2

Monte Carlo basics

The Monte Carlo method is built on probability and statistics. Here, we introduce the few basic concepts of probability and statistics necessary to understand the concept of sampling from a probability distribution. We then discuss several useful techniques to do this sampling from distribution functions that depend on just a few random variables. Functions depending on a large number of random variables require a different technique. For these functions we discuss the use of Markov chain sampling.

2.1 Some probability concepts

As a stochastic procedure, a Monte Carlo simulation generates a set of random events. Sometimes the order in which the events are generated matters; other times, it does not. The theory of Monte Carlo sampling, the procedure by which we access the random events, is based on probability theory. We begin by discussing a few essential basic concepts from probability theory.

In probability theory, the set of all possible *outcomes* $\{\chi_1, \chi_2, \ldots, \chi_n, \ldots\}$ of a real or imagined experiment defines a *sample space*. If our experiment was, for example, the roll of a pair of dice, there would be 36 outcomes in the sample space. An *event* is a set of one or more outcomes of this space that satisfies some criterion. The *probability* of an event A is a number assigned to the event in a way consistent with three axioms: (1) $0 \leq P(A) \leq 1$; (2) if the event includes all possible outcomes in the sample space, then $P(A) = 1$; and (3) if an event A breaks into events B and C that share no common outcome, then $P(A) = P(B) + P(C)$. In our roll of the dice experiment, if the event A were those outcomes whose sum equals 3, it would encompass two outcomes, and $P(A) = \frac{1}{18}$. If the event A were those outcomes whose sums are odd, $P(A) = \frac{1}{2}$.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

A consequence of the axioms is that the probability maps events to the interval [0, 1] in such a way that the sum over a set $\{A_1, A_2, \dots\}$ of mutually exclusive events that covers the sample space equals 1. Two events A_i and A_j are *mutually exclusive*, that is, share no common outcome, if and only if the occurrence of A_i implies that A_j does not occur and vice versa. Because the sum is 1, at least one event is possible.

A *random variable* X maps an outcome to a real number. The events

$$A = \{X(\chi) \leq x\}, \quad B = \{x_1 < X(\chi) < x_2\}, \text{ and } \quad C = \{X(\chi) = x_0\}$$

are the set of outcomes χ such that $X(\chi)$ is less than or equal to x, between x_1 and x_2, and equal to x_0, respectively. Normally, we work at the level of random variables; that is, our experiments generate events with numbers assigned. In the following, we write the probability $P(X(\chi) = x)$ as $P(x)$ or $P(X)$; in other words, the random variable and its value are used interchangeably. Also, we often do not distinguish an outcome (an elemental event) from an event.

The *cumulative distribution function* of a random variable X, $F_X(x)$, is defined by

$$F_X(x) = P(X \leq x). \tag{2.1}$$

More generally,

$$P(a < X \leq b) = F_X(b) - F_X(a). \tag{2.2}$$

$F_X(x)$ is a positive, monotonic, nondecreasing function of x defined over the entire real axis with the properties $F_X(-\infty) = 0$ and $F_X(\infty) = 1$. If $F_X(x)$ is everywhere differentiable, the random variable is continuous. Then a probability density $f_X(x)$ exists and satisfies

$$f_X(x) = \frac{dF_X(x)}{dx}.$$

This expression says that $f_X(x)dx$ is the probability that x is in the interval $[x, x + dx]$. By convention, $F_X(x)$ is continuous from the right; that is, $F_X(x) = \lim_{\epsilon \to 0} F_X(x + \epsilon)$. If $F_X(x)$ is step-wise continuous, with jumps $f_1, f_2, \dots$ at $x_1, x_2, \dots$, then the random variable is discrete. If discrete, we often write $f_X(x_i)$ as simply f_i. We do not explicitly consider distributions of mixed random variable types. Doing so is relatively straightforward.

Often it is useful to change from one random variable to another. Let us consider, for example, $Y = y(X)$ where $y(x)$ is a nondecreasing function of x. If we know $F_X(x)$ and $f_X(x)$, what are $F_Y(y)$ and $f_Y(y)$? Because $y(X) \leq y(x)$ when $X \leq x$ and vice versa, it follows that

$$P(y(X) = Y \leq y(x)) = P(X \leq x),$$

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

that is, $F_Y(y) = F_X(x)$. By differentiation, we obtain

$$f_Y(y)\frac{dy}{dx} = f_X(x).$$

Because $y(x)$ is nondecreasing, the derivative of y with respect to x is positive and both sides of the equation are positive.

If the change of variables is nonincreasing, it is easy to show that

$$F_Y(y) = 1 - F_X(x)$$

and

$$f_Y(y)\frac{dy}{dx} = -f_X(x).$$

More generally, we write

$$f_Y(y)\left|\frac{dy}{dx}\right| = f_X(x),$$

or for the inverse mapping $f_Y(y) = f_X(x)\left|\frac{dx}{dy}\right|$.

If we have two random variables X and Y, we represent the probability of both occurring as $P(X, Y)$. Of course, $P(Y, X) = P(X, Y)$. This bivariate function is called the *joint probability* of X and Y and is related to $P(X)$ and $P(Y)$ by

$$P(X) = \sum_Y P(X, Y) \quad \text{and} \quad P(Y) = \sum_X P(X, Y). \tag{2.3}$$

If $P(X, Y) = P(X)P(Y)$, the random variables are said to be *statistically independent*.

The *conditional probability* $P(X|Y)$ of X given that Y occurred is related to the joint probability by

$$P(X|Y) = \frac{P(X, Y)}{P(Y)}. \tag{2.4}$$

Similarly, the conditional probability of Y given that X occurred is

$$P(Y|X) = \frac{P(X, Y)}{P(X)}. \tag{2.5}$$

By comparing these two equations we find that

$$P(X|Y)P(Y) = P(Y|X)P(X), \tag{2.6}$$

and consequently

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}, \tag{2.7}$$

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

which is called *Bayes's theorem*. We note that

$$\sum_X P(X|Y) = \sum_X P(X,Y)/P(Y) = 1. \tag{2.8}$$

Statistical independence implies $P(X|Y) = P(X)$, provided $P(Y) > 0$.

A bivariate cumulative distribution is defined by

$$F_{XY}(x,y) = P(X \le x, Y \le y).$$

The analog to (2.2) is

$$P(a < X \le b, c < Y \le d) = F_{XY}(b,d) - F_{XY}(a,d) - F_{XY}(b,c) + F_{XY}(a,c)$$

and the associated densities are

$$f_{XY}(x,y) = \frac{\partial^2 F_{XY}(x,y)}{\partial x \partial y}. \tag{2.9}$$

Distributions and densities for more than two random variables are defined analogously.

If we change both random variables to new random variables U and V,

$$U = g(X,Y), \qquad V = h(X,Y),$$

and if the inverse mappings are

$$X = p(U,V), \qquad Y = q(U,V),$$

then their densities are related by

$$f_{UV}(u,v) = f_{XY}(x,y) \left| \frac{\partial(p,q)}{\partial(u,v)} \right|.$$

Note the absolute value of the Jacobian of the transformation.

It is often useful to integrate out one or more random variables from a multivariate distribution. The result is a new distribution called the *marginal probability density*. Let us say $f_{XY}(x,y)$ is the joint distribution of two random variables X and Y and suppose we are interested in the probability that $a < X < b$. This event occurs only when $a < X < b$ and $-\infty < Y < \infty$. For the continuum and discrete cases

$$P(a < X < b, -\infty < Y < \infty) = \begin{cases} \int_a^b \int_{-\infty}^{\infty} f_{XY}(x,y)\,dy\,dx, \\ \sum_{a<x<b} \sum_y f_{XY}(x,y). \end{cases}$$

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

The result of either the integration or summation is a function of x alone,

$$f_X(x) = \begin{cases} \int_{-\infty}^{\infty} f_{XY}(x,y)dy, \\ \sum_y f_{XY}(x,y). \end{cases}$$

This function $f_X(x)$ is the *marginal probability distribution* of X.

With the concept of a marginal distribution and the multivariate definition of a conditional probability distribution function,

$$f(x_1,\ldots,x_k|x_{k+1},\ldots,x_n) = \frac{f(x_1,\ldots,x_n)}{f(x_{k+1},\ldots,x_n)},$$

we can derive a number of interesting relations. One is the *Chapman-Kolmogoroff equation*

$$\int f(x_3|x_1,x_2)f(x_2|x_1)dx_2 = f(x_3|x_1).$$

To derive it we use the definition of a conditional density and write the left-hand side of the above as

$$\int \frac{f(x_1,x_2,x_3)}{f(x_1,x_2)}\frac{f(x_1,x_2)}{f(x_1)}dx_2.$$

Reusing this definition and invoking the one for a marginal distribution, we find

$$\int f(x_2,x_3|x_1)dx_2 = f(x_3|x_1).$$

We can use a similar reasoning to prove expressions such as

$$\int\int f(x_4,x_3,x_2|x_1)f(x_3,x_2|x_1)dx_3dx_2 = f(x_4|x_1).$$

We also note that repeated application of the definition of the multivariate conditional probability produces the *conditional probability chain rule*

$$f(x_1,\ldots,x_n) = f(x_n|x_{n-1},\ldots,x_1)\cdots f(x_2|x_1)f(x_1).$$

2.2 Random sampling

The most fundamental Monte Carlo concept we need is that of *random sampling*. Random sampling, or *sampling* for short, bridges probability theory and statistics. Implicit in the concept of probability is the existence of some experiment that produces a realization of a random variable from a set of possible outcomes.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

If executing this experiment N times produces the value x of the random variable X N_x times, then the ratio N_x/N is an empirical measure of $P(x)$. To sample a probability distribution means that we generate events with a frequency proportional to it.

An alternative approach to random sampling is to consider N experiments simultaneously (an *ensemble*). Associated with the i-th experiment is the random variable X_i, which is identical to the random variable X and by assumption has the same distribution as X; that is, $P_i(X_i) = P(X)$. The joint distribution of these N random variables is $P(X_1, X_2, \ldots, X_N)$. The statistical independence of the random variables means

$$P(X_1, X_2, \ldots, X_N) = P_1(X_1)P_2(X_2)\cdots P_N(X_N).$$

The ensemble of N experiments thus produces a set of outcomes $\chi = \{\chi_1, \chi_2, \ldots, \chi_N\}$ such that the i-th random variable takes the values $X_i(\chi) = X(\chi_i)$. The combination of independence and identical distributions is a hallmark of random samples.

There are various procedures for sampling a probability distribution, and they almost always assume the existence of a *random number generator*, which is some numerical procedure producing a random variable uniformly distributed over the interval $[0, 1]$.[1] In other words, it samples $u(x)dx = dx$ over this interval.[2] The numbers produced are actually quasi-random as opposed to being truly random: if a generator is sampled long enough, the numbers repeat themselves. For most applications the length of useful sequences is sufficient.

In Fig. 2.1 we show schematically the type of histogram a typical random number generator produces. It is flat to within statistical fluctuations. If the random variables are uniformly distributed over $[0, 1]$, then pairs of them should be uniformly distributed over the unit square; triplets, over the unit cube, etc. Generally, the problem with a random number generator is not an inability to produce a reasonably flat histogram, but its ability to produce independent random numbers. Correlations can exist between pairs or triplets or higher order tuples. Volumes have been written about what is a good random number generator and how it is constructed. We do not discuss these issues but simply assume that we have access to a good one. Today, vendor-supplied generators accompanying modern compilers and others found in recent textbooks are adequate. Serious, large-scale simulations might require other choices.

[1] In practice, random number generators return numbers from the interval $(0, 1]$, $[0, 1)$, or $(0, 1)$. Which is the case can matter if we need, for example, the logarithm of the random number. We ignore this practical issue in our discussions.

[2] Throughout we use $u(x)$ to represent a uniform distribution of the random variable over the interval $[0, 1]$ and ζ to represent a sample drawn from this distribution.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

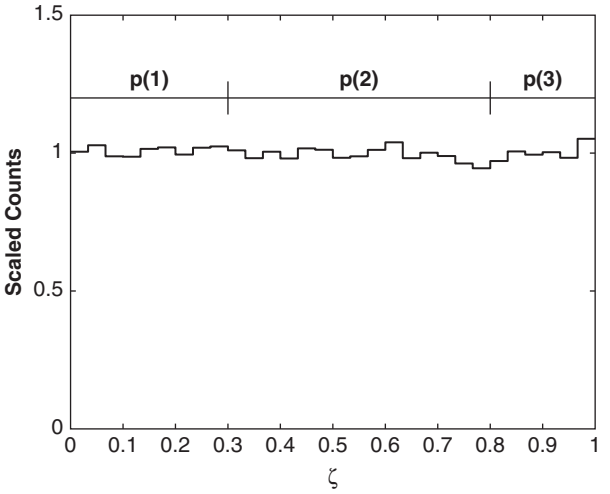

Figure 2.1 Schematic histogram of a uniform distribution returned by a random number generator. The generator was sampled 50,000 times and the results accumulated in 30 bins. The number of counts in a bin was divided by $50,000/30$, the expected number of counts per bin. Segments of a discrete probability are overlaid on the uniform distribution.

Direct sampling and *Markov chain methods* are two classes of sampling methods that are relevant for quantum Monte Carlo algorithms. We now discuss for each class various techniques for sampling discrete and continuous distribution functions.

2.3 Direct sampling methods

2.3.1 Discrete distributions

The simplest case of a direct sampling method is the standard procedure for sampling from a discrete probability function f_i for N events. To motivate it, we assume that $N = 3$ and that $f_1 = 0.3, f_2 = 0.5$, and $f_3 = 0.2$. Next, we lay out these probabilities over the uniform distribution as shown in Fig. 2.1. They segment the $[0, 1]$ interval into three regions. We see from this figure that f_1 overlays approximately 30% of the events generated by the random number generator, f_2 50%, and f_3 the remaining 20%. Thus, if we draw a uniformly distributed random number ζ from our generator, we observe that it properly samples event 1 if it is less or equal to f_1, samples event 2 if it is greater than f_1 but less than or equal to $f_1 + f_2$, and samples event 3 if it is greater than $f_1 + f_2$ but less than or equal to $f_1 + f_2 + f_3 = 1$. This observation translates into a simple strategy for sampling from a discrete distribution by using its cumulative probability function F_i. We detail this strategy in Algorithm 1.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

Algorithm 1 Sample a discrete distribution function via its cumulative distribution.

Input: Vector f of probabilities.

$F(0) \leftarrow 0$;
for $i = 1$ to N **do**
  $F(i) \leftarrow f(i) + F(i-1)$; ▷ Create cumulative distribution function
end for
Generate a uniform random number $\zeta \in [0, 1]$;
$k \leftarrow 0$;
repeat
  $k \leftarrow k + 1$;
until $\zeta \leq F(k)$.
return k.

More formally, for a uniform random variable we have $u(x) = f_X(x) = 1$ when $0 \leq x \leq 1$, so its cumulative distribution function is $F_X(x) = x$. We also have

$$P\left(0 \leq x_1 < \zeta \leq x_2 \leq 1\right) = F_X\left(x_2\right) - F_X\left(x_1\right) = x_2 - x_1,$$

which says that the probability that ζ lies in an interval $[x_1, x_2]$ of $[0, 1]$ is proportional to the length of the interval. Now if we have a set of discrete events with probability f_i, $i = 1, \ldots, n$, and we wish to sample one at random, we can divide $[0, 1]$ into segments of length f_i. The interval in which ζ lies then selects the event. We can accomplish the selection of event k by finding the k that satisfies

$$k = \min_n \left\{ \sum_{i=1}^{n} f_i \geq \zeta \right\}. \tag{2.10}$$

If the discrete probability has just a few elements, a simple linear search algorithm, as described in Algorithm 1, suffices. For a larger number of elements, a binary search becomes important for efficiency. If the vector of probabilities changes from sampling to sampling, Algorithm 1 is easily modified to eliminate the need to create and store the cumulative distribution; see Algorithm 2. If the vector of probabilities is very large and changes infrequently, if at all, other sampling algorithms such as the cut and the alias methods might be preferred (see Appendix A for Walker's alias method).

Algorithm 1 provides a way to sample an Ising configuration C_i from the Boltzmann probability $p(C)$ (1.8). After the $p(C_i)$ are constructed, we can use (2.10) to sample the C_i. As we already noted (Section 1.3), the construction of all the $p(C_i)$ becomes impractical when the lattice size becomes large. Sampling configurations from Boltzmann densities for large lattices is usually done efficiently by means of a Markov chain, a method we discuss in the next section.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

Algorithm 2 Sample a discrete distribution function.

Input: Vector f of probabilities.

Generate a uniform random number $\zeta \in [0, 1]$;
$k \leftarrow 0$;
repeat
  $k \leftarrow k + 1$;
  $\zeta \leftarrow \zeta - f(k)$;
until $\zeta < 0$
return k.

Some physical problems have very specific discrete distributions, such as the Poisson and binomial distributions, that require sampling. The cumulative probability method (2.10) can be used to sample from them. For these, and many other discrete distributions, exploiting specifics is often more efficient (Everett and Cashwell, 1983; Fishman, 1996). For example, while we could sample the Poisson probability (Fig. 2.2),

$$f_i = \frac{t^i}{i!} e^{-t}, \quad i = 0, 1, 2, \ldots \text{ and } t > 0 \tag{2.11}$$

via

$$k = \min_n \left\{ \sum_{i=0}^{n} \frac{t^i}{i!} \geq e^t \zeta \right\}, \tag{2.12}$$

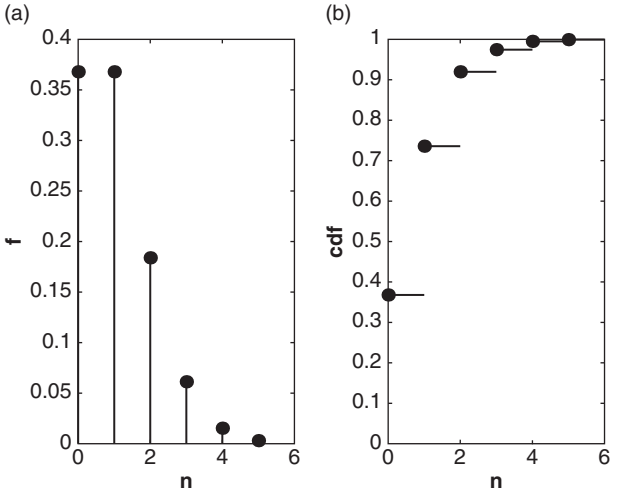

Figure 2.2 The (a) discrete and (b) cumulative probabilities for the Poisson distribution (2.11). Here $t = 1$.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

a more efficient way is

$$k = -1 + \min_n \{\zeta_1 \zeta_2 \cdots \zeta_n < e^{-t}\}, \tag{2.13}$$

where the ζ_i are successive draws from the random number generator. We leave the derivation of the latter procedure to the Exercises. It is also straightforward to show that the Poisson distribution (2.11) has a mean equal to t and a variance equal to t.

Equation (2.13) defines a way to determine the integer k. This integer, however, is a function of t. In many applications of a Poisson distribution, t defines the length of some interval, for example, $(0, t)$, and what is of interest is not only the number of events $k(t)$ occurring in this interval but also an ascending sequence of points $0 < t_1 < t_2 \cdots < t_k < t$ associated with these events. A stochastic process generating these $k(t)$ points is called a Poisson process. We discuss this process in Section 5.2.3.

2.3.2 Continuous distributions

As we shift from sampling discrete distributions to continuous ones, we also shift our attention from probability functions to probability densities $f(x)$. We can sample $f(x)$ by analogy with the procedure we described for a discrete distribution. After drawing our random number, we solve

$$\zeta = \int_{-\infty}^{x} f(y)dy. \tag{2.14}$$

for x. The content of this expression is $f(x)dx = d\zeta$; that is, the probability of x on $[x, x+dx]$ corresponds to the probability of the random number on $[\zeta, \zeta + d\zeta]$. This is the fundamental sampling concept for a continuous distribution.

Intuition likely suffices to justify the correctness of this procedure and the one for the discrete distribution. Mathematically, for both, we need to start with the observation that a function of a random variable is another random variable and the range of a cumulative distribution is $[0, 1]$, the same as the domain of our uniform random variable ζ. Next, we consider the properties of the generalized inverse of a nondecreasing function $F(x)$, for example, a cumulative distribution.

A *generalized inverse* (Fig. 2.3) of a nondecreasing function is defined as

$$F^-(y) = \inf \{x | F(x) \geq y\}.$$

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

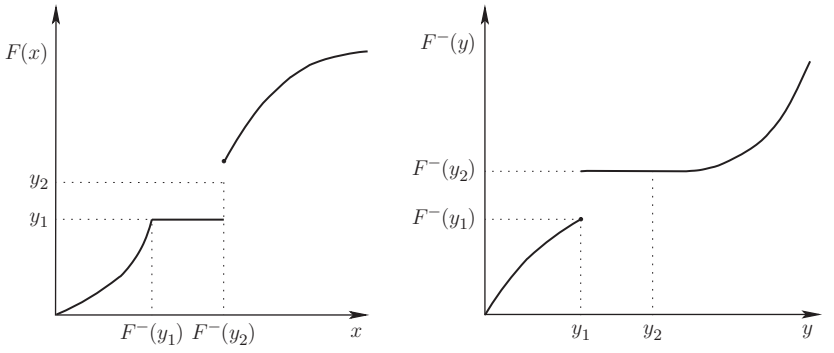

Figure 2.3 Graphical representation of the definition of a generalized inverse.

In general this inverse satisfies

$$F(F^{-}(y)) \geq y \quad \text{and} \quad F^{-}(F(x)) \leq x.$$

Therefore, the set $\{(x, y)\}$ of points for which $F^{-}(y) \leq x$ is the same as the set for which $F(x) \geq y$. Equating y with ζ and $F(x)$ with the cumulative distribution of the random variable X, we note that if y is in $[0, 1]$, then x is in $[F^{-}(0), F^{-}(1)]$. Then using the definition of a cumulative distribution (2.1), we can state

$$P(F^{-}(\zeta) \leq x) = P(\zeta \leq F(x)) = F(x).$$

Thus, to generate a random variable X that has a distribution $F(X)$ it is sufficient to generate a uniform random variable ζ and then make the transformation $x = F^{-}(\zeta)$.

As it stands, (2.14) is seldom easy to solve for x. If $F(x)$ is an explicit functional representation of the integral of $f(x)$, the problem, as just argued, is reduced to solving $\zeta = F(x)$ for x. This too can be difficult, but if the analytic inverse of $F(x)$ is known, then

$$x = F^{-1}(\zeta). \tag{2.15}$$

The simplest application of (2.14) is sampling from a uniform density over the interval $[a, b]$

$$\zeta = \int_{a}^{x} \frac{1}{b-a} dx = \frac{x-a}{b-a}.$$

Solving for x, we find

$$x = a + (b-a)\,\zeta,$$

a result we could easily have guessed.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

A more useful application of (2.15) is sampling from the exponential density defined over the interval $[0, \infty)$. We can integrate it to obtain

$$\zeta = \int_0^x \lambda e^{-\lambda y} dy = 1 - e^{-\lambda x},$$

and then solve for x,

$$x = -\ln(1-\zeta)/\lambda.$$

If ζ is a random variable on $(0, 1)$, then so is $1-\zeta$. Thus, the result is $x = -\ln \zeta/\lambda$.

We can sometimes sample other densities by a *change of variables*. Let us use this technique and a trick to derive the *Box-Muller method* for sampling from a Gaussian distribution

$$N(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\tfrac{1}{2}x^2\right), \quad -\infty < x < \infty.$$

The trick is the observation that it is easier to sample from the product of two Gaussian distributions

$$N(x_1)N(x_2) = \frac{1}{2\pi} \exp\left[-\tfrac{1}{2}\left(x_1^2 + x_2^2\right)\right].$$

By transforming to polar coordinates

$$x_1 = r\cos\theta, \quad x_2 = r\sin\theta,$$

we can write

$$N(x_1)N(x_2)dx_1dx_2 = \left[\exp\left(-\tfrac{1}{2}r^2\right) r dr\right]\left[\tfrac{1}{2\pi}d\theta\right].$$

There are now two probability densities we need to sample, and hence we need two random numbers. For the r-dependent distribution we have

$$\zeta_1 = \int_0^r \exp\left(-\tfrac{1}{2}s^2\right) s\, ds = 1 - \exp\left(-\tfrac{1}{2}r^2\right),$$

from which it follows that $r = [-2\log\zeta_1]^{1/2}$. For the θ-distribution we need to sample a uniform distribution over $[0, 2\pi]$ yielding

$$\theta = 2\pi\zeta_2.$$

From the two random numbers, two independent Gaussian random variables result:

$$x_1 = \left[-2\log\zeta_1\right]^{1/2}\cos(2\pi\zeta_2), \quad x_2 = \left[-2\log\zeta_1\right]^{1/2}\sin(2\pi\zeta_2). \tag{2.16}$$

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

In Appendix B we discuss the *rejection method*, a general-purpose technique for sampling from continuous distributions of a few random variables. An important feature of this technique is that the distribution being sampled need not be normalized. In the following section, we begin our discussion of Markov chains and common methods for generating them. Using these chains is almost the universal way we sample from discrete and continuous distributions of very high dimension.

2.4 Markov chain Monte Carlo

Before we define a Markov chain, we first make some notational adjustments and several remarks. In our discussion of Markov chains, and our subsequent discussion of Monte Carlo methods more generally, our random variable X is best regarded as a "random vector" $X = (Y_1, Y_2, \ldots, Y_n, \ldots)$ whose components Y_i are random variables. This understanding simplifies the notation for defining procedures that sample multivariate distributions by eliminating long strings of subscripts. The Y_i are independent random variables in the sense that we can vary the values of each irrespective of the values of the others. As the Y_i vary, they cause X to vary over the sample space of the problem. The Y_i the Markov chain generates, however, are not necessarily statistically independent from one sequence to another.

Let us recall our discussion of the Ising model in Chapter 1. There, we considered each Ising spin as a random variable that takes a value of plus or minus one. A *configuration* or a *state* corresponds to one outcome $(s_1, s_2, \ldots, s_N)$ in the sample space of the 2^N outcomes. The energy $E(C)$ of the Ising model is a random variable mapping a configuration to a real number. In this example, the random variable X represents a configuration C, and Y_i a single spin such as s_i. It is convenient to express the energy (and other observables) in terms of the individual values of the Ising variables, and thus to regard C as the collection $(s_1, s_2, \ldots, s_N)$. Furthermore, in a Markov chain, the transition from one configuration to the next is most easily implemented by changing one component of the configuration at a time.

The definition and properties of a Markov chain apply to both classical and quantum Monte Carlo algorithms. In a quantum Monte Carlo simulation, we may have to distinguish between the state of the system and the Monte Carlo configuration. In quantum mechanics, we work with the Hamiltonian operator H and the state vector $|\psi\rangle$. To deal with them numerically, we refer to some basis. Given some set $\{|C\rangle\}$ of "configuration states," we can evaluate matrix elements $\langle C'|H|C\rangle$ and wave functions $\psi(C) = \langle C|\psi\rangle$, which are classical objects and therefore amenable to the type of sampling we next discuss.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

In previous sections, we distinguished between random variables that were discrete (having a finite number of values) and continuous (having an infinite number of values). To continue doing so becomes cumbersome. In subsequent sections, the notation is most proper for discrete random variables, but in most cases it immediately generalizes to the continuous case. A few results we quote or prove are, however, most obviously correct only for finite, discrete-time Markov chains.

2.4.1 Markov chains

A *Markov chain* is a procedure to generate a sequence $x_1, x_2, \ldots, x_j, \ldots$ of the values of a random variable. Such a sequence is said to be Markovian if all the conditional probabilities for the associated random variables X_j satisfy

$$P(X_j|X_{j-1}, \ldots, X_1) = P(X_j|X_{j-1}),$$

that is, the probability of X_j depends only on the random variable X_{j-1} immediately preceding it and not on the others. The order of the random variables is important.

For simplicity we consider only discrete sequences of random variables with a finite number of values. Let us define $P_{ij} = P(X_i|X_j)$ and $p_i = P(X_i)$. A Markov chain is defined by an initial probability p_i^0 and a *transition probability* matrix P_{ij} normalized for each j by $\sum_i P_{ij} = 1$. The chain is typically generated recursively,

$$p_i^{(k+1)} = \sum_j P_{ij} p_j^{(k)}, \quad k = 0, 1, \ldots \tag{2.17}$$

with $p_i^{(0)} = p_i^0$. If we sum both sides of this equation over i and use the normalizations $\sum_i P_{ij} = 1$ and $\sum_i p_i^0 = 1$, we find that $\sum_i p_i^{(k)} = 1$ so that the iteration conserves probability.

What is remarkable about (2.17) is that under very general conditions on the P_{ij}, and independent of the starting point p_i^0, the iteration eventually reaches a stationary state p_i that satisfies the *stationary condition*

$$p_i = \sum_j P_{ij} p_j, \tag{2.18}$$

that is, it produces the eigenvector p_i of the matrix P_{ij} whose eigenvalue is 1 and whose vector components satisfy $\sum_i p_i = 1$.

In general, p_i is unknown. As we will see, in ground state quantum Monte Carlo simulations, finding p_i is the objective. In other applications, such as an Ising model simulation, producing a specific distribution is necessary. To do so requires specific choices of the transition probability. Metropolis et al. (1953) proposed that we will obtain a specific distribution p_i if the transition probability obeys

$$P_{ij}p_j = P_{ji}p_i. \tag{2.19}$$

This is the *detailed balance condition*, which equates the probability of being in state j and going to i with the probability of being in i and going to j.[3] If the right-hand side of (2.19) is substituted into the right-hand side of (2.18), then $p_i = \sum_j P_{ji}p_i = p_i$, where we used the normalization $\sum_j P_{ji} = 1$. Hence, P_{ij} and p_i are consistent in the stationary state. As we soon discuss, Metropolis et al. also gave a general definition for such a P_{ij} and an algorithm for sampling from it. The algorithm is called the *Metropolis algorithm* and has the remarkable property that the function to be sampled need not be normalized.

2.4.2 Stochastic matrices

We are now poised to define Monte Carlo algorithms that allow us to answer the two basic questions asked of a system of interacting particles, namely, what are its ground state properties? And what are its finite-temperature properties? To find the ground state, we need to construct a Markov chain whose transition probability projects to it. It is this type of Monte Carlo calculation that Fermi was suggesting (Chapter 1). To compute thermodynamic quantities we need a transition probability that satisfies detailed balance for the Boltzmann distribution. We defer the discussion of the ground state algorithms to Chapters 9, 10, and 11. Before we start discussing several detailed balance algorithms for equilibrium thermodynamics, we first justify the remarkable properties of a Markov chain and discuss the general conditions for their validity.

The property of a Markov chain that a unique stationary distribution satisfying (2.18) always exists follows from the transition probability P_{ij} being an ergodic stochastic matrix. A *stochastic matrix* is a matrix with no negative elements (a non-negative matrix) and each column sum equal to 1, that is, $\sum_i P_{ij} = 1$.[4] *Irreducibility* means that a series of permutations of rows and columns *must not* transform P_{ij} into the form

$$\begin{pmatrix} A & B \\ 0 & C \end{pmatrix},$$

where A and C are square matrices of any possible order. If it does, then the matrix is called *reducible*. Similarly, *aperiodicity* means that there is no permutation that transforms P_{ij} into the form

[3] Equation (2.19) is a sufficient but not a necessary condition for stationarity (see also the discussion in Sec. 2.6).

[4] A stochastic matrix is often defined as having the row (instead of the column) sums equal to 1. We choose the column-wise definition because it is consistent with the natural order of matrix-vector multiplication.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

$$\begin{pmatrix} 0 & A_1 & 0 & \cdots & 0 \\ \vdots & 0 & A_2 & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & A_{n-1} \\ A_n & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

An irreducible and aperiodic stochastic matrix is called an ergodic matrix.[5]

In general, a stochastic matrix is not symmetric; that is, $P_{ij} \neq P_{ji}$. Consequently, it has unequal left (x^α) and right (y^α) eigenvectors that share the same eigenvalues

$$\sum_i x_i^\alpha P_{ij} = \lambda_\alpha x_j^\alpha, \quad \sum_j P_{ij} y_j^\alpha = \lambda_\alpha y_i^\alpha.$$

If the eigenvalues are distinct, $\lambda_\alpha \neq \lambda_\beta$, then the right and left eigenvectors, x^α and y^β, are linearly independent and satisfy $[x^\alpha]^T \cdot y^\beta = 0$.

While the proof lies outside the scope of this book, one can show that an irreducible, stochastic matrix has a nondegenerate eigenvalue equal to 1 and the corresponding right eigenvector's components are all positive (Meyer, 2000, Chapter 8). These results follow from the application of the *Perron-Frobenius theorem*.[6] This theorem says that if an irreducible matrix (not necessarily stochastic) is non negative, then it has a nondegenerate eigenvalue equal to its spectral radius.[7] Accordingly, this eigenvalue is real and positive. The theorem states further that the components of the associated right eigenvector are also positive, and this eigenvector is unique (up to an overall scaling). If the matrix is stochastic, then we can show that the absolute value of the eigenvalue cannot exceed unity, and therefore at least one of the dominating eigenvalues is unity, and other dominating eigenvalues, if any, are complex numbers with absolute value 1.

To further nail down the dominating eigenvalue, we must require an additional condition, namely, the aperiodicity. We can show that, if the matrix P is aperiodic as well as irreducible, there exists a finite number n of matrix multiplications so that P^n is a positive matrix; that is, $[P^n]_{ij} > 0$ for any pair of i and j. When a non-negative matrix satisfies this condition, it is called *primitive*, and the sampling is called *ergodic*.

[5] In general, an irreducible and aperiodic non negative matrix is called a *primitive* matrix. An ergodic matrix is a primitive stochastic matrix.

[6] The analogous theorem for continuous operators is called the Jentzsch-Hopf theorem (Camp and Fisher, 1972; van Hove, 1950).

[7] The spectral radius is $\rho(A) = \max |\lambda|$, where λ is a member of the set of eigenvalues of A.

If a stochastic matrix is positive, then from any given state the chain can transition to any other state in a single step. Note that an irreducible, stochastic matrix generally has zero elements (and may have more than one eigenvalue of unit magnitude). This means that some states are inaccessible from a given state by one step in the Markov chain.[8] Further, one can show (Meyer, 2000, Chapter 8) that for a positive stochastic matrix, there is only one eigenvector with an eigenvalue equal to 1. It is straightforward to show that these properties also apply to the primitive stochastic matrix. Namely, if a stochastic matrix is primitive, the right eigenvector of P with the eigenvalue of 1 is unique and positive. In addition, when P is chosen so that it satisfies the stationary condition (2.18) with a given target distribution p, the right dominating eigenvector equals p. It means that after sufficiently many iterations in the Markov-chain Monte Carlo simulation, the probability distribution converges to the target distribution.

Now, let us take some (unnormalized) vector ψ and express it as a linear combination of the right eigenvectors y^α, that is, $\psi = \sum_\alpha a_\alpha y^\alpha$, with $a_1 = 1$, and order the eigenvalues as $1 > |\lambda_2| \geq |\lambda_3| \geq \ldots$ Repeated multiplication of the matrix P with this vector yields

$$P^k \psi = y^1 + \sum_{\alpha \geq 2} \lambda_\alpha^k y^\alpha .$$

As k becomes large, the eigenvectors with subdominant eigenvalues project out, leaving just the right eigenvector of the nondegenerate unit eigenvalue. Setting $y_i = p_i > 0$ leads us to (2.18). We note that this projected state does not depend on the starting state p_i^0. This exercise reveals another important feature of Markov chain Monte Carlo: *before we start taking samples, we have to iterate for a while to "equilibrate" the system.* This contrasts with the direct sampling methods previously considered.

While an algorithm might be formally ergodic, in an actual simulation it might behave as if it is not. For example, there might exist several high probability regions of phase space connected by low probability paths. Moving between these regions of high probability is a rare event, costly in computation time. As a result, the simulation might lock into one of the regions, implying false convergence and producing configurations uncharacteristic of the entire phase space. In such a situation, a proper sampling requires long simulation times, and the data analysis (Chapter 3) needs special care to assure statistical independence among measurements.

[8] An irreducible matrix must have at least one nonzero off-diagonal element in each column. In general, the product of a stochastic matrix with itself may also have zero elements, again leaving some states inaccessible.

2.5 Detailed balance algorithms

We now discuss two classes of algorithms that satisfy detailed balance and hence allow the sampling from a prespecified probability distribution. The first class encompasses the Metropolis and related algorithms; the second class, the *heat-bath algorithm* and related algorithms. For both classes, the new state is usually sampled by means of relatively local configuration changes. In contrast to the Metropolis algorithm, the heat-bath algorithm has no explicit rejection. In the broader literature, this algorithm is called the *Gibbs sampler*, and in the statistics literature in particular, a litany of Gibbs samplers exists. Although the two classes employ different methodologies, the heat-bath algorithm is actually a special case of what is called the *Metropolis-Hastings algorithm*.

2.5.1 Metropolis algorithm

The Metropolis algorithm (Metropolis et al., 1953) is ingenious with an elegance that derives from its simplicity. It was proposed more as a method that "seems reasonable" than one with any obviously intended connections to Markov chain mathematics. The inventors did recognize it as a computational tool that in principle could solve any problem in classical equilibrium statistical mechanics. We quote from the introductory paragraph of Metropolis et al. (1953):

> The purpose of this paper is to describe a general method, suitable for fast electronic computing machines, of calculating the properties of any substance which may be considered as composed of interacting individual molecules. Classical statistics is assumed ...

Indeed, the Metropolis algorithm has endured for decades, spreading well beyond statistical mechanics into all areas of analysis using stochastic processes. We now present the algorithm and show that it defines a stochastic transition matrix that satisfies the detailed balance condition. We then explain the Metropolis et al. procedure for sampling from this matrix and discuss its use in statistical mechanics.

To sample a probability distribution p_i asymptotically, Metropolis et al. proposed the following form of the transition probability matrix

$$P_{ij} = T_{ij}A_{ij}, \tag{2.20}$$

where the T_{ij} are the elements of a symmetric matrix of *trial transition (proposal) probabilities* satisfying

$$T_{ij} \geq 0, \quad T_{ij} = T_{ji}, \quad \text{and} \quad \sum_i T_{ij} = 1, \tag{2.21}$$

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

and the A_{ij} are elements of an *acceptance matrix*,

$$A_{ij} = \begin{cases} 1 & \text{if } p_i/p_j \geq 1, \\ p_i/p_j & \text{if } p_i/p_j < 1, \end{cases} \tag{2.22}$$

often written as

$$A_{ij} = \min\left\{1, p_i/p_j\right\}. \tag{2.23}$$

Strictly speaking the definitions (2.22) and (2.23) are valid only for $i \neq j$. This condition is typically an ingrained part of the algorithm's implementation: what is proposed, by using T_{ij}, is by design a change, so $i \neq j$. However, for the purpose of demonstrating that the transition probability matrix defined by the algorithm is a stochastic matrix, let us state the transition probability matrix as

$$P_{ij} = \begin{cases} \begin{cases} T_{ij} & \text{if } p_i/p_j \geq 1 \\ T_{ij}p_i/p_j & \text{if } p_i/p_j < 1 \end{cases} & i \neq j, \\ T_{jj} + \sum\limits_{\{k|p_k<p_j\}} T_{kj}\left(1 - p_k/p_j\right) & i = j. \end{cases}$$

The expression for P_{jj} defines the rejection component of the Metropolis algorithm. If the proposed move is rejected, then the current state is added to the Markov chain.

With the transition probability matrix now completely defined, its stochastic nature is easily demonstrated.[9] We need only to show that $\sum_i P_{ij} = 1$, because it is obvious from its definition that $P_{ij} \geq 0$. The steps of the proof are

$$\begin{aligned} \sum_i P_{ij} &= P_{jj} + \sum_{\{i|p_i>p_j\}} T_{ij} + \sum_{\{i|p_i<p_j\}} T_{ij}p_i/p_j \\ &= T_{jj} + \sum_{\{k|p_k<p_j\}} T_{kj}\left(1 - p_k/p_j\right) + \sum_{\{i|p_i>p_j\}} T_{ij} + \sum_{\{i|p_i<p_j\}} T_{ij}p_i/p_j \\ &= T_{jj} + \sum_{i\neq j} T_{ij} \\ &= 1. \end{aligned}$$

To show detailed balance (2.19), we first note that it is obvious for $i = j$. Then, for $i \neq j$, by assuming $p_i < p_j$ without loss of generality (due to the arbitrariness of labeling),

$$P_{ij} = T_{ij}p_i/p_j = T_{ji}p_i/p_j = P_{ji}p_i/p_j,$$

from which it follows that $P_{ij}p_j = P_{ji}p_i$. In obtaining this result, we used the symmetry of T, (2.20), and (2.23). Thus, the demonstration that detailed balance holds for all i and j is complete.

[9] The irreducibility of the matrix follows from nonzero diagonal elements.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

Algorithm 3 Metropolis algorithm.

Input: Proposal probability T, limiting distribution p, configuration j.
  Sample an i from T_{ij} ;
  Generate a uniform random number $\zeta \in [0, 1]$;
  if $\zeta \leq p_i/p_j$ **then**
    $j \leftarrow i$;
  end if
  return j.

To sample an i given a j is exceptionally simple (see Algorithm 3), but possibly not obvious in afterthought. In this procedure, Metropolis et al. use T_{ij} to create a proposal for the next configuration i in the Markov chain. This new configuration could in principle be any allowable one. Once it is proposed, they then use the A_{ij} element of the acceptance matrix to decide between i and j.

What does this sampling procedure mean in practice? From (2.23) we see that the transition probabilities depend only on the ratio of the weight of the proposed configuration to the current one and hence are independent of their normalization constant. In classical statistical mechanics, this ratio is a ratio of Boltzmann factors and equals $\exp(-\Delta E/kT)$ where ΔE is the energy difference between the two configurations. The Markov chain is constructed by visiting each lattice site or particle, deterministically or stochastically, and proposing a new configuration by changing the state of the random variable on the site or moving the particle. The changes are typically local. Rarely does a Metropolis Monte Carlo simulation change all, or even more than just a few, of the variables defining a configuration at once. The algorithm says "accept the proposed local change" if the new configuration has a lower energy. The algorithm also says "sometimes accept a change that increases the energy, but only with probability $\exp(-\Delta E/kT)$."

For the Ising model, a typical T_{ij} selects a specific lattice site and proposes a change in the spin state. The random variable at this site, that is, the Ising spin, has two possible values, the present one and the flipped one. The Metropolis transition probability selects which of the two configurations is added to the Markov chain. *It is important to note that the Metropolis algorithm makes repeating the configuration in the Markov chain one or more times a definite and necessary possibility.*

Is the Metropolis algorithm ergodic? The Metropolis algorithm transfers the burden of ergodicity to T_{ij}. If in a finite number of steps T_{ij} enables all of the phase space to be reached, then the Metropolis algorithm is ergodic. For the Ising example, T_{ij} allows each site to be visited, and at each visit any value of the random variable can be selected. Thus, all spin configurations may be realized.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

2.5.2 Generalized Metropolis algorithms

Generalizations of the Metropolis algorithm exist. Let us view them in terms of the general prescription that

$$P_{ij} = T_{ij}A_{ij,} \quad \text{and} \quad A_{ij} = \min\left\{1, \mathcal{R}_{ij}\right\}. \tag{2.24}$$

Here we no longer require that $T_{ij} = T_{ji}$. First we note that there is a general class of algorithms associated with the acceptance ratio

$$\mathcal{R}_{ij} = \frac{S_{ij}}{1 + \dfrac{T_{ij}p_j}{T_{ji}p_i}}, \tag{2.25}$$

where S_{ij} is any non negative symmetric matrix[10] that ensures $0 \leq A_{ij} \leq 1$ for all i and j. Two choices for S_{ij} are common. One is

$$S_{ij} = 1, \tag{2.26}$$

leading to

$$\mathcal{R}_{ij} = \frac{T_{ji}p_i}{T_{ij}p_j + T_{ji}p_i}, \tag{2.27}$$

which is called the Metropolis-Barker algorithm (Barker, 1965). The second, due to Hastings (1970), is

$$S_{ij} = \begin{cases} 1 + \dfrac{T_{ji}p_i}{T_{ij}p_j}, & \text{if} \quad T_{ij}p_j \geq T_{ji}p_i\,, \\ 1 + \dfrac{T_{ij}p_j}{T_{ji}p_i}, & \text{if} \quad T_{ji}p_i > T_{ij}p_j\,, \end{cases} \tag{2.28}$$

leading to the Metropolis-Hastings algorithm based on

$$\mathcal{R}_{ij} = \frac{T_{ji}p_i}{T_{ij}p_j}. \tag{2.29}$$

If T is symmetric, the Metropolis-Hastings algorithm reduces to the original Metropolis algorithm. We remark that Barker's and Hastings's algorithms share with the original Metropolis algorithm the property of not needing the normalization of the stationary distribution. We leave it to the Exercises to show that all three algorithms satisfy the detailed balance condition.

Not all algorithms that satisfy detailed balance are of the Metropolis form (2.24). Another general class has an acceptance matrix of the form $A_{ij} = S_{ij}/p_jT_{ij}$. Again, S_{ij} is any non negative symmetric matrix that ensures $0 \leq A_{ij} \leq 1$ for all i and j.

[10] A non negative matrix has all elements greater than or equal to zero.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

In this case the transition probability of the Markov chain becomes independent of the proposal probability, but dependent on the normalization of the stationary distribution. Many more generalizations of the original Metropolis et al. insights exist. The challenge is not in creating them but in deciding which is the most efficient. Fortunately, there are a few guidelines worth noting.

First we comment on the difference between computational efficiency and statistical efficiency. *Computational efficiency* is measured by the average computer time per step needed to execute the algorithm. Here, we address *statistical efficiency*. From the discussion in Section 2.4.2 we saw that the statistical efficiency, in the sense of the convergence of the Markov chain, is controlled by the magnitude of the second-largest eigenvalue of the transition probability matrix: the smaller the better. Because of the size of the matrix,[11] quantitative knowledge of this eigenvalue is typically unavailable. Recently, Monte Carlo methods were developed to determine this eigenvalue (Rubenstein et al., 2010). The available results support the analytic work by Peskun (1973).

The mathematical analysis of Peskun, and subsequently by others (Liu, 2001) established a theorem that if the off-diagonal elements of one transition matrix are greater than or equal to those of another, the variances of any expectation value produced by the Markov chain generated with the off-diagonally dominant transition matrix are smaller than those produced by the other chain. The general rule proposed by Peskun is that statistical efficiency increases as the off-diagonal matrix elements of P_{ij} dominate the diagonal ones. This makes sense. Off-diagonal dominance decreases the probability of a state being repeated in the chain and thus reduces statistical correlations among successive estimates of an observable. For *discrete* spaces Peskun also proved that the Metropolis-Hastings algorithm is statistically more efficient than the Metropolis-Barker algorithm.

Very generally, the detailed balance condition and the Metropolis-Hastings algorithm provide flexible tools for designing algorithms to sample from a specified probability density. Various alternatives to this algorithm are possible and may control the probability flow more optimally. Dramatic increases in computational efficiency have occurred by designing algorithms that are *not* of the Metropolis (or heat-bath) type. The rejection character of the algorithm, repeating the current configuration one or more times, is the principal source of statistical inefficiency as it inhibits the generation of statistically independent measurements. Breaking away from the Metropolis algorithm is, however, not easy. Successful examples are the cluster and loop algorithms, which are discussed in Chapters 4, 5, and 6.

[11] The elements of the transition probability matrix are computed on the fly. Except for trivial problems or small physical systems, computer memory is insufficient to store the entire matrix.

The important innovation in these algorithms is the step from local configuration changes to global ones.

2.5.3 Heat-bath algorithm

The *heat-bath algorithm* (Creutz, 1980) involves local moves, as does the Metropolis algorithm, but in contrast to the Metropolis algorithm, the proposed change in the value of the local variable x_i is independent of its current value and is always accepted. It may get repeated in the Markov sequence many times, but this is not the result of a rejection – rather, it is the result of a selection.

The strategy is to sample x_i from a univariate density that is conditional on the values of a small number of variables. In other words, we sample x_i from some $P(x_i|x_{i_1}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i_m})$ where $m \ll N$. If the local environment of x_i were replaced with a heat bath, defined by the conditional variables, then we would be sampling from the expected equilibrium distribution for that environment.

We can achieve the same effect with the Metropolis algorithm by using the standard proposal probability and repeatedly updating the configuration at the same site. Eventually, we start sampling from the local equilibrium probability. Thus, the heat-bath algorithm achieves in one step something that might take the Metropolis algorithm many steps. Because of this, the heat-bath algorithm is generally believed to be more efficient than the Metropolis algorithm. However, the repetition of the state in the chain because of selection can have the same effect as the rejection in the Metropolis case. The situation with respect to which algorithm is more efficient is in reality more complex (Rubenstein et al., 2010).

To illustrate the algorithm, we again invoke the Ising model as the example. We select some lattice site i and then fix the values of the spins at the other sites. The local energy of the spin at the selected site is $E(s_i) = -Js_i \sum_{j \neq i} s_j$, leading to the local Boltzmann factor $\exp[-E(s_i)/kT]$. Taking into account the two spin orientations, we can write the local probability for the spin $s_i = \pm 1$ as

$$P(s_i) = \frac{\exp[-E(s_i)/kT]}{\exp[-E(s_i)/kT] + \exp[-E(-s_i)/kT]}. \tag{2.30}$$

Sampling from this probability places an s_i with a value of $+1$ or -1 at site i, thereby generating a new configuration from the old one without regard to the previous value of the spin at the selected site.

To present more details and justification, we proceed as follows. If the probability function for a collection of random variables is $P(s_1, s_2, \ldots, s_N)$, the heat-bath algorithm generates a new configuration from the old one by sampling the conditional probability for each variable while the others are fixed (see Algorithm 4).

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

Algorithm 4 Heat-bath algorithm.

Input: A configuration $(s_1, s_2, \ldots, s_N)$.
  for $i = 1$ to N **do**
    Sample s_i' from $P(s_i'|s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_N)$;
    $s_i \leftarrow s_i'$;
  end for
  return the updated $(s_1, s_2, \ldots, s_N)$.

The definition of a multivariate conditional probability

$$P(s_i|s_1, \ldots .s_{i-1}, s_{i+1}, \ldots, s_N) = \frac{P(s_1, \ldots, s_N)}{P(s_1, \ldots .s_{i-1}, s_{i+1}, \ldots, s_N)}, \tag{2.31}$$

plus that of the marginalization

$$P(s_1, \ldots .s_{i-1}, s_{i+1}, \ldots, s_N) = \sum_{s_i} P(s_1, \ldots, s_N),$$

give the probabilities we need. For the zero-field, one-dimensional Ising model with nearest neighbor interactions, we can express these probabilities in a simple formula. The conditional probability becomes

$$\begin{aligned} P(s_i|s_{i-1}, s_{i+1}) &= \frac{e^{J(s_{i-1}s_i + s_i s_{i+1})/kT}}{\sum\limits_{\tilde{s}_i = \pm 1} e^{J(s_{i-1}\tilde{s}_i + \tilde{s}_i s_{i+1})/kT}} \\ &= \frac{e^{J(s_{i-1}+s_{i+1})s_i/kT}}{e^{J(s_{i-1}+s_{i+1})/kT} + e^{-J(s_{i-1}+s_{i+1})/kT}}, \end{aligned}$$

which is a more detailed expression of (2.30). Note that we started with all the random variables but finished with an expression dependent on only a few. The reduction is a consequence of the Ising model's short-ranged exchange interaction.

The heat-bath algorithm is a special case of the Metropolis-Hastings algorithm (2.28), with an acceptance probability equal to one. To see this, let us at the i-th step of the heat-bath algorithm form the Metropolis-Hastings acceptance ratio

$$\begin{aligned} \mathcal{R}_{s_i', s_i} &= \frac{P(s_1, \ldots, s_i', \ldots s_N) P(s_i|s_1, \ldots, s_{i-1}, s_{i+1} \ldots s_N)}{P(s_1, \ldots, s_i, \ldots s_N) P(s_i'|s_1, \ldots, s_{i-1}, s_{i+1} \ldots s_N)} \\ &= \frac{P(s_i'|s_1, \ldots, s_{i-1}, s_{i+1} \ldots s_N) P(s_i|s_1, \ldots, s_{i-1}, s_{i+1} \ldots s_N)}{P(s_i|s_1, \ldots, s_{i-1}, s_{i+1} \ldots s_N) P(s_i'|s_1, \ldots, s_{i-1}, s_{i+1} \ldots s_N)} = 1, \end{aligned}$$

where we invoked (2.31) twice, once for $P(s_1, \ldots, s_i', \ldots s_N)$ and the second time for $P(s_1, \ldots, s_i, \ldots s_N)$. Thus each step of the heat-bath algorithm corresponds to a Metropolis-Hastings sampling with $A_{ij} = 1$.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

2.6 Rosenbluth's theorem

As previously noted, detailed balance and the Metropolis algorithm were originally proposed more as something that seemed reasonable than as something that followed from the mathematics. About four years after the Metropolis et al. publication, Wood and Parker (1957) connected the algorithm with a Markov process, and this often repeated but seldom cited analysis defines the standard justification of the original proposal. Shortly after the Metropolis et al. paper, Rosenbluth (1953) wrote an unpublished, and only recently noticed, report (Gubernatis, 2003) proving the validity of the algorithm. His proof focuses on the conservation of probability. In fact, he proved that under the conditions of ergodicity and detailed balance, probability flows through phase space in such a way that the average squared-deviation of the probability density from the canonical ensemble not only becomes zero but does so monotonically. His insightful proof points to a special character of the convergence, absent from the standard Markov chain proof.

Let us offer here a related but simpler proof for the convergence of the Markov chain and the H-theorem. While to prove them in a general setting we need the Perron-Frobenius theorem, which itself requires many lines to prove, the assumption that the stationary condition (2.18) is satisfied with the target distribution simplifies the task. The fact that most of the Monte Carlo algorithms satisfy an even stronger condition, namely, the detailed-balance condition, justifies this assumption. As we will see, we require only the stationary condition and ergodicity (Section 2.4.2) in proving the convergence of the Markov process. We need the detailed balance condition (2.19) in establishing the H-theorem, or generalized Rosenbluth's theorem.

We consider two ensembles $\mathcal{E}_p$ and $\mathcal{E}_q$ of configurations i with probability densities p_i and q_i. We define the distance between these ensembles as

$$\|\mathcal{E}_p - \mathcal{E}_q\| = \sum_i |p_i - q_i|.$$

Now we consider the case where p_i and p'_i are related by one step in the Markov chain $p'_i = \sum_j P_{ij} p_j$. We already have noted that this operation conserves probability if P is a stochastic matrix. The distance between the ensemble $\mathcal{E}'$ and the target ensemble $\mathcal{E}^{\text{eq}}$, which satisfies the stationarity condition, is

$$\begin{aligned} ||\mathcal{E}' - \mathcal{E}^{\text{eq}}|| &= \sum_i \left| \sum_j P_{ij} \left(p_j - p_j^{\text{eq}}\right) \right| \\ &\leq \sum_i \sum_j P_{ij} \left|p_j - p_j^{\text{eq}}\right| = \sum_j \left|p_j - p_j^{\text{eq}}\right|. \end{aligned} \tag{2.32}$$

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

Note that we have used the stationarity condition $\sum_j P_{ij} p_j^{\rm eq} = p_i^{\rm eq}$ in the first step and the conservation of probability $\sum_i P_{ij} = 1$ in the last step.[12] On the right-hand side, we recognize $\|\mathcal{E} - \mathcal{E}^{\rm eq}\|$, so

$$\|\mathcal{E}' - \mathcal{E}^{\rm eq}\| \le \|\mathcal{E} - \mathcal{E}^{\rm eq}\|.$$

Thus, each step of the algorithm reduces the distance between the current ensemble and the equilibrium ensemble. Because the distance is trivially bounded from below by zero, it must converge to some value. The remaining question is whether this value is zero or not. We can answer this question by examining (2.32) more closely.

First, we note that (2.32) must hold even if we replace P_{ij} in (2.32) by $[P^n]_{ij}$ and let $\mathcal{E}'$ be the ensemble after n steps. Let n be large enough so that $[P^n]_{ij} > 0$ for all combinations of i and j. Choosing such an n is possible due to the ergodicity. Now, after convergence is reached, the equality must hold in (2.32). For it to hold, $p_j - p_j^{\rm eq}$ must have the same sign (or be zero) for all j reachable from the same i by n steps. Since $[P^n]_{ij} > 0$ for all j, it means $p_j \le p_j^{\rm eq}$ for all j or $p_j \ge p_j^{\rm eq}$ for all j. But because of the normalizations $\sum_j p_j = \sum_j p_j^{\rm eq} = 1$, each inequality must reduce to $p_j = p_j^{\rm eq}$. Thus, p_j must converge to $p_j^{\rm eq}$. This observation completes the proof of the theorem that an ergodic Markov process with a stationary condition (or detailed balance condition) must converge to the target distribution. It also follows that the eigenvalue with the largest magnitude is unity and that the corresponding eigenvector is unique and equal to $p_i^{\rm eq}$, because otherwise the process would not always converge to $p_i^{\rm eq}$ regardless of the initial distribution.

We can also show that if a Markov process satisfies detailed balance, the free energy decreases monotonically and eventually converges to its equilibrium value. We call this type of H-theorem for Monte Carlo simulations *Rosenbluth's theorem*. This monotonicity is a feature characteristic of a more general convex function than the averaged squared-deviation from the equilibrium density (Renyi, 1960; Kawashima, 2007). The assertion is that

$$p_i' = \sum_j P_{ij} p_j \tag{2.33}$$

together with the condition that $P_{ij} p_j^{\rm eq} = P_{ji} p_i^{\rm eq}$ causes

$$\Phi = \sum_i p_i^{\rm eq} f\left(\frac{p_i}{p_i^{\rm eq}}\right)$$

[12] The fact that we have not used the detailed balance condition is important for proving the convergence of Monte Carlo algorithms that do not satisfy the detailed balance condition, such as the directed-loop algorithm, although we do not discuss many such cases in the present book.

to converge monotonically if $f(x)$ is a convex function. The most important example of this family of observables can be obtained by setting $f(x) = x \ln x$. The result is the Kullback-Leibler information:

$$\Phi = I_{\mathrm{KL}}(p \,||\, p^{\mathrm{eq}}) \equiv \sum_i p_i \ln\left(\frac{p_i}{p_i^{\mathrm{eq}}}\right).$$

For physicists, a more familiar interpretation of this quantity is the excess free energy, which measures how much larger the current free energy is than its equilibrium value. Namely, when $p_i^{\mathrm{eq}} \propto e^{-E_i/kT}$, then $I_{\mathrm{KL}} = (F - F^{\mathrm{eq}})/(k_B T)$ with $F = \langle E \rangle - TS$ and $F^{\mathrm{eq}} = \langle E \rangle^{\mathrm{eq}} - TS^{\mathrm{eq}}$, where S is the von Neuman entropy, $S \equiv -\sum p \log p$. If we set $f(x) = x^n$, we obtain the n-th Renyi information:

$$\Phi = I_{\mathrm{R}}^{(n)}(p \,||\, p^{\mathrm{eq}}) \equiv \sum_i \frac{p_i^n}{(p_i^{\mathrm{eq}})^{n-1}} = \left\langle \left(\frac{p_i}{p_i^{\mathrm{eq}}}\right)^{n-1} \right\rangle.$$

The proof is as follows. At a particular step in the iteration, we have

$$\Phi' = \sum_i p_i^{\mathrm{eq}} f\left(\frac{p_i'}{p_i^{\mathrm{eq}}}\right).$$

Using (2.33), we rewrite this equation as

$$\Phi' = \sum_i p_i^{\mathrm{eq}} f\left(\frac{1}{p_i^{\mathrm{eq}}} \sum_j P_{ij} p_j\right) = \sum_i p_i^{\mathrm{eq}} f\left(\sum_j P_{ij} \frac{p_j^{\mathrm{eq}}}{p_i^{\mathrm{eq}}} \frac{p_j}{p_j^{\mathrm{eq}}}\right),$$

which on the use of the detailed balance condition becomes

$$\Phi' = \sum_i p_i^{\mathrm{eq}} f\left(\sum_j P_{ji} \frac{p_j}{p_j^{\mathrm{eq}}}\right).$$

Now we employ Jensen's inequality for the convex function f, that is, $f(\sum_i a_i x_i) \leq \sum_i a_i f(x_i)$ for any probability distribution a_i, to obtain

$$\Phi' \leq \sum_{ij} P_{ji} p_i^{\mathrm{eq}} f\left(\frac{p_j}{p_j^{\mathrm{eq}}}\right).$$

Using the detailed balance condition and $\sum_i P_{ij} = 1$, we find

$$\Phi' \leq \sum_j p_j^{\mathrm{eq}} f\left(\frac{p_j}{p_j^{\mathrm{eq}}}\right) = \Phi.$$

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

This quick derivation represents a proof quite different from the one presented by Rosenbluth. His proof was in the continuum and he argued on the basis of the need to conserve the number of particles $\rho(r)dr$ in a differential phase space volume dr as they move through phase space.

2.7 Entropy content

The *entropy* of a distribution,

$$S = -\sum_{i=1}^{n} p_i \ln p_i,$$

provides information about its global character. For example, if all the p_i are equal, the entropy for an n-state distribution takes its maximum value of $\ln n$, and we say that the distribution contains the minimal amount of information. In other words, one event is as likely as any other, and we have the maximum uncertainty about which event the chain would generate next. If the probability of a particular event is one, which means the probability of all others must be zero, then the value of the entropy takes its minimum value of zero. The event is a sure thing with no uncertainty. Once the Markov chain reaches stationarity, that is, samples a fixed distribution, its information entropy content, as defined by the distribution it generates, becomes constant. We refer to this constant as the entropy content of the Markov process. (In fact, the information that is actually minimized at the stationary point is the Kullback-Leibler information, which is proportional to the excess free energy; see Section 2.6.)

If the chain takes r steps beyond some point of stationarity, it generates a sequence of events $K = \{k_1, k_2, \ldots, k_r\}$ with probability $P(K) = P_{k_r,k_{r-1}} \cdots P_{k_3,k_2} P_{k_2,k_1} p_{k_1}$. Let us consider the set of all n^r sequences. Then, one can prove (Khinchin, 1957) that given $\varepsilon > 0$ and $\eta > 0$, no matter how small, for a sufficiently large r, the set divides into two subsets with one subset having the property that the probability $P(K)$ of any sequence in it satisfies

$$e^{-r(R+\eta)} < P(K) < e^{-r(R-\eta)}$$

and the other subset having the property that the sum of the probabilities of all sequences in it is less than ε. The expression

$$R = -\sum_i \sum_j p_j P_{ij} \ln P_{ij}$$

is a measure of the amount of information obtained when the Markov chain moves one step ahead.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

We note that there are in total n^r possible r-term sequences. If we were to arrange these sequences in order of decreasing probability $P(K)$, then another theorem (Khinchin, 1957) says that if we sum these probabilities until the sum just exceeds the positive number λ that satisfies $0 < \lambda < 1$, then the number $N_r(\lambda)$ of sequences used satisfies

$$\lim_{r\to\infty} \frac{\ln N_r(\lambda)}{r} = R,$$

independent of the value of λ.

While the proof of these two theorems is outside the scope of this book, they make the important point that while the number of possible sequences of events equals $n^r = \exp(r \ln n)$, the number the Markov chain selects is approximately $\exp(rR)$. In other words, e^R is the effective number of candidate states that we can choose for the next step in the Markov chain. Since the maximum value of the entropy is $\ln n$, we almost certainly have $R < \ln n$. For example, in the case of the single-spin update for the Ising model, we choose one of N spins at random and choose between two possible choices (spin up or spin down) for the value of the spin in the next Monte Carlo step. Since there are $N \times 2$ choices for the next spin configuration, R is bounded from above by $\ln(2N)$, and it is natural to assume that R is of the order of $\ln N$. Considering $n = 2^N$ in this example, R is not only smaller than $\ln n$, it is much smaller than $\ln n$ for large systems. As is clear from this example, in typical situations, the latter theorem by Khinchin indicates that a negligibly small fraction of the total number of sequences accounts for almost all the probability; that is, for this small fraction $\sum_K P(K) \approx 1$. Therein lies the power of Markov chain sampling.

We can also show that the entropy decreases as the chain steps away from the initial state. The behavior of the entropy, while similar, does differ from the actual relaxation behavior of the chain to stationarity, which depends on the magnitude of the second largest eigenvalue of the transition probability matrix. The entropy content of the chain does provide a complementary perspective on the behavior of Markov chains and why they can be an effective sampler of distributions of a large number of random variables.

Suggested reading

M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*, vol. 1: *The Basics* (New York: Wiley-Interscience, 1986), chapters 1–3 and appendix.

W. H. Press, S. A. Teukolsky, W. T. Vettering, and B. P. Flannery, *Numerical Recipes* (Cambridge University, 1992), chapter 7.

J. S. Liu, *Monte Carlo Strategies in Scientific Computing* (New York: Springer, 2001), chapters 5, 7, and 13.

Exercises

2.1 Use the method of mathematical induction to prove the validity of (2.13); that is, show the result is true for $n = 1$, assume it is true for n, and then show it is true for $n + 1$.

2.2 If $0 \leq p \leq 0$ and $q = 1 - p$, the binomial distribution

$$p(k) = \binom{n}{k} p^s q^{n-k} \tag{2.34}$$

describes the probability of k successes in n trials. Show that one way to sample k is

$$k = \min_n \{\zeta_1 \zeta_2 \cdots \zeta_n \leq p\}.$$

2.3 If $0 \leq p \leq 0$ and $q = 1 - p$, show that

$$\binom{n}{k} p^k q^{n-k} = \frac{(np)^k}{k} \left(1 - \frac{np}{n}\right)^n Q_n,$$

where

$$Q_n = \frac{\prod_{r=2}^{k} \left(1 - \frac{r-1}{n}\right)}{(1-p)^k}.$$

Now let $p = \lambda/n$ for $n > \lambda$ and show that as $n \to \infty$, $Q_n \to 1$ and

$$\binom{n}{k} p^k q^{n-k} \to \frac{\lambda^k}{k!} e^{-\lambda} \quad \text{as } n \to \infty.$$

This result shows that for large n and small p the binomial distribution is approximately the same as the Poisson distribution, provided np is constant.

2.4 Propose a method to sample x from the density

$$p(x) = \sum_{i=1}^{N} f_i(x), \quad \text{where } f_i(x) \geq 0$$

if the f_i are easily normalized and if they are not.

2.5 Consider a two-state Markov chain defined by the stochastic matrix

$$P = \begin{pmatrix} \alpha & 1-\beta \\ 1-\alpha & \beta \end{pmatrix} \quad \text{for } 0 < \alpha, \beta < 1.$$

1. Find its eigenvalues and verify that the dominant eigenvalue $\lambda_1 = 1$.
2. Find its left- and right-hand eigenvectors x_i and y_i for each λ_i and verify that the components of the left-hand eigenvector of the dominant eigenvalues are all equal and that $x_i^T y_j = 0$ if $i \neq j$.

3. Use these eigenvectors to find P^n analytically.
4. Show that as $n \to \infty, P^n p^0$ is independent of the vector p_0 if its components sum to unity.
5. Show that the limiting vector of $P^n p^0$ is the right-hand eigenvector associated with λ_1.

2.6 Demonstrate that the Metropolis-Barker (2.26) and the Metropolis-Hastings (2.28) algorithms satisfy detailed balance. Do the same for the heat-bath algorithm.

2.7 A typical simulation uses just one sampling method, say, either the Metropolis or the heat-bath method. For the heat-bath method, the conditional probabilities may not always be as easy to sample as is the case for the Ising model.

1. Discuss the algorithmic issues and opportunities of using the Metropolis algorithm within the heat-bath algorithm to do this sampling.
2. Propose a scenario where it might be advantageous to use the heat-bath algorithm within the Metropolis algorithm.

2.8 If the proposal probability T_{ij} is independent of the current state and equals $\pi(i)$, the transition probability for the Metropolis-Hastings algorithm is

$$P_{ij} = \begin{cases} \pi_i \min\left(0, w_i/w_j\right) & \text{if } i \neq j, \\ \pi_j + \sum_k \pi_k \max\left(0, 1 - w_k/w_j\right) & \text{if } i = j, \end{cases}$$

where p_i is the target distribution and $w_i = p_i/\pi_i$.

1. Show that this transition probability satisfies detailed balance.
2. If the states are labeled so that $w_1 \geq w_2 \geq \cdots \geq w_n$, show that

$$P = \begin{pmatrix} \pi_1 + \lambda_1 & \pi_1 & \cdots & \pi_1 & \pi_1 \\ p_2/w_1 & \pi_2 + \lambda_2 & \cdots & \pi_2 & \pi_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{n-1}/w_1 & p_{n-1}/w_2 & \cdots & \pi_{n-1} + \lambda_{n-1} & \pi_{n-1} \\ p_n/w_1 & p_n/w_2 & \cdots & p_n/w_{n-1} & \pi_n \end{pmatrix}$$

where $\lambda_k = \sum_k (\pi_i - p_i/w_k)$. Use your knowledge of the value of the first eigenvalue and the components of eigenvectors of a transition matrix to show that $\lambda_2 = 1 - 1/w_1$. What are the corresponding eigenvectors?
3. Argue that λ_k is the probability of rejection if the next step is the current state at k. Liu (2001) has shown that for this transition probability the λ_k are the eigenvalues of P.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press

2.9 Starting with

$$S(A) = -\sum_i p(A_i) \ln p(A_i)$$

and

$$S(A, B) = -\sum_{ij} p(A_i, B_j) \ln p(A_i, B_j),$$

prove $S(A, B) = S(A) + S(A|B)$.

https://doi.org/10.1017/CBO9780511902581.003 Published online by Cambridge University Press