

## 13.1: Basic Formulation

The basic idea behind the MCMC method is simple. Suppose we have a set of states labeled by an integer index  $n \in \{0, 1, 2, \dots\}$ , where each state is associated with a probability  $\pi_n$ . For example, in statistical mechanics, for a system maintained at a constant temperature  $T$ , each state occurs with probability

$$\pi_n = \frac{\exp\left(-\frac{E_n}{kT}\right)}{Z}, \quad (13.1.1)$$

where  $E_n$  is the energy of the state,  $k$  is [Boltzmann's constant](#), and  $Z$  is the partition function

$$Z = \sum_n \exp\left(-\frac{E_n}{kT}\right). \quad (13.1.2)$$

From  $\{\pi_n\}$ , we would like to calculate various expectation values, which describe the thermodynamic properties of the system. For example, we might be interested in the average energy, which is defined as

$$\langle E \rangle = \sum_n E_n \pi_n. \quad (13.1.3)$$

The most straightforward way to find  $\langle E \rangle$  is to explicitly calculate the above sum. But if the number of states is very large, this is prohibitively time-consuming (unless there is a tractable analytic solution, and frequently there isn't). For example, if we are interested in describing 1000 distinct atoms each having 2 possible energy levels, the total number of states is  $2^{1000} \approx 10^{301}$ . Trying to calculate a sum over this mind-bogglingly many terms would take longer than the age of the universe.

The MCMC method gets around this problem by *selectively* sampling the states. To accomplish this, we *design* a Markov process whose stationary distribution is identically equal to the given probabilities  $\pi_n$ . We will discuss how to design the Markov process in the next section. Once we have an appropriate Markov process, we can use it to generate a long Markov chain, and use that chain to calculate [moving averages](#) of our desired quantities, like  $\langle E \rangle$ . If the Markov chain is sufficiently long, the average calculated this way will converge to the true expectation value  $\langle E \rangle$ .

The key fact which makes all this work is that the required length of the Markov chain is usually *much* less than the total number of possible states. For the above-mentioned problem of 1000 distinct two-level atoms, there are  $10^{301}$  states, but a Markov chain of as few as  $10^6$  steps can get within several percent of the true value of  $\langle E \rangle$ . (The actual accuracy will vary from system to system.) The reason for this is that the vast majority of states are extremely unlikely, and their contributions to the sum leading to  $\langle E \rangle$  are very small. A Markov chain can get a good estimate for  $\langle E \rangle$  by sampling the states that have the highest probabilities, without spending much time on low-probability states.

### 13.1.1 The Metropolis Algorithm

The MCMC method requires us to design a Markov process to match a given stationary distribution  $\{\pi_n\}$ . This is an open-ended problem, and generally there are many good ways to accomplish this goal. The most common method, called the **Metropolis algorithm**, is based on the principle of detailed balance, which we discussed in the article on Markov chains. To recap, the principle of detailed balance states that under generic circumstances (which are frequently met in physics), a Markov process's transition probabilities are related to the stationary distribution by

$$P(n|m) \pi_m = P(m|n) \pi_n \quad \text{for all } m, n. \quad (13.1.4)$$

The Metropolis algorithm specifies the following Markov process:

1. Suppose that on step  $k$ , the system is in state  $n$ . Randomly choose a candidate state,  $m$ , by making an unbiased random step through the space of possible states. (Just how this choice is made is system-dependent, and we'll discuss this below.)
2. Compare the probabilities  $\pi_n$  and  $\pi_m$ :
  - If  $\pi_m \geq \pi_n$ , *accept* the candidate.
  - If  $\pi_m < \pi_n$ , *accept* the candidate with probability  $\pi_m/\pi_n$ . Otherwise, *reject* the candidate.
3. If the candidate is accepted, the state on step  $k+1$  is  $m$ . Otherwise, the state on step  $k+1$  remains  $n$ .
4. Repeat.

Based on the above description, let us verify that the stationary distribution of the Markov process satisfies the desired detailed-balance condition. Consider any two states  $a$ ,  $b$ , and assume without loss of generality that  $\pi_a \leq \pi_b$ . If we start from  $a$ , suppose we choose a candidate step  $a \rightarrow b$  with some probability  $q$ . Then, according to the Metropolis rules, the probability of actually making this transition,  $a \rightarrow b$ , is  $q$  times the acceptance probability 1. On the other hand, suppose we start from  $b$  instead. Because the candidate choice is unbiased, we will choose a candidate step  $b \rightarrow a$  with the same probability  $q$  as in the previous case. Hence, the transition probability for  $b \rightarrow a$  is  $q$  times the acceptance probability of  $\pi_a/\pi_b$ . As a result,

$$\begin{cases} P(b|a) = q \times 1 \\ P(a|b) = q \times \frac{\pi_a}{\pi_b} \end{cases} \Rightarrow P(a|b) \pi_b = P(b|a) \pi_a \quad (13.1.5)$$

Since this reasoning holds for arbitrary  $a$ ,  $b$ , the principle of detailed balance implies that the stationary distribution of our Markov process follows the desired distribution  $\{\pi_n\}$ .

### Expression in Terms of Energies

In physics, the MCMC method is commonly applied to thermodynamic states, for which

$$\pi_n = \frac{\exp\left(-\frac{E_n}{kT}\right)}{Z}. \quad (13.1.6)$$

In such cases, the Metropolis algorithm can be equivalently expressed in terms of the state energies:

1. Suppose that on step  $k$ , the system is in state  $n$ . Randomly choose a candidate state,  $m$ , by making an unbiased random step through the space of possible states.
2. Compare the energies  $E_n$  and  $E_m$ :
  - If  $E_m \leq E_n$ , *accept* the candidate.
  - If  $E_m > E_n$ , *accept* the candidate with probability  $\exp[(E_n - E_m)/kT]$ . Otherwise, *reject* the candidate.
3. If the candidate is accepted, the state on step  $k + 1$  is  $m$ . Otherwise, the state on step  $k + 1$  remains  $n$ .
4. Repeat.

### 13.1.2 Stepping Through State Space

One way of thinking about the Metropolis algorithm is that it takes a scheme for performing an *unbiased* random walk through the space of possible states (represented by our candidate choices), and converts it into a scheme for performing a *biased* random walk. The biased random walk corresponds to a Markov process with the stationary distribution we are interested in.

The way the Metropolis candidates are chosen (i.e., the "unbiased random walk" part) varies from system to system, and once again there are multiple valid schemes that we could employ. For example, suppose we have a collection of 6 atoms, where each atom can be in the level labeled 0 or the level labeled 1. Each state of the overall system is described by a list of 6 symbols, e.g. 011001. Then we can make an unbiased walk through the "state space" by randomly choosing one of the 6 atoms (with equal probability), and flipping it. For example, we might choose to flip the second atom:

$$011001 \xrightarrow{\text{flip second atom}} 001001 \quad (q = 1/6). \quad (13.1.7)$$

If we start from the other state, the reverse process has the same probability:

$$001001 \xrightarrow{\text{flip second atom}} 011001 \quad (q = 1/6). \quad (13.1.8)$$

Hence, this scheme for walking through the "state space" is said to be *unbiased*. Note that, for a given walking scheme, it is not always possible to connect every two states by a single step; for example, in this case we can't go from 000000 to 111111 in one step.

There is more than one possible walking scheme; for instance, a different scheme could involve randomly choosing *two* atoms and flipping them. Whatever scheme we choose, however, the most important thing is that the walk must be unbiased: each possible step must occur with the same probability as the reverse step. Otherwise, the above proof that the Metropolis algorithm satisfies detailed balance would not work.

This page titled [13.1: Basic Formulation](#) is shared under a [CC BY-SA 4.0](#) license and was authored, remixed, and/or curated by [Y. D. Chong](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.