensionality [13]. Such methods are appropriate in cases where multiple parameter models of varying dimensionality are involved. Thus, the Markov chain is given the liberty to jump between models of different dimensionality.

Variational inference or Monte Carlo methods

In the beginning of this chapter we mentioned that the variational inference techniques, which were considered in Chapter 13, are the deterministic alternatives of the Monte Carlo methods. We will now attempt to sketch in a few lines the pros and cons of each of the two approaches. The main advantages concerning the former path to Bayesian learning are as follows:

- They are computationally more efficient for small- and medium-scale tasks.
- It is fairly easy to determine when to stop iterations and to know when convergence has been achieved.
- One can compute lower bounds for the likelihood function.

The advantages concerning Monte Carlo methods are as follows:

- They can be applied to more general cases, for example, models without computationally convenient priors, or to models whose structure is changing.
- They do not rely on approximations such as the mean-field approximation.
- They can be more efficient for large-scale tasks.

14.11 A CASE STUDY: CHANGE-POINT DETECTION

The task of change-point detection is of major importance in a number of scientific disciplines, ranging from engineering and sociology to economics and environmental studies. The accumulated literature is vast; see, for example, [1, 19, 36], and the references therein. The aim of the change-point identification task is to detect partitions in a sequence of observations, in order for the data in each block to be statistically "similar," in other words, to be distributed according to a common probability distribution. The hidden Markov models (HMMs) and the dynamical Bayesian methods, which are discussed in Chapter 17, come under this more general umbrella of problems. Our goal in this example is to demonstrate the use of Gibbs sampling in the context of the change-point detection task (see, e.g., [4]).

Let x_n be a discrete random variable that corresponds to the count of an event, for example, the number of requests for telephone calls within an interval of time, requests for individual documents on a web server, particle emissions in radioactive materials, number of accidents in a working environment, and so on. We adopt the Poisson process to model the distribution of x_n, that is,

$$P(x;\lambda) = \frac{(\lambda\tau)^x}{x!}e^{-\lambda\tau}. \tag{14.52}$$

Poisson processes have been widely used to model the number of events that take place in a time interval, τ. For our example, we have chosen $\tau = 1$. The parameter λ is known as the *intensity* of the process (see, e.g., [32]).

We assume that our observations, $x_n,\ n = 1, 2, \ldots, N$, have been generated by two different Poisson processes, $P(x;\lambda_1)$ and $P(x;\lambda_2)$. Also, the change of the model has taken place suddenly at an unknown time instant, n_0. Our goal is to estimate the posterior,

$$P(n_0|\lambda_1, \lambda_2, \boldsymbol{x}_{1:N}).$$

Moreover, the exact values of λ_1 and λ_2 are not known. The only available information is that the Poisson process intensities, $\lambda_i,\ i = 1, 2$, are distributed according to a (prior) gamma distribution, that is,

$$p(\lambda) = \text{Gamma}(\lambda|a,b) = \frac{1}{\Gamma(a)}b^a\lambda^{a-1}\exp(-b\lambda),$$

for some known positive values a, b. We will finally assume that we have no prior information on when the time of change occurred; thus, the prior is chosen to be the uniform distribution, $P(n_0) = \frac{1}{N}$. Based on the previous assumptions, the corresponding joint distribution is given by,

$$p(n_0, \lambda_1, \lambda_2, \boldsymbol{x}_{1:N}) = p(\boldsymbol{x}_{1:N}|\lambda_1, \lambda_2, n_0)p(\lambda_1)p(\lambda_2)P(n_0)$$

or

$$p(n_0, \lambda_1, \lambda_2, \boldsymbol{x}_{1:N}) = \prod_{n=1}^{n_0} P(x_n|\lambda_1) \prod_{n=n_0+1}^{N} P(x_n|\lambda_2)p(\lambda_1)p(\lambda_2)P(n_0).$$

Taking the logarithm in order to get rid of the products, and integrating out respective variables, the following conditionals needed in Gibbs sampling are obtained (Problem 14.15):

$$p(\lambda_1|n_0, \lambda_2, \boldsymbol{x}_{1:N}) = \text{Gamma}(\lambda_1|a_1, b_1), \tag{14.53}$$

with

$$a_1 = a + \sum_{n=1}^{n_0} x_n, \quad b_1 = b + n_0,$$

$$p(\lambda_2|n_0, \lambda_1, \boldsymbol{x}_{1:N}) = \text{Gamma}(\lambda_2|a_2, b_2), \tag{14.54}$$

$$a_2 = a + \sum_{n=n_0+1}^{N} x_n, \quad b_2 = b + (N - n_0),$$

and

$$\begin{aligned} P(n_0|\lambda_1, \lambda_2, \boldsymbol{x}_{1:N}) = \ln\lambda_1 \sum_{n=1}^{n_0} x_n - n_0\lambda_1 + \ln\lambda_2 \sum_{n=n_0+1}^{N} x_n \\ - (N - n_0)\lambda_2, \quad n_0 = 1, 2, \ldots, N. \end{aligned} \tag{14.55}$$

Note that the first two conditionals are gamma distributed and, as we said at the end of Section 14.4, a number of different approaches are available for generating samples from it. The last distribution is a discrete one, and samples can be drawn as discussed in Algorithm 14.1. We are now ready to apply Gibbs sampling.

Algorithm 14.5 (Gibbs sampling for change-point detection).

```
• Having obtained $\boldsymbol{x}_{1:N} := \{x_1, \ldots, x_N\}$, select $a$ and $b$.
• Initialize $n_0^{(0)}$
• For $i = 1, 2, \ldots,$ Do
    • $\lambda_1^{(i)} \sim \text{Gamma}\big(\lambda|a + \sum_{n=1}^{n_0^{(i-1)}} x_n,\ b + n_0^{(i-1)}\big)$
    • $\lambda_2^{(i)} \sim \text{Gamma}\big(\lambda|a + \sum_{n=n_0^{(i-1)}+1}^{N} x_n,\ b + (N - n_0^{(i-1)})\big)$
    • $n_0^{(i)} \sim P(n_0|\lambda_1^{(i)}, \lambda_2^{(i)}, x_{1:N})$
• End For
```

Figure 14.12 shows the number of deadly accidents per year in the coal mines in England spanning the years 1851-1962. Looking at the graph, it is readily observed that the "front" part of the graph looks different from its "back" end, with a change around 1890-1900. As a matter of fact, in 1890, new health and safety regulations were introduced, following pressure from the coal miners' unions. We will use the model explained before and draw samples according to Algorithm 14.5 in order to determine the point, n_0, where a change in the statistical distributions describing the data occurred [4]. The values of a and b were chosen equal to $a = 2$ and $b = 1$, although the obtained results are not sensitive in their choice. The burn-in phase was 200 samples. Figure 14.13 shows the obtained histogram of the values of n_0 drawn by the algorithm, which clearly indicates a peak at the year 1890. Figure 14.14 shows the plot of the points drawn for λ_1 and λ_2. The plot clearly indicates that the intensity of the Poisson process dropped from $\lambda_1 = 3$ to $\lambda_2 = 1$ after the introduction of the safety regulations.

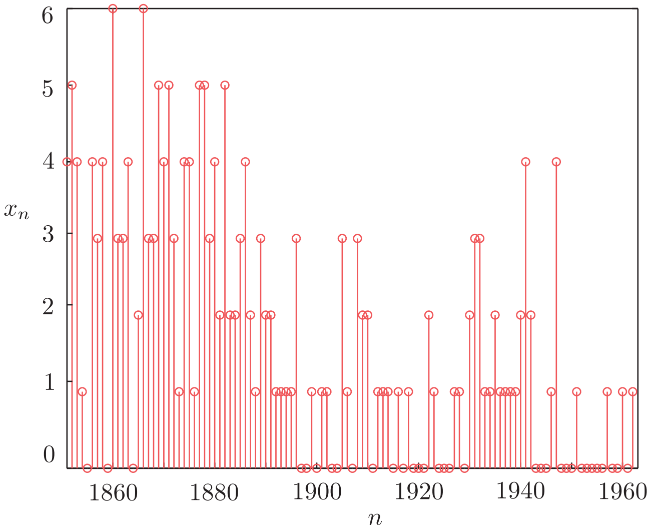


FIGURE 14.12

Number of deadly accidents per year in the coal mines in England over the period 1851-1962.

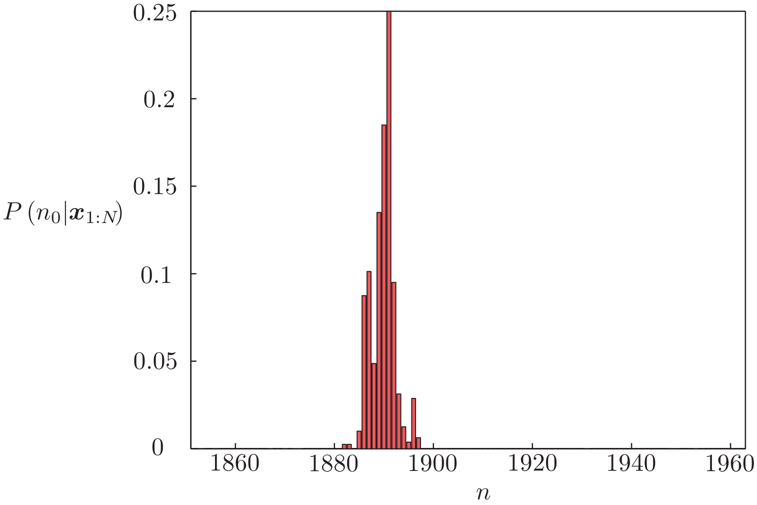


FIGURE 14.13

The histogram obtained from the values of n_0 generated by the algorithm, which approximates the posterior for n_0, for the case study of Section 14.11. Observe that the histogram peaks at 1890, the year when the new regulations were introduced.

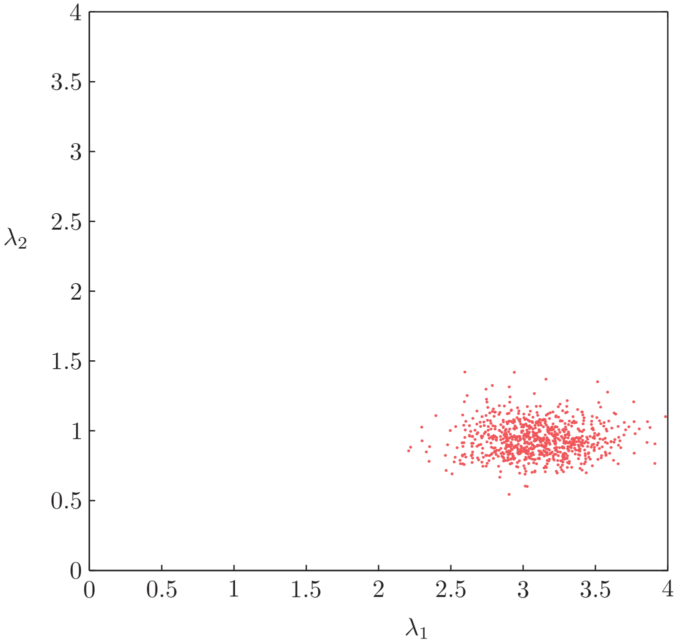


FIGURE 14.14

Case study of Section 14.11: The cluster formed by the obtained values of λ_1, λ_2.

PROBLEMS

14.1 Show that if $F_{\mathrm{x}}(x)$ is the cumulative distribution function of a random variable x, then the random variable $\mathrm{u} = F_{\mathrm{x}}(\mathrm{x})$ follows the uniform distribution in [0, 1].

14.2 Show that if u follows the uniform distribution and

$$\mathrm{x} = F_{\mathrm{x}}^{-1}(\mathrm{u}) := g(\mathrm{u}), \tag{14.56}$$

then indeed x is distributed according to $F_{\mathrm{x}}(x) = \int_{-\infty}^{x} p(x)\,dx$.

14.3 Consider the random variables r and ϕ with exponential and uniform distributions

$$p_{\mathrm{r}}(r) = \frac{1}{2}\exp\left(-\frac{r}{2}\right), \quad r \geq 0,$$

and

$$p_{\phi}(\phi) = \begin{cases} \frac{1}{2\pi} & 0 \leq \phi \leq 2\pi, \\ 0 & \text{otherwise}, \end{cases}$$

respectively. Show that the transformation

$$\mathrm{x} = \sqrt{\mathrm{r}}\cos\phi = g_{\mathrm{x}}(\mathrm{r}, \phi),$$

$$\mathrm{y} = \sqrt{\mathrm{r}}\sin\phi = g_{\mathrm{y}}(\mathrm{r}, \phi),$$

renders both x and y to follow the normalized Gaussian $\mathcal{N}(0,1)$.

14.4 Show that if

$$p_{\mathrm{x}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|0, I),$$

then **y** given by the transformation

$$\mathbf{y} = L\mathbf{x} + \boldsymbol{\mu}$$

is distributed according to

$$p_{\mathrm{y}}(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}, \Sigma),$$

where $\Sigma = LL^T$.

14.5 Consider two Gaussians

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|0, \sigma_p^2 I), \quad \sigma_p^2 = 0.1,$$

and

$$q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|0, \sigma_q^2 I), \quad \sigma_q^2 = 0.11,$$

$\boldsymbol{x} \in \mathbb{R}^l$. In order to use $q(\boldsymbol{x})$ for drawing samples from $p(\boldsymbol{x})$ via the rejection sampling method, a constant c has to be computed so that

$$cq(\boldsymbol{x}) \geq p(\boldsymbol{x}).$$

Show that

$$c \geq \left(\frac{\sigma_q}{\sigma_p}\right)^l,$$

and compute the probability of accepting samples.

14.6 Show that using importance sampling leads to an unbiased estimator for the normalizing constant of the desired distribution,

$$p(\boldsymbol{x}) = \frac{1}{Z}\phi(\boldsymbol{x}).$$

However, the estimator of $\mathbb{E}[f(\mathbf{x})]$ for a function $f(\cdot)$ is a biased one.

14.7 Let $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\mathbf{0}, \sigma_1^2 I)$. Choose the proposal distribution for importance sampling as

$$q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\mathbf{0}, \sigma_2^2 I).$$

The weights are computed as

$$w(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}.$$

If $w(\mathbf{0})$ is the weight at $\boldsymbol{x} = \mathbf{0}$, then the ratio $\frac{w(\boldsymbol{x})}{w(\mathbf{0})}$ is given by

$$\frac{w(\boldsymbol{x})}{w(\mathbf{0})} = \exp\frac{1}{2}\left(\frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^2\sigma_2^2}\sum_{i=1}^{l} x_i^2\right).$$

Observe that even for a very good match between $q(\boldsymbol{x})$ and $p(\boldsymbol{x})$ ($\sigma_1^2 \simeq \sigma_2^2$), for large values of l, the values of the weights can change significantly due to the exponential dependence.

14.8 Show that a stochastic matrix P always has the value $\lambda = 1$ as its eigenvalue.

14.9 Show that if the eigenvalue of a transition matrix is not equal to one, its magnitude cannot be larger than one, that is, $|\lambda| \leq 1$.

14.10 Prove that if P is a stochastic matrix and $\lambda \neq 1$, then the elements of the corresponding eigenvector add to zero.

14.11 Prove the square root dependence of the distance traveled by a random walk, with infinite many integer states, on the time, n.

14.12 Prove, using the detailed balance condition, that the invariant distribution associated with the Markov chain implied by the Metropolis-Hastings algorithm is the desired distribution, $p(\boldsymbol{x})$.

14.13 Show that in Gibbs sampling, the desired joint distribution is invariant with respect to each one of the base transition pdfs.

14.14 Show that the acceptance rate for the Gibbs sampling is equal to one.

14.15 Derive the formulae for the conditional distributions of Section 14.11.

MATLAB Exercise

14.16 Develop a MATLAB code for the Gibbs sampler. Then, use it to draw samples from the two-dimensional Gaussian distribution, with mean value and covariance matrix equal to

$$\boldsymbol{\mu} = [0, 0]^T, \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}.$$

Derive the conditional pdf of each one of the variables with respect to the other (use Appendix in Section 12.9). Then use the conditional pdfs to implement the Gibbs sampler. Plot the generated points in the two-dimensional space after 20, 50, 100, 300, and 1000 iterations. What do you observe concerning convergence?

REFERENCES

[1] D. Barry, J.A. Hartigan, A Bayesian analysis for change point problems, J. Am. Stat. Assoc., 88 (1993) 309-319.

[2] N. Bhatnagar, A. Bogdanov, E. Mossel, The Computational Complexity of Estimating Convergence Time, 2010, arXiv:1007.0089v1 [cs.DS].

[3] G. Celeux, J. Diebolt, The SEM algorithm: a probabilistic teacher derive from the EM algorithm for the mixture problem, Comput. Stat. Q. 2 (1985) 73-82.

[4] A.T. Cemgil, A tutorial introduction to Monte Carlo methods, Markov chain Monte Carlo and particle filtering, in: R. Chellappa, S. Theodoridis (Eds.), Academic Press Library in Signal Processing, Vol. 1, Academic Press, San Diego, CA, (2014) 1065-1113.

[5] M.H. Chen, Q.M. Shao, J.G. Ibrahim (Eds.), Monte Carlo Methods in Bayesian Computation, Springer, New York, 2001.

[6] M.K. Cowles, B.P. Carlin, Markov chain Monte Carlo convergence diagnostics: a comparative review, J. Am. Stat. Assoc. 91 (1996) 883-904.

[7] L. Devroye, Non-Uniform Random Variate Generation, Springer-Verlag, New York, 1986.

[8] A. Gelman, D.B. Rubin, Inference from iterative simulation using multiple sequences, Stat. Sci. 7 (1992) 457-511.

[9] A.E. Gelfand, A.F.M. Smith, Sampling based approaches to calculating marginal densities, J. Am. Stat. Assoc. 85 (1990) 398-409.

[10] S. Geman, D. Geman, Stochastic relaxation Gibbs distributions and the Bayesian restoration of images, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 721-741.

[11] W.R. Gilks, P. Wild, Adaptive rejection sampling for Gibbs sampling, Appl. Stat. 41 (1992) 337-348.

[12] W.R. Gilks, G.O. Roberts, E.I. George, Adaptive direction sampling, Statistician, 43 (1994) 179-189.

[13] P.J. Green, Reversible jump Markof chain Monte Carlo computation and Bayesian model determination, Biometrika 82 (1995) 711-732.

[14] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1970) 97-109.

[15] D.M. Higdon, Auxiliary variable methods for Markov chain Monte Carlo with Applications, J. Am. Stat. Assoc. 93 (1994) 179-189.

[16] C.A. Jensen, A. Kong, U. Kjaeruff, Blocking Gibbs sampling in very large probabilistic expert systems, Int. J. Hum. Comput. Stud. 42 (1995) 647-666.

[17] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671-680.

[18] D.E. Knuth, The Art of Computer Programming, second ed., Addison Wesley, Reading, MA, 1981.

[19] T.L. Lai, Sequential change point detection in quality control and dynamical systems, J. R. Stat. Soc. B, 57 (1995) 613-658.

[20] D.H. Lehmer, Mathematical methods in large scale computing units, Ann. Comput. Lab. Harvard Univ. 26 (1951).

[21] D.A. Levin, Y. Peres, E.L. Wilmer, Markov Chains and Mixing Times, American Mathematical Society, Providence, RI, 2008.

[22] F. Liang, C. Liu, R.J. Caroll, Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples, John Wiley, New York, 2010.

[23] J.S. Liu, The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem, J. Am. Stat. Assoc. 89 (427) (1994) 958-966.

[24] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer, New York, 2001.

[25] E. Marinari, G. Parisi, Simulated tempering: a new Monte Carlo scheme, Europhys. Lett. 19 (6) (1992) 451-458.

[26] D.J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, 2003.

[27] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (1953) 1087-1091.

[28] N. Metropolis, The beginning of Monte Carlo methods, Los Alamos Sci. (1987) 125-130.

[29] J. Moller, A.N. Pettitt, R. Reeves, K.K. Berthelsen, An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants, Biometrica 93 (2006) 451-668.

[30] R.M. Neal, Probabilistic inference using Markov chain Monte Carlo methods, Technical Report (GR-TR-93-1), Department of Computer Science, University of Toronto, Canada, 1993.

[31] R.M. Neal, Slice sampling, Ann. Stat. 31 (2003) 705-767.

[32] A. Papoulis, S.U. Pillai, Probability, Random Variables and Stochastic Processes, fourth ed., McGraw-Hill, New York, 2002.

[33] P.H. Peskun, Optimum Monte Carlo sampling using Markov chains, Biometrika, 60 (1973) 607-612.

[34] S.K. Park, K.W. Miller, Random number generations: Good ones are hard to find, Commun. ACM 31 (10) (1988) 1192-1201.

[35] M. Plummer, N. Best, K. Cowles, CODA: output analysis and diagnostics for Markov chain Monte Carlo simulations, 2006, http://cran.r-project.org.

[36] J. Reeves, J. Chen, X.L. Wang, R. Lund, Q.Q. Lu, A review and comparison of changepoint detection techniques for climate data, J. Appl. Meteorol. Climatol. 46 (2007) 900-915.

[37] B. Ripley, Stochastic Simulation, John Wiley, New York, 1987.

[38] C.P. Robert, G. Casella, Monte Carlo Statistical Methods, second ed., Springer, New York, 2004.

[39] A. Sinclair, Algorithms for Random Generation and Counting: A Markov chain Approach, Birkhüuser, Boston, 1993.

[40] L. Tierney, Markov chains for exploring posterior distribution, Ann. Stat. 22 (1994) 1701-1762.