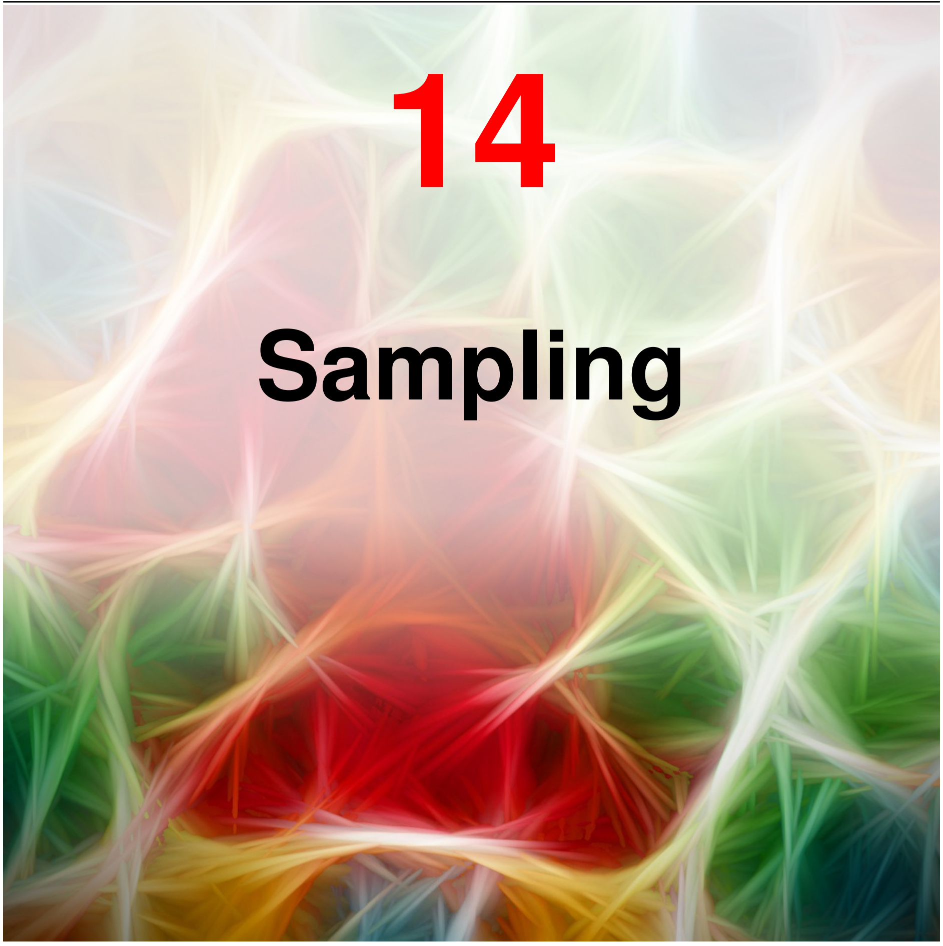


# 14 Sampling

There are many situations in deep learning where we need to create synthetic examples of a variable $\mathbf{z}$ from a probability distribution $p(\mathbf{z})$. Here $\mathbf{z}$ might be a scalar and the distribution might be a univariate Gaussian, or $\mathbf{z}$ might be a high-resolution image and $p(\mathbf{z})$ might be a generative model defined by a deep neural network. The process of creating such examples is known as *sampling*, also known as *Monte Carlo sampling*. For many simple distributions there are numerical techniques that generate suitable samples directly, whereas for more complex distributions, including ones that are defined implicitly, we may need more sophisticated approaches. We adopt the convention of referring to each instantiated value as a sample, in contrast to the convention used in classical statistics whereby 'sample' refers to a set of values.

In this chapter we focus on aspects of sampling that are most relevant to deep learning. Further information on Monte Carlo methods more generally can be found in Gilks, Richardson, and Spiegelhalter (1996) and Robert and Casella (1999).

## 14.1. Basic Sampling Algorithms

In this section, we explore a variety of relatively simple strategies for generating random samples from a given distribution. Because the samples will be generated by a computer algorithm, they will in fact be *pseudo-random*, that is, they will be calculated using a deterministic algorithm but must nevertheless pass appropriate tests for randomness. Here we will assume that an algorithm has been provided that generates pseudo-random numbers distributed uniformly over $(0, 1)$, and indeed most software environments have such a facility built in.

### 14.1.1 Expectations

Although for some applications the samples themselves may be of direct interest, in other situations the goal is to evaluate *expectations* with respect to the distribution. Suppose we wish to find the expectation of a function $f(\mathbf{z})$ with respect to a probability distribution $p(\mathbf{z})$. Here, the components of $\mathbf{z}$ might comprise discrete or continuous variables or some combination of the two. For continuous variables the expectation is defined by

$$\mathbb{E}[f] = \int f(\mathbf{z})p(\mathbf{z})\,\mathrm{d}\mathbf{z} \tag{14.1}$$

where the integral is replaced by summation for discrete variables. This is illustrated schematically for a single continuous variable in Figure 14.1. We will suppose that such expectations are too complex to be evaluated exactly using analytical techniques.

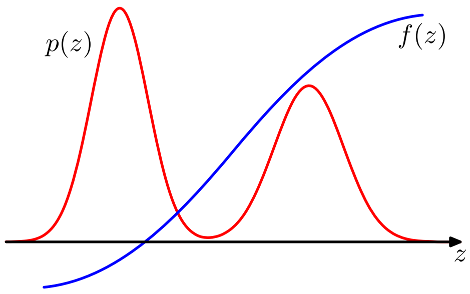
The general idea behind sampling methods is to obtain a set of samples $\mathbf{z}^{(l)}$ (where $l = 1, \ldots, L$) drawn independently from the distribution $p(\mathbf{z})$. This allows the expectation (14.1) to be approximated by a finite sum:

$$\overline{f} = \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{z}^{(l)}). \tag{14.2}$$

If the samples $\mathbf{z}^{(l)}$ are drawn from the distribution $p(\mathbf{z})$, then $\mathbb{E}[\overline{f}] = \mathbb{E}[f(\mathbf{z})]$ and so the estimator $\overline{f}$ has the correct mean. We can also write this in the form

*Exercise 14.1*

Figure 14.1 Schematic illustration of a function $f(z)$ whose expectation is to be evaluated with respect to a distribution $p(z)$.



$$\mathbb{E}[f(\mathbf{z})] \simeq \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{z}^{(l)}) \tag{14.3}$$

where the symbol $\simeq$ denotes that the right-hand side is an unbiased estimator of the left-hand side, that is the two sides are equal when averaged over the noise distribution.

*Exercise 14.2*

The variance of the estimator (14.2) is given by

$$\mathrm{var}[\overline{f}] = \frac{1}{L}\mathbb{E}\left[(f - \mathbb{E}[f])^2\right], \tag{14.4}$$

which is the variance of the function $f(\mathbf{z})$ under the distribution $p(\mathbf{z})$. Note that the linear decrease of this variance with increasing $L$ does not depend on the dimensionality of $\mathbf{z}$, and that, in principle, high accuracy may be achievable with a relatively small number of samples $\{\mathbf{z}^{(l)}\}$. The problem, however, is that the samples $\{\mathbf{z}^{(l)}\}$ might not be independent, and so the effective sample size might be much smaller than the apparent sample size. Also, referring back to Figure 14.1, note that if $f(\mathbf{z})$ is small in regions where $p(\mathbf{z})$ is large and vice versa, then the expectation may be dominated by regions of small probability, implying that relatively large sample sizes will be required to achieve sufficient accuracy.

### 14.1.2 Standard distributions

We now consider how to generate random numbers from simple nonuniform distributions, assuming that we already have available a source of uniformly distributed random numbers. Suppose that $z$ is uniformly distributed over the interval $(0, 1)$, and that we transform the values of $z$ using some function $g(\cdot)$ so that $y = g(z)$. The distribution of $y$ will be governed by

*Section 2.4*

$$p(y) = p(z)\left|\frac{\mathrm{d}z}{\mathrm{d}y}\right| \tag{14.5}$$

where, in this case, $p(z) = 1$. Our goal is to choose the function $g(z)$ such that the resulting values of $y$ have some specific desired distribution $p(y)$. Integrating (14.5) we obtain

$$z = \int_{-\infty}^{y} p(\widehat{y}) \equiv h(y)\,\mathrm{d}\widehat{y} \tag{14.6}$$

*Exercise 14.3*

which is the indefinite integral of $p(y)$. Thus, $y = h^{-1}(z)$, and so we have to transform the uniformly distributed random numbers using a function that is the inverse of the indefinite integral of the desired distribution. This is illustrated in Figure 14.2.

Consider for example the *exponential distribution*

$$p(y) = \lambda \exp(-\lambda y) \tag{14.7}$$

where $0 \leqslant y < \infty$. In this case the lower limit of the integral in (14.6) is 0, and so $h(y) = 1 - \exp(-\lambda y)$. Thus, if we transform our uniformly distributed variable $z$ using $y = -\lambda^{-1}\ln(1 - z)$, then $y$ will have an exponential distribution.

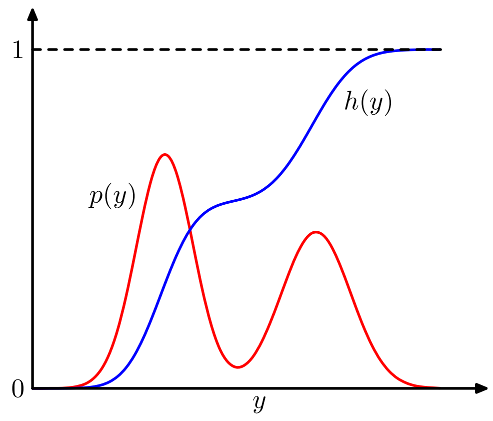


**Figure 14.2** Geometrical interpretation of the transformation method for generating non-uniformly distributed random numbers. $h(y)$ is the indefinite integral of the desired distribution $p(y)$. If a uniformly distributed random variable $z$ is transformed using $y = h^{-1}(z)$, then $y$ will be distributed according to $p(y)$.

Another example of a distribution to which the transformation method can be applied is given by the Cauchy distribution

$$p(y) = \frac{1}{\pi}\frac{1}{1+y^2}. \tag{14.8}$$

*Exercise 14.4*

In this case, the inverse of the indefinite integral can be expressed in terms of the $\tan$ function.

*Section 2.4*

The generalization to multiple variables involves the Jacobian of the change of variables, so that

$$p(y_1,\ldots,y_M) = p(z_1,\ldots,z_M)\left|\frac{\partial(z_1,\ldots,z_M)}{\partial(y_1,\ldots,y_M)}\right|. \tag{14.9}$$

As a final example of the transformation technique, we consider the Box–Muller method for generating samples from a Gaussian distribution. First, suppose we generate pairs of uniformly distributed random numbers $z_1, z_2 \in (-1,1)$, which we can do by transforming a variable distributed uniformly over $(0,1)$ using $z \to 2z-1$. Next we discard each pair unless it satisfies $z_1^2 + z_2^2 \leqslant 1$. This leads to a uniform distribution of points inside the unit circle with $p(z_1,z_2) = 1/\pi$, as illustrated in Figure 14.3. Then, for each pair $z_1, z_2$ we evaluate the quantities

$$y_1 = z_1\left(\frac{-2\ln r^2}{r^2}\right)^{1/2} \tag{14.10}$$

$$y_2 = z_2\left(\frac{-2\ln r^2}{r^2}\right)^{1/2} \tag{14.11}$$

*Exercise 14.5*

where $r^2 = z_1^2 + z_2^2$. Then the joint distribution of $y_1$ and $y_2$ is given by

$$\begin{aligned} p(y_1,y_2) &= p(z_1,z_2)\left|\frac{\partial(z_1,z_2)}{\partial(y_1,y_2)}\right| \\ &= \left[\frac{1}{\sqrt{2\pi}}\exp(-y_1^2/2)\right]\left[\frac{1}{\sqrt{2\pi}}\exp(-y_2^2/2)\right] \end{aligned} \tag{14.12}$$

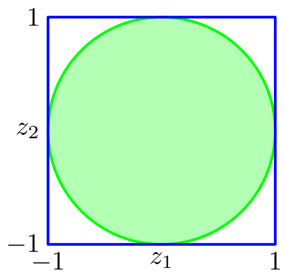


**Figure 14.3** The Box–Muller method for generating Gaussian-distributed random numbers starts by generating samples from a uniform distribution inside the unit circle.

and so $y_1$ and $y_2$ are independent and each has a Gaussian distribution with zero mean and unit variance.

If $y$ has a Gaussian distribution with zero mean and unit variance, then $\sigma y + \mu$ will have a Gaussian distribution with mean $\mu$ and variance $\sigma^2$. To generate vector-valued variables having a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, we can make use of the *Cholesky decomposition*, which takes the form $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$ (Deisenroth, Faisal, and Ong, 2020). Then, if $\mathbf{z}$ is a random vector whose components are independent and Gaussian distributed with zero mean and unit variance, then $\mathbf{y} = \boldsymbol{\mu} + \mathbf{L}\mathbf{z}$ will be Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. *Exercise 14.6*

Clearly, the transformation technique depends for its success on the ability to calculate and then invert the indefinite integral of the required distribution. Such operations are feasible only for a limited number of simple distributions, and so we must turn to alternative approaches in search of a more general strategy. Here we consider two techniques called *rejection sampling* and *importance sampling*. Although mainly limited to univariate distributions and thus not directly applicable to complex problems in many dimensions, they do form important components in more general strategies.

## 14.1.3 Rejection sampling

The rejection sampling framework allows us to sample from relatively complex distributions, subject to certain constraints. We begin by considering univariate distributions and subsequently discuss the extension to multiple dimensions.

Suppose we wish to sample from a distribution $p(\mathbf{z})$ that is not one of the simple, standard distributions considered so far and that sampling directly from $p(\mathbf{z})$ is difficult. Furthermore suppose, as is often the case, that we are easily able to evaluate $p(\mathbf{z})$ for any given value of $\mathbf{z}$, up to some normalizing constant $Z$, so that

$$p(z) = \frac{1}{Z_p}\widetilde{p}(z) \tag{14.13}$$

where $\widetilde{p}(z)$ can readily be evaluated, but $Z_p$ is unknown.

To apply rejection sampling, we need some simpler distribution $q(z)$, sometimes called a *proposal distribution*, from which we can readily draw samples. We next introduce a constant $k$ whose value is chosen such that $kq(z) \geqslant \widetilde{p}(z)$ for all values of $z$. The function $kq(z)$ is called the comparison function and is illustrated

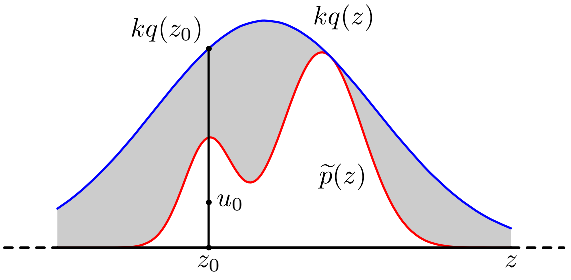


**Figure 14.4** In the rejection sampling method, samples are drawn from a simple distribution $q(z)$ and rejected if they fall in the grey area between the unnormalized distribution $\widetilde{p}(z)$ and the scaled distribution $kq(z)$. The resulting samples are distributed according to $p(z)$, which is the normalized version of $\widetilde{p}(z)$.

for a univariate distribution in Figure 14.4. Each step of the rejection sampler involves generating two random numbers. First, we generate a number $z_0$ from the distribution $q(z)$. Next, we generate a number $u_0$ from the uniform distribution over $[0, kq(z_0)]$. This pair of random numbers has uniform distribution under the curve of the function $kq(z)$. Finally, if $u_0 > \widetilde{p}(z_0)$ then the sample is rejected, otherwise $u_0$ is retained. Thus, the pair is rejected if it lies in the grey shaded region in Figure 14.4. The remaining pairs then have uniform distribution under the curve of $\widetilde{p}(z)$, and hence the corresponding $z$ values are distributed according to $p(z)$, as desired.

*Exercise 14.7*

The original values of $z$ are generated from the distribution $q(z)$, and these samples are then accepted with probability $\widetilde{p}(z)/kq(z)$, and so the probability that a sample will be accepted is given by

$$\begin{aligned} p(\text{accept}) &= \int \{\widetilde{p}(z)/kq(z)\}\, q(z)\, \mathrm{d}z \\ &= \frac{1}{k}\int \widetilde{p}(z)\, \mathrm{d}z. \end{aligned} \tag{14.14}$$

Thus, the fraction of points that are rejected by this method depends on the ratio of the area under the unnormalized distribution $\widetilde{p}(z)$ to the area under the curve $kq(z)$. We therefore see that the constant $k$ should be as small as possible subject to the limitation that $kq(z)$ must be nowhere less than $\widetilde{p}(z)$.

As an illustration of the use of rejection sampling, consider the task of sampling from the gamma distribution

$$\mathrm{Gam}(z|a,b) = \frac{b^a z^{a-1}\exp(-bz)}{\Gamma(a)}, \tag{14.15}$$

which, for $a > 1$, has a bell-shaped form, as shown in Figure 14.5. A suitable proposal distribution is therefore the Cauchy (14.8) because this too is bell-shaped and because we can use the transformation method, discussed earlier, to sample from it. We need to generalize the Cauchy slightly to ensure that it nowhere has a smaller value than the gamma distribution. This can be achieved by transforming a uniform random variable $y$ using $z = b\tan y + c$, which gives random numbers distributed according to

*Exercise 14.8*

$$q(z) = \frac{k}{1 + (z-c)^2/b^2}. \tag{14.16}$$

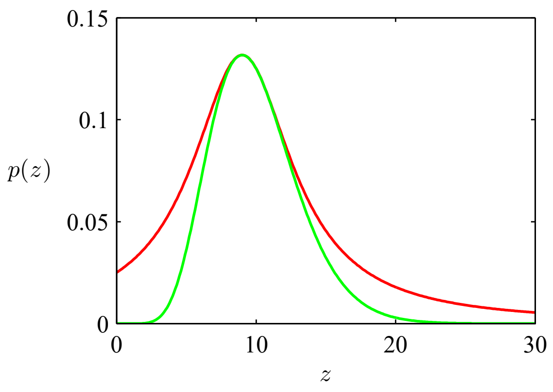


**Figure 14.5** Plot showing the gamma distribution given by (14.15) as the green curve, with a scaled Cauchy proposal distribution shown by the red curve. Samples from the gamma distribution can be obtained by sampling from the Cauchy and then applying the rejection sampling criterion.

The minimum reject rate is obtained by setting $c = a-1$, and $b^2 = 2a-1$ and choosing the constant $k$ to be as small as possible while still satisfying the requirement $kq(z) \geqslant \widetilde{p}(z)$. The resulting comparison function is also illustrated in Figure 14.5.

### 14.1.4 Adaptive rejection sampling

In many instances where we might wish to apply rejection sampling, it can be difficult to determine a suitable analytic form for the envelope distribution $q(z)$. An alternative approach is to construct the envelope function on the fly based on measured values of the distribution $p(z)$ (Gilks and Wild, 1992). Constructing an envelope function is particularly straightforward when $p(z)$ is log concave, in other words when $\ln p(z)$ has derivatives that are non-increasing functions of $z$. The construction of a suitable envelope function is illustrated graphically in Figure 14.6.

The function $\ln p(z)$ and its gradient are evaluated at some initial set of grid points, and the intersections of the resulting tangent lines are used to construct the envelope function. Next a sample value is drawn from the envelope distribution. *Exercise 14.10* This is straightforward because the log of the envelope distribution is a succession of linear functions, and hence the envelope distribution itself comprises a piecewise exponential distribution of the form

$$q(z) = k_i \lambda_i \exp\left\{-\lambda_i (z - z_{i-1})\right\}, \qquad z_{i-1} < z \leqslant z_i. \tag{14.17}$$

Once a sample has been drawn, the usual rejection criterion can be applied. If the sample is accepted, then it will be a draw from the desired distribution. If, however, the sample is rejected, then it is incorporated into the set of grid points, a new tangent line is computed, and the envelope function is thereby refined. As the number of grid points increases, so the envelope function becomes a better approximation of the desired distribution $p(z)$ and the probability of rejection decreases.

There is a variant of the algorithm exists that avoids the evaluation of derivatives (Gilks, 1992). The adaptive rejection sampling framework can also be extended to distributions that are not log concave, simply by following each rejection sampling

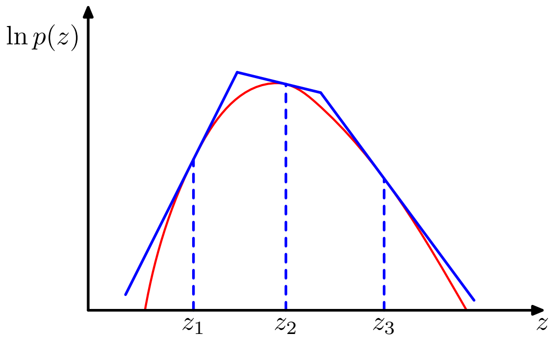


**Figure 14.6** In rejection sampling, if a distribution is log concave then an envelope function can be constructed using the tangent lines computed at a set of grid points. If a sample point is rejected, it is added to the set of grid points and used to refine the envelope distribution.

step with a Metropolis–Hastings step (to be discussed in Section 14.2.3), giving rise to *adaptive rejection Metropolis* sampling (Gilks, Best, and Tan, 1995).

For rejection sampling to be of practical value, we require that the comparison function is close to the required distribution so that the rate of rejection is kept to a minimum. Now let us examine what happens when we try to use rejection sampling in spaces of high dimensionality. Consider, for illustration, a somewhat artificial problem in which we wish to sample from a zero-mean multivariate Gaussian distribution with covariance $\sigma_p^2\mathbf{I}$, where $\mathbf{I}$ is the unit matrix, by rejection sampling from a proposal distribution that is itself a zero-mean Gaussian distribution having covariance $\sigma_q^2\mathbf{I}$. Clearly, we must have $\sigma_q^2 \geqslant \sigma_p^2$ to ensure that there exists a $k$ such that $kq(z) \geqslant p(z)$. In $D$-dimensions, the optimum value of $k$ is given by $k = (\sigma_q/\sigma_p)^D$, as illustrated for $D = 1$ in Figure 14.7. The acceptance rate will be the ratio of volumes under $p(z)$ and $kq(z)$, which, because both distributions are normalized, is just $1/k$. Thus, the acceptance rate diminishes exponentially with dimensionality. Even if $\sigma_q$ exceeds $\sigma_p$ by just 1%, for $D = 1{,}000$ the acceptance ratio will be approximately $1/20{,}000$. In this illustrative example, the comparison function is close to the required distribution. For more practical examples, where the desired distribution may be multimodal and sharply peaked, it will be extremely difficult to find

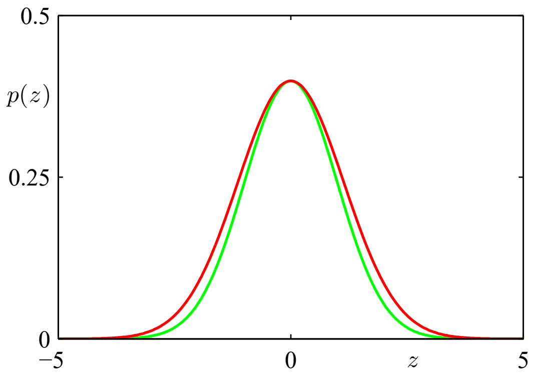


**Figure 14.7** Illustrative example used to highlight a limitation of rejection sampling. Samples are drawn from a Gaussian distribution $p(z)$ shown by the green curve, by using rejection sampling from a proposal distribution $q(z)$ that is also Gaussian and whose scaled version $kq(z)$ is shown by the red curve.

a good proposal distribution and comparison function. Furthermore, the exponential decrease of the acceptance rate with dimensionality is a generic feature of rejection sampling. Although rejection can be a useful technique in one or two dimensions, it is unsuited to problems of high dimensionality. It can, however, play a role as a subroutine in more sophisticated algorithms for sampling in high-dimensional spaces.

### 14.1.5 Importance sampling

One reason for wishing to sample from complicated probability distributions is to evaluate expectations of the form (14.1). The technique of *importance sampling* provides a framework for approximating expectations directly but does not itself provide a mechanism for drawing samples from a distribution $p(\mathbf{z})$.

The finite sum approximation to the expectation, given by (14.2), depends on being able to draw samples from the distribution $p(\mathbf{z})$. Suppose, however, that it is impractical to sample directly from $p(\mathbf{z})$ but that we can evaluate $p(\mathbf{z})$ easily for any given value of $\mathbf{z}$. One simplistic strategy for evaluating expectations would be to discretize $\mathbf{z}$-space into a uniform grid and to evaluate the integrand as a sum of the form

$$\mathbb{E}[f] \simeq \sum_{l=1}^{L} p(\mathbf{z}^{(l)}) f(\mathbf{z}^{(l)}). \tag{14.18}$$

An obvious problem with this approach is that the number of terms in the summation grows exponentially with the dimensionality of $\mathbf{z}$. Furthermore, as we have already noted, the kinds of probability distributions of interest will often have much of their mass confined to relatively small regions of $\mathbf{z}$-space and so uniform sampling will be very inefficient because in high-dimensional problems, only a very small proportion of the samples will make a significant contribution to the sum. We would really like to choose sample points from regions where $p(\mathbf{z})$ is large or ideally where the product $p(\mathbf{z})f(\mathbf{z})$ is large.

As with rejection sampling, importance sampling is based a proposal distribution $q(\mathbf{z})$ from which it is easy to draw samples, as illustrated in Figure 14.8. We can then express the expectation in the form of a finite sum over samples $\{\mathbf{z}^{(l)}\}$ drawn from $q(\mathbf{z})$:

$$\begin{aligned}\mathbb{E}[f] &= \int f(\mathbf{z})p(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &= \int f(\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}f(\mathbf{z}^{(l)}).\end{aligned} \tag{14.19}$$

The quantities $r_l = p(\mathbf{z}^{(l)})/q(\mathbf{z}^{(l)})$ are known as *importance weights*, and they correct the bias introduced by sampling from the wrong distribution. Note that, unlike rejection sampling, all the samples generated are retained.

Often the distribution $p(\mathbf{z})$ can be evaluated only up to a normalization constant, so that $p(\mathbf{z}) = \widetilde{p}(\mathbf{z})/Z_p$ where $\widetilde{p}(\mathbf{z})$ can be evaluated easily, whereas $Z_p$ is unknown.

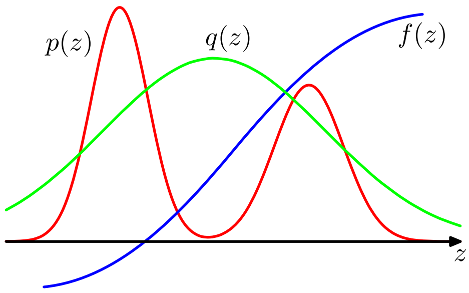


**Figure 14.8** Importance sampling addresses the problem of evaluating the expectation of a function $f(z)$ with respect to a distribution $p(z)$ from which it is difficult to draw samples directly. Instead, samples $\{z^{(l)}\}$ are drawn from a simpler distribution $q(z)$, and the corresponding terms in the summation are weighted by the ratios $p(z^{(l)})/q(z^{(l)})$.

Similarly, we may wish to use an importance sampling distribution $q(\mathbf{z}) = \widetilde{q}(\mathbf{z})/Z_q$, which has the same property. We then have

$$\begin{aligned} \mathbb{E}[f] &= \int f(\mathbf{z})p(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &= \frac{Z_q}{Z_p}\int f(\mathbf{z})\frac{\widetilde{p}(\mathbf{z})}{\widetilde{q}(\mathbf{z})}q(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &\simeq \frac{Z_q}{Z_p}\frac{1}{L}\sum_{l=1}^{L}\widetilde{r}_l f(\mathbf{z}^{(l)}) \end{aligned} \tag{14.20}$$

where $\widetilde{r}_l = \widetilde{p}(\mathbf{z}^{(l)})/\widetilde{q}(\mathbf{z}^{(l)})$. We can use the same sample set to evaluate the ratio $Z_p/Z_q$ with the result

$$\begin{aligned} \frac{Z_p}{Z_q} &= \frac{1}{Z_q}\int \widetilde{p}(\mathbf{z})\,\mathrm{d}\mathbf{z} = \int \frac{\widetilde{p}(\mathbf{z})}{\widetilde{q}(\mathbf{z})}q(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\widetilde{r}_l \end{aligned} \tag{14.21}$$

and hence the expectation in (14.20) is given by a weighted sum:

$$\mathbb{E}[f] \simeq \sum_{l=1}^{L} w_l f(\mathbf{z}^{(l)}) \tag{14.22}$$

where we have defined

$$w_l = \frac{\widetilde{r}_l}{\sum_m \widetilde{r}_m} = \frac{\widetilde{p}(\mathbf{z}^{(l)})/q(\mathbf{z}^{(l)})}{\sum_m \widetilde{p}(\mathbf{z}^{(m)})/q(\mathbf{z}^{(m)})}. \tag{14.23}$$

Note that $\{w_l\}$ are non-negative numbers that sum to one.

As with rejection sampling, the success of importance sampling depends crucially on how well the sampling distribution $q(\mathbf{z})$ matches the desired distribution $p(\mathbf{z})$. If, as is often the case, $p(\mathbf{z})f(\mathbf{z})$ is strongly varying and has a significant proportion of its mass concentrated over relatively small regions of $\mathbf{z}$-space, then the set of importance weights $\{r_l\}$ may be dominated by a few weights having large values, with the remaining weights being relatively insignificant. Thus, the effective sample size can be much smaller than the apparent sample size $L$. The problem is even more severe if none of the samples falls in the regions where $p(\mathbf{z})f(\mathbf{z})$ is large. In that case, the apparent variances of $r_l$ and $r_l f(\mathbf{z}^{(l)})$ may be small even though the estimate of the expectation may be severely wrong. Hence, a major drawback of importance sampling is its potential to produce results that are arbitrarily in error and with no diagnostic indication. This also highlights a key requirement for the sampling distribution $q(\mathbf{z})$, namely that it should not be small or zero in regions where $p(\mathbf{z})$ may be significant.

### 14.1.6 Sampling-importance-resampling

The rejection sampling method discussed in Section 14.1.3 depends in part for its success on the determination of a suitable value for the constant $k$. For many pairs of distributions $p(\mathbf{z})$ and $q(\mathbf{z})$, it will be impractical to determine a suitable value for $k$ as any value that is sufficiently large to guarantee a bound on the desired distribution will lead to impractically small acceptance rates.

As with rejection sampling, the *sampling-importance-resampling* approach also makes use of a sampling distribution $q(\mathbf{z})$ but avoids having to determine the constant $k$. There are two stages to the scheme. In the first stage, $L$ samples $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(L)}$ are drawn from $q(\mathbf{z})$. Then in the second stage, weights $w_1, \ldots, w_L$ are constructed using (14.23). Finally, a second set of $L$ samples is drawn from the discrete distribution $(\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(L)})$ with probabilities given by the weights $(w_1, \ldots, w_L)$.

The resulting $L$ samples are only approximately distributed according to $p(\mathbf{z})$, but the distribution becomes correct in the limit $L \to \infty$. To see this, consider the univariate case, and note that the cumulative distribution of the resampled values is given by

$$\begin{aligned} p(z \leqslant a) &= \sum_{l:z^{(l)} \leqslant a} w_l \\ &= \frac{\sum_l I(z^{(l)} \leqslant a)\widetilde{p}(z^{(l)})/q(z^{(l)})}{\sum_l \widetilde{p}(z^{(l)})/q(z^{(l)})} \end{aligned} \tag{14.24}$$

where $I(.)$ is the indicator function (which equals 1 if its argument is true and 0 otherwise). Taking the limit $L \to \infty$ and assuming suitable regularity of the distributions, we can replace the sums by integrals weighted according to the original sampling distribution $q(z)$:

$$\begin{aligned} p(z \leqslant a) &= \frac{\displaystyle\int I(z \leqslant a)\,\{\widetilde{p}(z)/q(z)\}\,q(z)\,\mathrm{d}z}{\displaystyle\int \{\widetilde{p}(z)/q(z)\}\,q(z)\,\mathrm{d}z} \\ &= \frac{\displaystyle\int I(z \leqslant a)\widetilde{p}(z)\,\mathrm{d}z}{\displaystyle\int \widetilde{p}(z)\,\mathrm{d}z} \\ &= \int I(z \leqslant a) p(z)\,\mathrm{d}z, \end{aligned} \tag{14.25}$$

which is the cumulative distribution function of $p(z)$. Again, we see that normalization of $p(z)$ is not required.

For a finite value of $L$ and a given initial sample set, the resampled values will only approximately be drawn from the desired distribution. As with rejection sampling, the approximation improves as the sampling distribution $q(\mathbf{z})$ gets closer to the desired distribution $p(\mathbf{z})$. When $q(\mathbf{z}) = p(\mathbf{z})$, the initial samples $(\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(L)})$ have the desired distribution and the weights $w_n = 1/L$, so that the resampled values also have the desired distribution.

If moments with respect to the distribution $p(\mathbf{z})$ are required, then they can be evaluated directly using the original samples together with the weights, because

$$\begin{aligned} \mathbb{E}[f(\mathbf{z})] &= \int f(\mathbf{z}) p(\mathbf{z})\,\mathrm{d}\mathbf{z} \\ &= \frac{\displaystyle\int f(\mathbf{z})[\widetilde{p}(\mathbf{z})/q(\mathbf{z})]q(\mathbf{z})\,\mathrm{d}\mathbf{z}}{\displaystyle\int [\widetilde{p}(\mathbf{z})/q(\mathbf{z})]q(\mathbf{z})\,\mathrm{d}\mathbf{z}} \\ &\simeq \sum_{l=1}^{L} w_l f(\mathbf{z}_l). \end{aligned} \tag{14.26}$$

## 14.2. Markov Chain Monte Carlo

In the previous section, we discussed the rejection sampling and importance sampling strategies for evaluating expectations of functions, and we saw that they suffer from severe limitations particularly in spaces of high dimensionality. We therefore turn in this section to a very general and powerful framework called Markov chain Monte Carlo, which allows sampling from a large class of distributions and which scales well with the dimensionality of the sample space. Markov chain Monte Carlo methods have their origins in physics (Metropolis and Ulam, 1949), and it was only towards the end of the 1980s that they started to have a significant impact in the field of statistics.

As with rejection and importance sampling, we again sample from a proposal distribution. This time, however, we maintain a record of the current state $\mathbf{z}^{(\tau)}$, and the proposal distribution $q(\mathbf{z}|\mathbf{z}^{(\tau)})$ is conditioned on this current state, and so the sequence of samples $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots$ forms a Markov chain. Again, if we write $p(\mathbf{z}) = \widetilde{p}(\mathbf{z})/Z_p$, we will assume that $\widetilde{p}(\mathbf{z})$ can readily be evaluated for any given value of $\mathbf{z}$, although the value of $Z_p$ may be unknown. The proposal distribution is chosen to be sufficiently simple that it is straightforward to draw samples from it directly. At each cycle of the algorithm, we generate a candidate sample $\mathbf{z}^{\star}$ from the proposal distribution and then accept the sample according to an appropriate criterion.

*Section 14.2.2*

### 14.2.1 The Metropolis algorithm

In the basic *Metropolis* algorithm (Metropolis *et al.*, 1953), we assume that the proposal distribution is symmetric, that is $q(\mathbf{z}_A|\mathbf{z}_B) = q(\mathbf{z}_B|\mathbf{z}_A)$ for all values of $\mathbf{z}_A$ and $\mathbf{z}_B$. The candidate sample is then accepted with probability

$$A(\mathbf{z}^{\star}, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\widetilde{p}(\mathbf{z}^{\star})}{\widetilde{p}(\mathbf{z}^{(\tau)})}\right). \tag{14.27}$$

This can be achieved by choosing a random number $u$ with uniform distribution over the unit interval $(0, 1)$ and then accepting the sample if $A(\mathbf{z}^{\star}, \mathbf{z}^{(\tau)}) > u$. Note that if the step from $\mathbf{z}^{(\tau)}$ to $\mathbf{z}^{\star}$ causes an increase in the value of $p(\mathbf{z})$, then the candidate point is certain to be kept.

If the candidate sample is accepted, then $\mathbf{z}^{(\tau+1)} = \mathbf{z}^{\star}$, otherwise the candidate point $\mathbf{z}^{\star}$ is discarded, $\mathbf{z}^{(\tau+1)}$ is set to $\mathbf{z}^{(\tau)}$, and another candidate sample is drawn from the distribution $q(\mathbf{z}|\mathbf{z}^{(\tau+1)})$. This is in contrast to rejection sampling, where rejected samples are simply discarded. In the Metropolis algorithm, when a candidate point is rejected, the previous sample is included in the final list of samples, leading to multiple copies of samples. Of course, in a practical implementation, only a single copy of each retained sample would be kept, along with an integer weighting factor recording how many times that state appears. As we will see, if $q(\mathbf{z}_A|\mathbf{z}_B)$ is positive for any values of $\mathbf{z}_A$ and $\mathbf{z}_B$ (this is a sufficient but not necessary condition), the distribution of $\mathbf{z}^{(\tau)}$ tends to $p(\mathbf{z})$ as $\tau \to \infty$. It should be emphasized, however, that the sequence $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots$ is not a set of independent samples from $p(\mathbf{z})$ because successive samples are highly correlated. If we wish to obtain independent samples, then we can discard most of the sequence and just retain every $M$th sample. For $M$ sufficiently large, the retained samples will for all practical purposes be independent. The Metropolis algorithm in summarized in Algorithm 14.1. Figure 14.9 shows a simple illustrative example of sampling from a two-dimensional Gaussian distribution using the Metropolis algorithm in which the proposal distribution is an isotropic Gaussian.

Further insight into the nature of Markov chain Monte Carlo algorithms can be gleaned by looking at the properties of a specific example, namely a simple random

**Algorithm 14.1:** Metropolis sampling

```
Input: Unnormalized distribution $\widetilde{p}(\mathbf{z})$
       Proposal distribution $q(\mathbf{z}|\widehat{\mathbf{z}})$
       Initial state $\mathbf{z}^{(0)}$
       Number of iterations $T$
Output: $\mathbf{z} \sim \widetilde{p}(\mathbf{z})$
-----------------------------------------------------------
$\mathbf{z}_{\text{prev}} \leftarrow \mathbf{z}^{(0)}$
// Iterative message-passing
for $\tau \in \{1, \dots, T\}$ do
    $\mathbf{z}^{\star} \sim q(\mathbf{z}|\mathbf{z}_{\text{prev}})$ // Sample from proposal distribution
    $u \sim \mathcal{U}(0, 1)$ // Sample from uniform
    if $\widetilde{p}(\mathbf{z}^{\star}) \,/\, \widetilde{p}(\mathbf{z}_{\text{prev}}) > u$ then
        $\mathbf{z}_{\text{prev}} \leftarrow \mathbf{z}^{\star}$ // $\mathbf{z}^{(\tau)} = \mathbf{z}^{\star}$
    else
        $\mathbf{z}_{\text{prev}} \leftarrow \mathbf{z}_{\text{prev}}$ // $\mathbf{z}^{(\tau)} = \mathbf{z}^{(\tau-1)}$
    end if
end for
return $\mathbf{z}_{\text{prev}}$ // $\mathbf{z}^{(T)}$
```

walk. Consider a state space $z$ consisting of the integers, with probabilities

$$p(z^{(\tau+1)} = z^{(\tau)}) = 0.5 \tag{14.28}$$

$$p(z^{(\tau+1)} = z^{(\tau)} + 1) = 0.25 \tag{14.29}$$

$$p(z^{(\tau+1)} = z^{(\tau)} - 1) = 0.25 \tag{14.30}$$

where $z^{(\tau)}$ denotes the state at step $\tau$. If the initial state is $z^{(0)} = 0$, then by symmetry the expected state at time $\tau$ will also be zero $\mathbb{E}[z^{(\tau)}] = 0$, and similarly it is easily seen that $\mathbb{E}[(z^{(\tau)})^2] = \tau/2$. Thus, after $\tau$ steps, the random walk has travelled only a distance that on average is proportional to the square root of $\tau$. This square root dependence is typical of random walk behaviour and shows that random walks are very inefficient in exploring the state space. As we will see, a central goal in designing Markov chain Monte Carlo methods is to avoid random walk behaviour.

*Exercise 14.11*

### 14.2.2 Markov chains

Before discussing Markov chain Monte Carlo methods in more detail, it is useful to study some general properties of Markov chains. In particular, we ask under what circumstances will a Markov chain converge to the desired distribution. A first-order

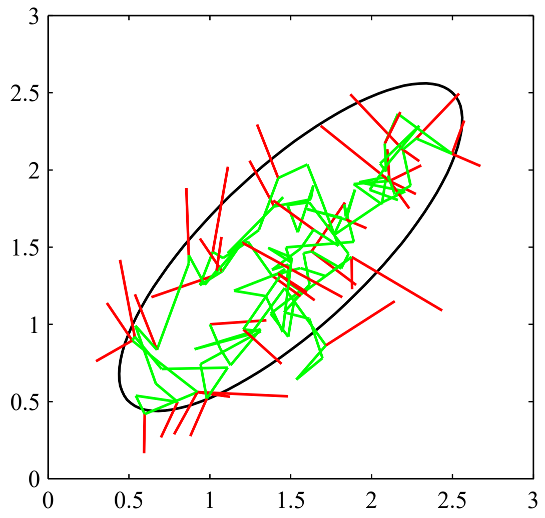


**Figure 14.9** A simple illustration using the Metropolis algorithm to sample from a Gaussian distribution whose one standard-deviation contour is shown by the ellipse. The proposal distribution is an isotropic Gaussian distribution whose standard deviation is 0.2. Steps that are accepted are shown as green lines, and rejected steps are shown in red. A total of 150 candidate samples are generated, of which 43 are rejected.

Markov chain is defined to be a series of random variables $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(M)}$ such that the following conditional independence property holds for $m \in \{1, \ldots, M-1\}$:

$$p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}) = p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)}), \tag{14.31}$$

*Figure 11.29* which can be represented as a directed graphical model in the form of a chain. We can then specify the Markov chain by giving the probability distribution for the initial variable $p(\mathbf{z}^{(0)})$ together with the conditional distributions for subsequent variables in the form of *transition probabilities* $T_m(\mathbf{z}^{(m)}, \mathbf{z}^{(m+1)}) \equiv p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)})$. A Markov chain is called *homogeneous* if the transition probabilities are the same for all $m$.

The marginal probability for a particular variable can be expressed in terms of the marginal probability for the previous variable in the chain:

$$p(\mathbf{z}^{(m+1)}) = \int p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)})p(\mathbf{z}^{(m)})\,\mathrm{d}\mathbf{z}^{(m)} \tag{14.32}$$

where the integral is replaced by a summation for discrete variables. A distribution is said to be invariant, or stationary, with respect to a Markov chain if each step in the chain leaves that distribution invariant. Thus, for a homogeneous Markov chain with transition probabilities $T(\mathbf{z}', \mathbf{z})$, the distribution $p^\star(\mathbf{z})$ is invariant if

$$p^\star(\mathbf{z}) = \int T(\mathbf{z}', \mathbf{z})p^\star(\mathbf{z}')\,\mathrm{d}\mathbf{z}'. \tag{14.33}$$

Note that a given Markov chain may have more than one invariant distribution. For instance, if the transition probabilities are given by the identity transformation, then any distribution will be invariant.

A sufficient (but not necessary) condition for ensuring that the required distribution $p(\mathbf{z})$ is invariant is to choose the transition probabilities to satisfy the property of *detailed balance*, defined by

$$p^{\star}(\mathbf{z})T(\mathbf{z},\mathbf{z}') = p^{\star}(\mathbf{z}')T(\mathbf{z}',\mathbf{z}) \tag{14.34}$$

for the particular distribution $p^{\star}(\mathbf{z})$. It is easily seen that a transition probability that satisfies detailed balance with respect to a particular distribution will leave that distribution invariant, because

$$\int p^{\star}(\mathbf{z}')T(\mathbf{z}',\mathbf{z})\,\mathrm{d}\mathbf{z}' = \int p^{\star}(\mathbf{z})T(\mathbf{z},\mathbf{z}')\,\mathrm{d}\mathbf{z}' \tag{14.35}$$

$$= p^{\star}(\mathbf{z})\int p(\mathbf{z}'|\mathbf{z})\,\mathrm{d}\mathbf{z}' \tag{14.36}$$

$$= p^{\star}(\mathbf{z}). \tag{14.37}$$

A Markov chain that respects detailed balance is said to be *reversible*.

Our goal is to use Markov chains to sample from a given distribution. We can achieve this if we set up a Markov chain such that the desired distribution is invariant. However, we must also require that for $m \to \infty$, the distribution $p(\mathbf{z}^{(m)})$ converges to the required invariant distribution $p^{\star}(\mathbf{z})$, irrespective of the choice of initial distribution $p(\mathbf{z}^{(0)})$. This property is called *ergodicity*, and the invariant distribution is then called the *equilibrium* distribution. Clearly, an ergodic Markov chain can have only one equilibrium distribution. It can be shown that a homogeneous Markov chain will be ergodic, subject only to weak restrictions on the invariant distribution and the transition probabilities (Neal, 1993).

In practice we often construct the transition probabilities from a set of 'base' transitions $B_1, \ldots, B_K$. This can be achieved through a mixture distribution of the form

$$T(\mathbf{z}',\mathbf{z}) = \sum_{k=1}^{K} \alpha_k B_k(\mathbf{z}',\mathbf{z}) \tag{14.38}$$

for some set of mixing coefficients $\alpha_1, \ldots, \alpha_K$ satisfying $\alpha_k \geqslant 0$ and $\sum_k \alpha_k = 1$. Alternatively, the base transitions may be combined through successive application, so that

$$T(\mathbf{z}',\mathbf{z}) = \sum_{\mathbf{z}_1} \ldots \sum_{\mathbf{z}_{n-1}} B_1(\mathbf{z}',\mathbf{z}_1)\ldots B_{K-1}(\mathbf{z}_{K-2},\mathbf{z}_{K-1})B_K(\mathbf{z}_{K-1},\mathbf{z}). \tag{14.39}$$

If a distribution is invariant with respect to each of the base transitions, then clearly it will also be invariant with respect to either of the $T(\mathbf{z}',\mathbf{z})$ given by (14.38) or (14.39). For the mixture (14.38), if each of the base transitions satisfies detailed balance, then the mixture transition $T$ will also satisfy detailed balance. This does not hold for the transition probability constructed using (14.39), although by symmetrizing the order of application of the base transitions, namely $B_1, B_2, \ldots, B_K, B_K, \ldots, B_2, B_1$, detailed balance can be restored. A common example of the use of composite transition probabilities is where each base transition changes only a subset of the variables.

### 14.2.3 The Metropolis–Hastings algorithm

Earlier we introduced the basic Metropolis algorithm without actually demonstrating that it samples from the required distribution. Before giving a proof, we first discuss a generalization, known as the *Metropolis–Hastings* algorithm (Hastings, 1970), which applies when the proposal distribution is no longer a symmetric function of its arguments. In particular at step $\tau$ of the algorithm, in which the current state is $\mathbf{z}^{(\tau)}$, we draw a sample $\mathbf{z}^\star$ from the distribution $q_k(\mathbf{z}|\mathbf{z}^{(\tau)})$ and then accept it with probability $A_k(\mathbf{z}^\star, \mathbf{z}^{(\tau)})$ where

$$A_k(\mathbf{z}^\star, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\widetilde{p}(\mathbf{z}^\star)q_k(\mathbf{z}^{(\tau)}|\mathbf{z}^\star)}{\widetilde{p}(\mathbf{z}^{(\tau)})q_k(\mathbf{z}^\star|\mathbf{z}^{(\tau)})}\right). \tag{14.40}$$

Here $k$ labels the members of the set of possible transitions being considered. Again, evaluating the acceptance criterion does not require knowledge of the normalizing constant $Z_p$ in the probability distribution $p(\mathbf{z}) = \widetilde{p}(\mathbf{z})/Z_p$. For a symmetric proposal distribution, the Metropolis–Hastings criterion (14.40) reduces to the standard Metropolis criterion given by (14.27). Metropolis–Hastings sampling is summarized in Algorithm 14.2.

We can show that $p(\mathbf{z})$ is an invariant distribution of the Markov chain defined by the Metropolis–Hastings algorithm by showing that detailed balance, defined by (14.34), is satisfied. Using (14.40) we have

$$\begin{aligned} p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z})A_k(\mathbf{z}', \mathbf{z}) &= \min\left(p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z}), p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}')\right) \\ &= \min\left(p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}'), p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z})\right) \\ &= p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}')A_k(\mathbf{z}, \mathbf{z}') \end{aligned} \tag{14.41}$$

as required.

The specific choice of proposal distribution can have a marked effect on the performance of the algorithm. For continuous state spaces, a common choice is a Gaussian centred on the current state, leading to an important trade-off in determining the variance parameter of this distribution. If the variance is small, then the proportion of accepted transitions will be high, but progress through the state space takes the form of a slow random walk leading to long correlation times. However, if the variance parameter is large, then the rejection rate will be high because, in the kind of complex problems we are considering, many of the proposed steps will be to states for which the probability $p(\mathbf{z})$ is low. Consider a multivariate distribution $p(\mathbf{z})$ having strong correlations between the components of $\mathbf{z}$, as illustrated in Figure 14.10. The scale $\rho$ of the proposal distribution should be as large as possible without incurring high rejection rates. This suggests that $\rho$ should be of the same order as the smallest length scale $\sigma_{\min}$. The system then explores the distribution along the more extended direction by means of a random walk, and so the number of steps to arrive at a state that is more or less independent of the original state is of order $(\sigma_{\max}/\sigma_{\min})^2$. In fact in two dimensions, the increase in rejection rate as $\rho$ increases is offset by the larger step sizes of those transitions that are accepted, and more generally for a multivariate Gaussian, the number of steps required to obtain independent samples scales like

**Algorithm 14.2:** Metropolis-Hastings sampling

```
Input: Unnormalized distribution $\widetilde{p}(\mathbf{z})$
       Proposal distributions $\{q_k(\mathbf{z}|\widehat{\mathbf{z}}) : k \in 1, \dots, K\}$
       Mapping from iteration index to distribution index $M(\cdot)$
       Initial state $\mathbf{z}^{(0)}$
       Number of iterations $T$
Output: $\mathbf{z} \sim \widetilde{p}(\mathbf{z})$
$\mathbf{z}_{\text{prev}} \leftarrow \mathbf{z}^{(0)}$
// Iterative message-passing
for $\tau \in \{1, \dots, T\}$ do
    $k \leftarrow M(\tau)$ // get distribution index for this iteration
    $\mathbf{z}^{\star} \sim q_k(\mathbf{z}|\mathbf{z}_{\text{prev}})$ // sample from proposal distribution
    $u \sim \mathcal{U}(0, 1)$ // sample from uniform
    if $\widetilde{p}(\mathbf{z}^{\star})q(\mathbf{z}_{\text{prev}}|\mathbf{z}^{\star}) \; / \; \widetilde{p}(\mathbf{z}_{\text{prev}})q(\mathbf{z}^{\star}|\mathbf{z}_{\text{prev}}) > u$ then
        $\mathbf{z}_{\text{prev}} \leftarrow \mathbf{z}^{\star}$ // $\mathbf{z}^{(\tau)} = \mathbf{z}^{\star}$
    else
        $\mathbf{z}_{\text{prev}} \leftarrow \mathbf{z}_{\text{prev}}$ // $\mathbf{z}^{(\tau)} = \mathbf{z}^{(\tau-1)}$
    end if
end for
return $\mathbf{z}_{\text{prev}}$ // $\mathbf{z}^{(T)}$
```

$(\sigma_{\max}/\sigma_2)^2$ where $\sigma_2$ is the second-smallest standard deviation (Neal, 1993). These details aside, if the length scales over which the distributions vary are very different in different directions, then the Metropolis Hastings algorithm can have very slow convergence.

### 14.2.4 Gibbs sampling

Gibbs sampling (Geman and Geman, 1984) is a simple and widely applicable Markov chain Monte Carlo algorithm and can be seen as a special case of the Metropolis–Hastings algorithm. Consider the distribution $p(\mathbf{z}) = p(z_1, \dots, z_M)$ from which we wish to sample, and suppose that we have chosen some initial state for the Markov chain. Each step of the Gibbs sampling procedure involves replacing the value of one of the variables by a value drawn from the distribution of that variable conditioned on the values of the remaining variables. Thus, we replace $z_i$ by a value drawn from the distribution $p(z_i|\mathbf{z}_{\setminus i})$, where $z_i$ denotes the $i$th component of $\mathbf{z}$, and $\mathbf{z}_{\setminus i}$ denotes $\{z_1, \dots, z_M\}$ but with $z_i$ omitted. This procedure is repeated either by cycling through the variables in some particular order or by choosing the

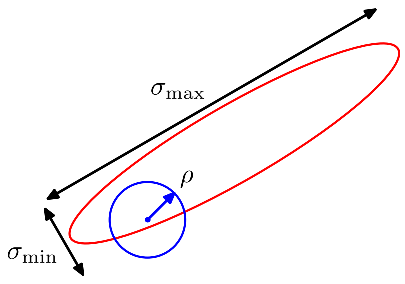


**Figure 14.10** Schematic illustration of using an isotropic Gaussian proposal distribution (blue circle) to sample from a correlated multivariate Gaussian distribution (red ellipse) having very different standard deviations in different directions, using the Metropolis–Hastings algorithm. To keep the rejection rate low, the scale $\rho$ of the proposal distribution should be of the order of the smallest standard deviation $\sigma_{\min}$, which leads to random walk behaviour in which the number of steps separating states that are approximately independent is of order $(\sigma_{\max}/\sigma_{\min})^2$ where $\sigma_{\max}$ is the largest standard deviation.

variable to be updated at each step at random from some distribution.

For example, suppose we have a distribution $p(z_1, z_2, z_3)$ over three variables, and at step $\tau$ of the algorithm we have selected values $z_1^{(\tau)}$, $z_2^{(\tau)}$, and $z_3^{(\tau)}$. We first replace $z_1^{(\tau)}$ by a new value $z_1^{(\tau+1)}$ obtained by sampling from the conditional distribution

$$p(z_1|z_2^{(\tau)}, z_3^{(\tau)}). \tag{14.42}$$

Next we replace $z_2^{(\tau)}$ by a value $z_2^{(\tau+1)}$ obtained by sampling from the conditional distribution

$$p(z_2|z_1^{(\tau+1)}, z_3^{(\tau)}) \tag{14.43}$$

so that the new value for $z_1$ is used straight away in subsequent sampling steps. Then we update $z_3$ with a sample $z_3^{(\tau+1)}$ drawn from

$$p(z_3|z_1^{(\tau+1)}, z_2^{(\tau+1)}) \tag{14.44}$$

and so on, cycling through the three variables in turn. Gibbs sampling is summarized in Algorithm 14.3.

To show that this procedure samples from the required distribution, we first note that the distribution $p(\mathbf{z})$ is an invariant of each of the Gibbs sampling steps individually and hence of the whole Markov chain. This follows since when we sample from $p(z_i|\mathbf{z}_{\setminus i})$, the marginal distribution $p(\mathbf{z}_{\setminus i})$ is clearly invariant because the value of $\mathbf{z}_{\setminus i}$ is unchanged. Also, each step by definition samples from the correct conditional distribution $p(z_i|\mathbf{z}_{\setminus i})$. Because these conditional and marginal distributions together specify the joint distribution, we see that the joint distribution is itself invariant.

The second requirement to be satisfied to ensure that the Gibbs sampling procedure samples from the correct distribution is that it is ergodic. A sufficient condition for ergodicity is that none of the conditional distributions are anywhere zero. If this is the case, then any point in $z$-space can be reached from any other point in a finite number of steps involving one update of each of the component variables. If this requirement is not satisfied, so that some of the conditional distributions have zeros, then ergodicity, if it applies, must be proven explicitly.

**Algorithm 14.3:** Gibbs sampling

```
Input: Initial values $\{z_i : i \in 1, \ldots, M\}$
       Conditional distributions $\{p(z_i|\{z_{j\neq i}\}) : i \in 1, \ldots, M\}$
       Number of iterations $T$
Output: Final values $\{z_i : i \in 1, \ldots, M\}$

for $\tau \in \{1, \ldots, T\}$ do
    for $i \in \{1, \ldots, M\}$ do
        $z_i \sim p(z_i|\{z_{j\neq i}\})$
    end for
end for
return $\{z_i : i \in 1, \ldots, M\}$
```

The distribution of initial states must also be specified to complete the algorithm, although samples drawn after many iterations will effectively become independent of this distribution. Of course, successive samples from the Markov chain will be highly correlated, and so to obtain samples that are nearly independent it will be necessary to sub-sample the sequence.

We can obtain the Gibbs sampling procedure as a particular instance of the Metropolis–Hastings algorithm as follows. Consider a Metropolis–Hastings sampling step involving the variable $z_k$ in which the remaining variables $\mathbf{z}_{\setminus k}$ remain fixed, and for which the transition probability from $\mathbf{z}$ to $\mathbf{z}^\star$ is given by $q_k(\mathbf{z}^\star|\mathbf{z}) = p(z_k^\star|\mathbf{z}_{\setminus k})$. Note that $\mathbf{z}^\star_{\setminus k} = \mathbf{z}_{\setminus k}$ because these components are unchanged by the sampling step. Also, $p(\mathbf{z}) = p(z_k|\mathbf{z}_{\setminus k})p(\mathbf{z}_{\setminus k})$. Thus, the factor that determines the acceptance probability in the Metropolis–Hastings (14.40) is given by

$$A(\mathbf{z}^\star, \mathbf{z}) = \frac{p(\mathbf{z}^\star)q_k(\mathbf{z}|\mathbf{z}^\star)}{p(\mathbf{z})q_k(\mathbf{z}^\star|\mathbf{z})} = \frac{p(z_k^\star|\mathbf{z}^\star_{\setminus k})p(\mathbf{z}^\star_{\setminus k})p(z_k|\mathbf{z}^\star_{\setminus k})}{p(z_k|\mathbf{z}_{\setminus k})p(\mathbf{z}_{\setminus k})p(z_k^\star|\mathbf{z}_{\setminus k})} = 1 \tag{14.45}$$

where we have used $\mathbf{z}^\star_{\setminus k} = \mathbf{z}_{\setminus k}$. Thus, the Metropolis–Hastings steps are always accepted.

As with the Metropolis algorithm, we can gain some insight into the behaviour of Gibbs sampling by investigating its application to a Gaussian distribution. Consider a correlated Gaussian in two variables, as illustrated in Figure 14.11, having conditional distributions of width $l$ and marginal distributions of width $L$. The typical step size is governed by the conditional distributions and will be of order $l$. Because the state evolves according to a random walk, the number of steps needed to obtain independent samples from the distribution will be of order $(L/l)^2$. Of course if the Gaussian distribution were uncorrelated, then the Gibbs sampling procedure would be optimally efficient. For this simple problem, we could rotate the coordinate system such that the new variables are uncorrelated. However, in practical applications

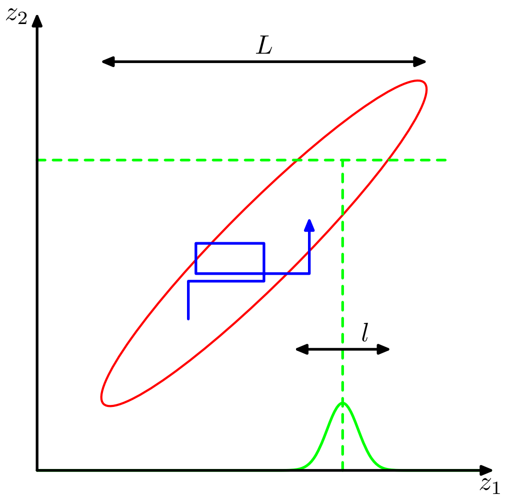


**Figure 14.11** Illustration of Gibbs sampling by alternate updates of two variables whose distribution is a correlated Gaussian. The step size is governed by the standard deviation of the conditional distribution (green curve), and is $\mathcal{O}(l)$, leading to slow progress in the direction of elongation of the joint distribution (red ellipse). The number of steps needed to obtain an independent sample from the distribution is $\mathcal{O}((L/l)^2)$.

it will generally be infeasible to find such transformations.

One approach to reducing the random walk behaviour in Gibbs sampling is called *over-relaxation* (Adler, 1981). In its original form, it applies to problems for which the conditional distributions are Gaussian, which represents a more general class of distributions than the multivariate Gaussian because, for example, the non-Gaussian distribution $p(z, y) \propto \exp(-z^2y^2)$ has Gaussian conditional distributions. At each step of the Gibbs sampling algorithm, the conditional distribution for a particular component $z_i$ has some mean $\mu_i$ and some variance $\sigma_i^2$. In the over-relaxation framework, the value of $z_i$ is replaced with

$$z_i' = \mu_i + \alpha_i(z_i - \mu_i) + \sigma_i(1 - \alpha_i^2)^{1/2}\nu \tag{14.46}$$

where $\nu$ is a Gaussian random variable with zero mean and unit variance, and $\alpha$ is a parameter such that $-1 < \alpha < 1$. For $\alpha = 0$, the method is equivalent to standard Gibbs sampling, and for $\alpha < 0$ the step is biased to the opposite side of the mean. This step leaves the desired distribution invariant because if $z_i$ has mean $\mu_i$ and variance $\sigma_i^2$, then so too does $z_i'$. The effect of over-relaxation is to encourage directed motion through state space when the variables are highly correlated. The framework of *ordered over-relaxation* (Neal, 1999) generalizes this approach to non-Gaussian distributions.

*Exercise 14.14*

The practical applicability of Gibbs sampling depends on the ease with which samples can be drawn from the conditional distributions $p(z_k|\mathbf{z}_{\setminus k})$. For probability distributions specified using directed graphical models, the conditional distributions for individual nodes depend only on the variables in the corresponding Markov blanket, as illustrated in Figure 14.12. For directed graphs, a wide choice of conditional distributions for the individual nodes conditioned on their parents will lead to conditional distributions for Gibbs sampling that are log concave. The adaptive rejection sampling methods discussed in Section 14.1.4 therefore provide a framework for Monte Carlo sampling from directed graphs with broad applicability.

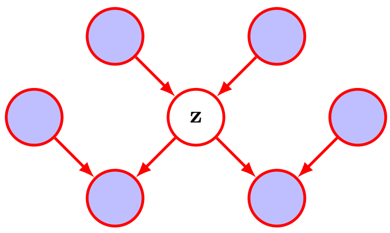


**Figure 14.12** The Gibbs sampling method requires samples to be drawn from the conditional distribution of a variable $\mathbf{z}$ conditioned on the remaining variables. For directed graphical models, this conditional distribution is a function of only the states of the nodes in the Markov blanket, shaded in blue, which comprises the parents, the children, and the co-parents.

Because the basic Gibbs sampling technique considers one variable at a time, there are strong dependencies between successive samples. At the opposite extreme, if we could draw samples directly from the joint distribution (an operation that we are supposing is intractable), then successive samples would be independent. We can hope to improve on the simple Gibbs sampler by adopting an intermediate strategy in which we sample successively from groups of variables rather than individual variables. This is achieved in the *blocking Gibbs* sampling algorithm by choosing blocks of variables, not necessarily disjoint, and then sampling jointly from the variables in each block in turn, conditioned on the remaining variables (Jensen, Kong, and Kjaerulff, 1995).

### 14.2.5 Ancestral sampling

For many models, the joint distribution $p(\mathbf{z})$ is conveniently specified in terms of a graphical model. For a directed graph with no observed variables, it is straightforward to sample from the joint distribution using the following *ancestral sampling* approach. The joint distribution is specified by

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i|\mathrm{pa}(i)) \tag{14.47}$$

where $\mathbf{z}_i$ are the set of variables associated with node $i$, and $\mathrm{pa}(i)$ denotes the set of variables associated with the parents of node $i$. To obtain a sample from the joint distribution, we make one pass through the set of variables in the order $\mathbf{z}_1, \ldots, \mathbf{z}_M$ sampling from the conditional distributions $p(\mathbf{z}_i|\mathrm{pa}(i))$. This is always possible because at each step, all the parent values will have been instantiated. After one pass through the graph, we will have obtained a sample from the joint distribution. This assumes that it is possible to sample from the individual conditional distributions at each node.

Now consider a directed graph in which some of the nodes, which comprise the *evidence set* $\mathcal{E}$, are instantiated with observed values. We can in principle extend the above procedure, at least for nodes representing discrete variables, to give the following *logic sampling* approach (Henrion, 1988), which can be seen as a special case of *importance sampling*. At each step, when a sampled value is obtained for a variable $\mathbf{z}_i$ whose value is observed, the sampled value is compared to the observed value, and if they agree then the sample value is retained and the algorithm proceeds to the next variable in turn. However, if the sampled value and the observed value disagree, then the whole sample so far is discarded and the algorithm starts again

*Section 14.1.5*

with the first node in the graph. This algorithm samples correctly from the posterior distribution because it corresponds simply to drawing samples from the joint distribution of hidden variables and data variables and then discarding those samples that disagree with the observed data (with the slight saving of not continuing with the sampling from the joint distribution as soon as one contradictory value is observed). However, the overall probability of accepting a sample from the posterior decreases rapidly as the number of observed variables increases and as the number of states that those variables can take increases, and so this approach is rarely used in practice.

An improvement on this approach is called *likelihood weighted sampling* (Fung and Chang, 1990; Shachter and Peot, 1990). It is based on ancestral sampling combined with importance sampling. For each variable in turn, if that variable is in the evidence set, then it is just set to its instantiated value. If it is not in the evidence set, then it is sampled from the conditional distribution $p(\mathbf{z}_i|\mathrm{pa}(i))$ in which the conditioning variables are set to their currently sampled values. The weighting associated with the resulting sample $\mathbf{z}$ is then given by

*Exercise 14.15*

$$r(\mathbf{z}) = \prod_{\mathbf{z}_i \notin \mathbf{e}} \frac{p(\mathbf{z}_i|\mathrm{pa}(i))}{p(\mathbf{z}_i|\mathrm{pa}(i))} \prod_{\mathbf{z}_i \in \mathbf{e}} \frac{p(\mathbf{z}_i|\mathrm{pa}(i))}{1} = \prod_{\mathbf{z}_i \in \mathbf{e}} p(\mathbf{z}_i|\mathrm{pa}(i)). \tag{14.48}$$

This method can be further extended using *self-importance sampling* (Shachter and Peot, 1990) in which the importance sampling distribution is continually updated to reflect the current estimated posterior distribution.

## 14.3. Langevin Sampling

The Metropolis–Hastings algorithm draws samples from a probability distribution by creating a Markov chain of candidate samples using a proposal distribution and then accepting or rejecting them using the criterion (14.40). This can be relatively inefficient since the proposal distribution is often a simple, fixed distribution that can generate updates in any direction in the data space, leading to a random walk.

We have seen that when training neural networks, it is hugely advantageous to make use of the gradient of the log likelihood with respect to the learnable parameters of the model in order to maximize the likelihood function. By analogy, we can introduce Markov chain sampling algorithms that make use of the gradient of the probability density with respect to the data vector so as to take steps that preferentially move towards regions of higher probability. One such technique is called *Hamiltonian Monte Carlo*, also known as *hybrid Monte Carlo*. This again makes use of a Metropolis acceptance test (Duane *et al.*, 1987; Bishop, 2006). Here we will focus on a different approach that is widely used in deep learning, called *Langevin sampling*. Although it avoids the use of an acceptance test, the algorithm has to be designed carefully to ensure that the resulting samples are unbiased. An important application of Langevin sampling arises in the context of machine learning models defined in terms of energy functions.

### 14.3.1 Energy-based models

Many generative models can be expressed as conditional probability distributions $p(\mathbf{x}|\mathbf{w})$ where $\mathbf{x}$ is the data vector and $\mathbf{w}$ represents a vector of learnable parameters. Such models can be trained by maximizing the corresponding likelihood function defined with respect to a training data set. However, to represent a valid probability distribution, the model must satisfy

$$\int p(\mathbf{x}|\mathbf{w})p(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1. \tag{14.49}$$

Ensuring that this requirement is met can significantly limit the allowable forms for the model. If we put aside the normalization constraint then we can consider a much broader class of models called *energy-based models* (LeCun *et al.*, 2006). Suppose we have a function $E(\mathbf{x}, \mathbf{w})$, called the *energy function*, which is a real-valued function of its arguments but which has no other constraints. The exponential $\exp\{-E(\mathbf{x}, \mathbf{w})\}$ is a non-negative quantity and can therefore be viewed as an unnormalized probability distribution over $\mathbf{x}$. Here the introduction of the minus sign in the exponent is simply a convention, and it means that higher values of energy correspond to lower values of probability. We can then define a normalized distribution using

$$p(\mathbf{x}|\mathbf{w}) = \frac{1}{Z(\mathbf{w})}\exp\{-E(\mathbf{x}, \mathbf{w})\} \tag{14.50}$$

*Exercise 14.16* where the normalizing constant $Z(\mathbf{w})$, known as the *partition function*, is defined by

$$Z(\mathbf{w}) = \int \exp\{-E(\mathbf{x}, \mathbf{w})\}\,\mathrm{d}\mathbf{x}. \tag{14.51}$$

The energy function is often modelled using a deep neural network with input vector $\mathbf{x}$ and a scalar output $E(\mathbf{x}, \mathbf{w})$, where $\mathbf{w}$ represents the weights and biases in the network.

Note that the partition function depends on $\mathbf{w}$, which creates problems for training. For example, the log likelihood function for a data set $\mathcal{D} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ of i.i.d. data has the form *Exercise 14.17*

$$\ln p(\mathcal{D}|\mathbf{w}) = -\sum_{n=1}^{N} E(\mathbf{x}_n, \mathbf{w}) - N \ln Z(\mathbf{w}). \tag{14.52}$$

To compute the gradient of $\ln p(\mathcal{D}|\mathbf{w})$ with respect to $\mathbf{w}$, we need to know the form of $Z(\mathbf{w})$. However, for many choices of the energy function $E(\mathbf{x}, \mathbf{w})$, it will be impractical to evaluate the partition function in (14.51) because this involves integrating (or summing for discrete variables) over all the whole of $\mathbf{x}$-space. The term ‘energy-based model’ is generally used for models where this integral is intractable. Note, however, that probabilistic models can be seen as special cases of energy-based models, and therefore many of the models discussed in this book can be viewed as energy-based models. The big advantage of energy-based models, therefore, is their flexibility in that they bypass the requirement for normalization. A corresponding disadvantage, however, is that since the normalizing constant is unknown, they can be more difficult to train.

## 14.3.2 Maximizing the likelihood

Various approximation methods have been developed to train energy-based models without having to evaluate the partition function (Song and Kingma, 2021). Here we look at techniques based on Markov chain Monte Carlo. An alternative approach, called score matching, will be discussed in the context of diffusion models. *Chapter 20*

We have seen that for energy-based models, the likelihood function cannot be evaluated explicitly due to the unknown partition function $Z(\mathbf{w})$. However, we can make use of Monte Carlo sampling methods to approximate the gradient of the log likelihood with respect to the model parameters. Once an energy-based model has been trained, by whatever means, we also need a way to draw samples from the model, and again we can make use of Monte Carlo methods.

Using (14.50), the gradient, with respect to the model parameters, of the log likelihood function for an energy-based model can be written in the form

$$\nabla_{\mathbf{w}} \ln p(\mathbf{x}|\mathbf{w}) = -\nabla_{\mathbf{w}} E(\mathbf{x},\mathbf{w}) - \nabla_{\mathbf{w}} \ln Z(\mathbf{w}). \tag{14.53}$$

This is the likelihood function for a single data point $\mathbf{x}$, but in practice we want to maximize the likelihood defined over a training set of data points drawn from some unknown distribution $p_{\mathcal{D}}(\mathbf{x})$. If we assume the data points are i.i.d., then we can consider the gradient of the expectation of the log likelihood with respect to $p_{\mathcal{D}}(\mathbf{x})$, which is then given by

$$\mathbb{E}_{\mathbf{x}\sim p_{\mathcal{D}}}\left[\nabla_{\mathbf{w}} \ln p(\mathbf{x}|\mathbf{w})\right] = -\mathbb{E}_{\mathbf{x}\sim p_{\mathcal{D}}}\left[\nabla_{\mathbf{w}} E(\mathbf{x},\mathbf{w})\right] - \nabla_{\mathbf{w}} \ln Z(\mathbf{w}) \tag{14.54}$$

where we have made use of the fact that the final term $-\nabla_{\mathbf{w}} \ln Z(\mathbf{w})$ does not depend on $\mathbf{x}$ and can therefore be taken outside the expectation. The partition function $Z(\mathbf{w})$ is assumed to be unknown, but we can make use of (14.51) and rearrange to obtain *Exercise 14.18*

$$-\nabla_{\mathbf{w}} \ln Z(\mathbf{w}) = \int \{\nabla_{\mathbf{w}} E(\mathbf{x},\mathbf{w})\}\, p(\mathbf{x}|\mathbf{w})\,\mathrm{d}\mathbf{x}. \tag{14.55}$$

The right-hand side of (14.55) corresponds to an expectation over the model distribution $p(\mathbf{x}|\mathbf{w})$ given by

$$\int \{\nabla_{\mathbf{w}} E(\mathbf{x},\mathbf{w})\}\, p(\mathbf{x}|\mathbf{w})\,\mathrm{d}\mathbf{x} = \mathbb{E}_{\mathbf{x}\sim\mathcal{M}}\left[\nabla_{\mathbf{w}} E(\mathbf{x},\mathbf{w})\right]. \tag{14.56}$$

*Exercise 14.18* Combining (14.54), (14.55), and (14.56) we obtain

$$\begin{aligned}\nabla_{\mathbf{w}}\mathbb{E}_{\mathbf{x}\sim p_{\mathcal{D}}}\left[\ln p(\mathbf{x}|\mathbf{w})\right] = -&\mathbb{E}_{\mathbf{x}\sim p_{\mathcal{D}}}\left[\nabla_{\mathbf{w}} E(\mathbf{x},\mathbf{w})\right] \\ &+ \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{M}}(\mathbf{x})}\left[\nabla_{\mathbf{w}} E(\mathbf{x},\mathbf{w})\right].\end{aligned} \tag{14.57}$$

This result is illustrated in Figure 14.13, and has a nice interpretation, as follows. Our goal is to find values for the parameters $\mathbf{w}$ that maximize the likelihood function, and therefore consider a small change to $\mathbf{w}$ in the direction of the gradient $\nabla_{\mathbf{w}} \ln p(\mathbf{x}|\mathbf{w})$. From (14.57) we see that expected value of this gradient can be expressed as two terms, having opposite signs. The first term on the right-hand side of (14.57) acts

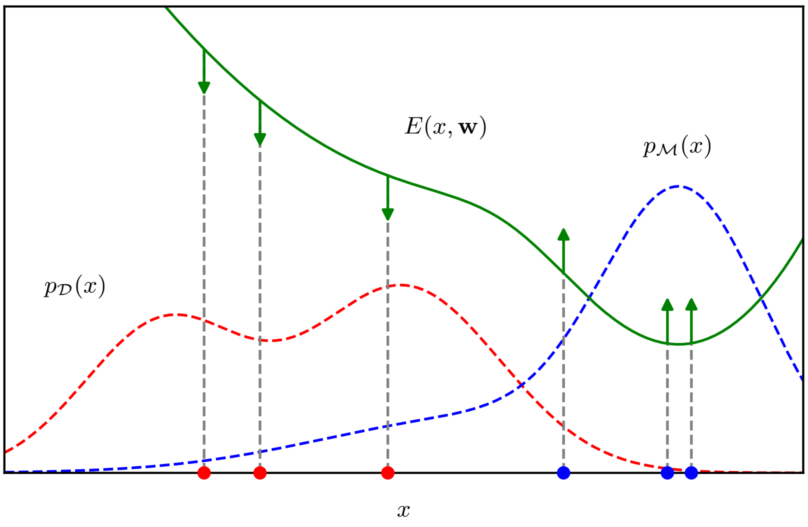


Figure 14.13 Illustration of the training of an energy-based model by maximizing the likelihood, showing the energy function $E(x, \mathbf{w})$ in green along with the associated model distribution $p_{\mathcal{M}}(x)$ and the true data distribution $p_{\mathcal{D}}(x)$. Increasing the expected log likelihood by using (14.57) pushes the energy function up at points corresponding to samples from the model (shown as blue dots) and pushes it down at points corresponding to samples from the data set (shown as red dots).

to decrease $E(\mathbf{x}, \mathbf{w})$, and therefore to increase the probability density defined by the model, for points $\mathbf{x}$ drawn from $p_{\mathcal{D}}(\mathbf{x})$. The second term on the right-hand side of (14.57) acts to increase the value of $E(\mathbf{x}, \mathbf{w})$, and therefore to decrease the probability density defined by the model, for data points drawn from the model itself. In regions where the model density exceeds the training data density, the net effect will be to increase the energy and therefore reduce the probability. Conversely, in regions where training data density exceeds the model density, the net effect will be to reduce the energy and therefore increase the probability density. Together these two terms move probability mass away from regions where there is a low density of training data and towards regions of high data density, as desired. The two terms will be equal in magnitude when the model distribution matches the data distribution, at which point the gradient on the left-hand-side of (14.57) will equal zero.

### 14.3.3 Langevin dynamics

When applying (14.57) as a practical training method, we need to approximate the two terms on the right-hand side. For any given value of $\mathbf{x}$, we can evaluate $\nabla_{\mathbf{w}} E(\mathbf{x}, \mathbf{w})$ using automatic differentiation. For the first term in (14.57), we can use the training data set to estimate the expectation over $\mathbf{x}$:

$$\mathbb{E}_{\mathbf{x}\sim p_{\mathcal{D}}}\left[\nabla_{\mathbf{w}} E(\mathbf{x}, \mathbf{w})\right] \simeq \frac{1}{N}\sum_{n=1}^{N} \nabla_{\mathbf{w}} E(\mathbf{x}_n, \mathbf{w}). \tag{14.58}$$

The second term is more challenging because we need to draw samples from the model distribution defined by an energy function whose corresponding partition function is intractable. This can be done using Markov chain Monte Carlo methods. One popular approach is called *stochastic gradient Langevin dynamics* or simply *Langevin sampling* (Parisi, 1981; Welling and Teh, 2011). This term depends on the distribution $p(\mathbf{x}|\mathbf{w})$ only through the *score function*, which is defined to be the gradient of the log likelihood with respect to the data vector $\mathbf{x}$, and is given by

$$\mathbf{s}(\mathbf{x}, \mathbf{w}) = \nabla_{\mathbf{x}} \ln p(\mathbf{x}|\mathbf{w}). \tag{14.59}$$

It is worth emphasising that this gradient is taken with respect to the data point $\mathbf{x}$ and is therefore not the usual gradient with respect to the learnable parameters $\mathbf{w}$. If we substitute (14.50) into (14.59) we obtain

$$\mathbf{s}(\mathbf{x}, \mathbf{w}) = -\nabla_{\mathbf{x}} E(\mathbf{x}, \mathbf{w}) \tag{14.60}$$

where we see that the partition function no longer appears at it is independent of $\mathbf{x}$.

We start by drawing an initial value $\mathbf{x}^{(0)}$ from a prior distribution, and then we iterate the following Markov chain steps:

$$\mathbf{x}^{(\tau+1)} = \mathbf{x}^{(\tau)} + \eta \nabla_{\mathbf{x}} \ln p(\mathbf{x}^{(\tau)}, \mathbf{w}) + \sqrt{2\eta}\boldsymbol{\epsilon}^{(\tau)}, \qquad \tau \in 1, \ldots, \mathcal{T} \tag{14.61}$$

where $\boldsymbol{\epsilon}^{(\tau)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are independent samples from a zero-mean, unit-covariance Gaussian distribution, and the parameter $\eta$ controls the step size. Each iteration of the Langevin equation takes a step in the direction of the gradient of the log likelihood, and then adds Gaussian noise. It can be show that, in the limits of $\eta \to 0$ and $\mathcal{T} \to \infty$, the value of $\mathbf{z}^{(\mathcal{T})}$ is an independent sample from the distribution $p(\mathbf{x})$. Langevin sampling is summarized in Algorithm 14.4.

We can repeat the process to generate a set of samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ and then approximate the second term in (14.57) using

$$\mathbb{E}_{\mathbf{x} \sim p_{\mathcal{M}}(\mathbf{x})} \left[ \nabla_{\mathbf{w}} E(\mathbf{x}, \mathbf{w}) \right] \simeq \frac{1}{M} \sum_{m=1}^{M} \nabla_{\mathbf{w}} E(\mathbf{x}_m, \mathbf{w}). \tag{14.62}$$

Running long Markov chains to generate independent samples can be computationally expensive, and so we need to consider practical approximations. One approach is called *contrastive divergence* (Hinton, 2002). Here the samples used to evaluate (14.62) are obtained by running a Monte Carlo chain starting with one of the training data points $\mathbf{x}_n$. If the chain is run for a large number of steps, then the resulting value will be essentially an unbiased sample from the model distribution. Instead Hinton (2002) proposes running for only a few steps of Monte Carlo, perhaps even as few as one step, which is computationally much less costly. The resulting sample will be far from unbiased and will lie close to the data manifold. As a result, the effect of using gradient descent will be to shape the energy surface, and hence the probability density, only in the neighbourhood of the data manifold. This can prove effective for tasks such as discrimination but is expected to be less effective in learning a generative model.

**Algorithm 14.4:** Langevin sampling

```
Input: Initial value $\mathbf{x}^{(0)}$
       Probability density $p(\mathbf{x}, \mathbf{w})$
       Learning rate parameter $\eta$
       Number of iterations $T$
Output: Final value $\mathbf{x}^{(T)}$
$\mathbf{x} \leftarrow \mathbf{x}_0$
for $\tau \in \{1, \ldots, T\}$ do
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \mathbf{I})$
    $\mathbf{x} \leftarrow \mathbf{x} + \eta \nabla_{\mathbf{x}} \ln p(\mathbf{x}, \mathbf{w}) + \sqrt{2\eta}\boldsymbol{\epsilon}$
end for
return $\mathbf{x}$ // Final value $\mathbf{x}^{(T)}$
```

## Exercises

**14.1** ($\star$) Show that $\overline{f}$ defined by (14.2) is an unbiased estimator, in other words that the expectation of the right-hand side is equal to $\mathbb{E}[f(\mathbf{z})]$.

**14.2** ($\star$) Show that $\overline{f}$ defined by (14.2) has variance given by (14.4).

**14.3** ($\star$) Suppose that $z$ is a random variable with uniform distribution over $(0, 1)$ and that we transform $z$ using $y = h^{-1}(z)$ where $h(y)$ is given by (14.6). Show that $y$ has the distribution $p(y)$.

**14.4** ($\star\star$) Given a random variable $z$ that is uniformly distributed over $(0, 1)$, find a transformation $y = f(z)$ such that $y$ has a Cauchy distribution given by (14.8).

**14.5** ($\star\star$) Suppose that $z_1$ and $z_2$ are uniformly distributed over the unit circle, as shown in Figure 14.3, and that we make the change of variables given by (14.10) and (14.11). Show that $(y_1, y_2)$ will be distributed according to (14.12).

**14.6** ($\star\star$) Let $\mathbf{z}$ be a $D$-dimensional random variable having a Gaussian distribution with zero mean and unit covariance matrix, and suppose that the positive definite symmetric matrix $\boldsymbol{\Sigma}$ has the Cholesky decomposition $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, where $\mathbf{L}$ is a lower-triangular matrix (i.e., one with zeros above the leading diagonal). Show that the variable $\mathbf{y} = \boldsymbol{\mu} + \mathbf{L}\mathbf{z}$ has a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. This provides a technique for generating samples from a general multivariate Gaussian using samples from a univariate Gaussian having zero mean and unit variance.

**14.7** ($\star\star$) In this exercise, we show more carefully that rejection sampling does indeed draw samples from the desired distribution $p(\mathbf{z})$. Suppose the proposal distribution is $q(\mathbf{z})$. Show that the probability of a sample value $\mathbf{z}$ being accepted is given by