

integrals at reduced cost. Sometimes we use this to provide a significant speedup to a costly but tractable sum, as in the case when we subsample the full training cost with minibatches. In other cases, our learning algorithm requires us to approximate an intractable sum or integral, such as the gradient of the log partition function of an undirected model. In many other cases, sampling is actually our goal, in the sense that we want to train a model that can sample from the training distribution.

### 17.1.2 Basics of Monte Carlo Sampling

When a sum or an integral cannot be computed exactly (for example the sum has an exponential number of terms and no exact simplification is known) it is often possible to approximate it using Monte Carlo sampling. The idea is to view the sum or integral as if it was an expectation under some distribution and to *approximate the expectation by a corresponding average*. Let

$$s = \sum_{\mathbf{x}} p(\mathbf{x}) f(\mathbf{x}) = E_p[f(\mathbf{x})] \quad (17.1)$$

or

$$s = \int p(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} = E_p[f(\mathbf{x})] \quad (17.2)$$

be the sum or integral to estimate, rewritten as an expectation, with the constraint that  $p$  is a probability distribution (for the sum) or a probability density (for the integral) over random variable  $\mathbf{x}$ .

We can approximate  $s$  by drawing  $n$  samples  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$  from  $p$  and then forming the empirical average

$$\hat{s}_n = \frac{1}{n} \sum_{i=1}^n f(\mathbf{x}^{(i)}). \quad (17.3)$$

This approximation is justified by a few different properties. The first trivial observation is that the estimator  $\hat{s}$  is unbiased, since

$$\mathbb{E}[\hat{s}_n] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[f(\mathbf{x}^{(i)})] = \frac{1}{n} \sum_{i=1}^n s = s. \quad (17.4)$$

But in addition, the **law of large numbers** states that if the samples  $\mathbf{x}^{(i)}$  are i.i.d., then the average converges almost surely to the expected value:

$$\lim_{n \rightarrow \infty} \hat{s}_n = s, \quad (17.5)$$