DRP Fall 25, Markov Chain Monte Carlo

Runze Lu, Mentor: Lewis Pan

December 15, 2025

Abstract

This is the write up for UC Berkeley's Mathematics department's Directed Reading Program, Fall 2025. In this write-up, I aim to provide a brief introduction to Markov Chain Monte Carlo (MCMC) methods, focusing on the Metropolis-Hastings algorithm, acceleration via Replica Exchange, and the Ising model. The code for this write-up can be found here.

This write up is largely based on the University of Washington's Stat 516 lecture notes by Yen-Chi Chen [4]. The code for this write up referenced Tanya Schlusser's blog post on MCMC and the Ising model in which they elaborate more on other interesting problems on the Ising model and practical python implementations [3].

1 Background: Markov Chain, Monte Carlo

[4]

1.1 Stochastic Process and Markov Chains

Before the discussion on Markov Chain Monte Carlo methods, here are some definitions that are relevant for the purposes of this short write up.

1.1.1 Important/Relevant Definitions

Definition 1. ***Random Variable***
A random variable $X(\omega) : \Omega \to \mathbb{R}$ is a mapping/function from a sample space Ω to a real number.

Example 1. Flipping a coin 2 times gives the sample space $\Omega = \{HH, HT, TH, TT\}$. We can define X to be the number of tails in two tosses. Then $X\big(\{TT\}\big) = 2, X\big(\{HH\}\big) = 0$.

Definition 2. ***Discrete-time Stochastic Process***

A discrete-time stochastic process is a family of random variables X_t, where $t \in T$, $T = \{0, 1, 2, 3, ...\}$.

Definition 3. ***Discrete-time Markov Chain***

A discrete-time Markov chain is a discrete-time stochastic process $\{X_0, X_1, X_2, ...\}$ such that:

$$P(X_n = x_n | X_{n-1} = x_{n-1}, ..., X_0 = x_0) = P(X_n = x_n | X_{n-1} = x_{n-1})$$

for all $n \geq 0, x_n, x_{n-1}, ...x_0 \in S$, where S is the state space.

There are two common types of stochastic processes: the discrete case which is defined above, and the continuous case in which T is continuous (2), e.g. $T = [0, \infty)$. In this write up, we are only concerned with the discrete case.

Next we introduce some concepts that arise when analyzing Markov chains. The definitions are kept short and much theory is omitted, for more details see [4].

Definition 4. ***Homogeneous Markov Chain***

A Markov chain is called homogeneous if the probability of moving from one state to another is the same no matter at which step it occurs:

$$P(X_n = x_n | X_{n-1} = x_{n-1}) = p_{ij}, \ \forall n \geq 0$$

where p_{ij} is the probability of going from state i to state j. $i, j, \in S$. And n denotes the time step.

Definition 5. ***Transition Probability Matrix***

Assuming that a Markov chain is homogeneous (4)*, then it makes sense to talk about its transition probability matrix P, where P_{ij} denotes the transition probability from state i to state j.*

Example 2. The transition state matrix

$$P = \begin{pmatrix} 0.1 & 0.9 \\ 0.5 & 0.5 \end{pmatrix}$$

means that state 0 transitions to state 1 with probability $p_{01} = 0.9$ at any time step, etc.

Definition 6. ***Irreducible Markov Chains***

A Markov chain is called irreducible if every pair of states can reach each other (eventually). Note that this does not mean that $p_{ij} > 0$ for all i, j. A state can reach another state through a series of "hops".

We can actually think of a homogeneous Markov chain (4) as a directed graph, with the states as vertices and the transition probability as edge weights. In this interpretation, the transition probability matrix (5) is the adjacency list of this graph, and a Markov chain being irreducible would equate to the induced graph being strongly connected.

Definition 7. ***Recurrence and Transience***

A state i is recurrent if the probability of starting at state i and coming back to state i eventually (possibly in infinite time) is 1.

- *A state i is* ***transient*** *if it is not recurrent.*
- *Given state i is recurrent, it is* ***positive recurrent*** *if the the expected number of time steps needed to return to i starting from i is finite.*
- *A state i is* ***null recurrent*** *if it is not positive recurrent*

Note that if a state is positive recurrent, then the Markov chain can visit this state infinitely often.

In the context of MCMC sampling, we always want our Markov chain to be homogeneous, irreducible and positive recurrent. These properties work in tandem with the next two definitions in MCMC's convergence theory.

Definition 8. ***Stationary Distribution***

A probability vector π on a Markov chain state space is called a ***stationary distribution*** *if*

$$\pi^T P = \pi^T,$$

where P is the transition matrix (5)*. Intuitively, this is the state that the system tends to in the limit $t \to \infty$. This distribution is stationary in that right applications of P does not modify the distribution π. It is important to think of this in the long-term average sense.*

Definition 9. ***Global Balance and Detailed Balance***

- *A vector π satisfies **global balance** if $\pi^T = \pi^T P$. Note that this is the **same equality as stationary distribution*** (8)
- *A vector π satisfies **detailed balance** if $\pi_i P_{ij} = \pi_j P_{ji}$ for all i, j*

The notion of global and detailed balance stems from looking at the flow of probability. Detailed balance ensures pairwise symmetric flow of probability.

1.1.2 Important/Relevant Properties

Proposition 1. ***Condition for Positive Recurrence***

An irreducible homogeneous Markov chain (4), (6) *on a finite state space is positive recurrent* (7).

Proposition 2. ***Detailed Balance $\implies$ Global Balance***

Theorem 1. ***Existence of Stationary Distribution***

An irreducible homogeneous Markov chain is positive recurrent if and only if it has a stationary distribution. And if a stationary distribution π exists, it is unique and $\pi_i > 0$ for all $i \in S$.

Note that this implies that an irreducible homogeneous Markov chain on a finite state space has an unique stationary distribution

Theorem 2. ***Ergodic Theorem***

Let $\{X_n\}$ be an irreducible, homogeneous, and positive recurrent Markov chain on state space S with stationary distribution π, Let $f : S \to \mathbb{R}$ such that $\sum_{i \in S} |f(i)|\pi_i < \infty$. Then for any initial distribution

$$\frac{1}{N} \sum_{j=1}^{N} f(X_j) \xrightarrow{a.s.} \sum_{i \in S} f(i)\pi_i \tag{1}$$

Here, a.s. stands for almost surely, although for the purposes of this short write up, it suffices to just think of this as convergence in the standard sense.

For detailed proofs of these propositions and theorems, see [4]

1.2 Monte Carlo

The idea of Monte Carlo methods boils down to solving problems with randomness. One classic example is estimating the value of π via projecting random points

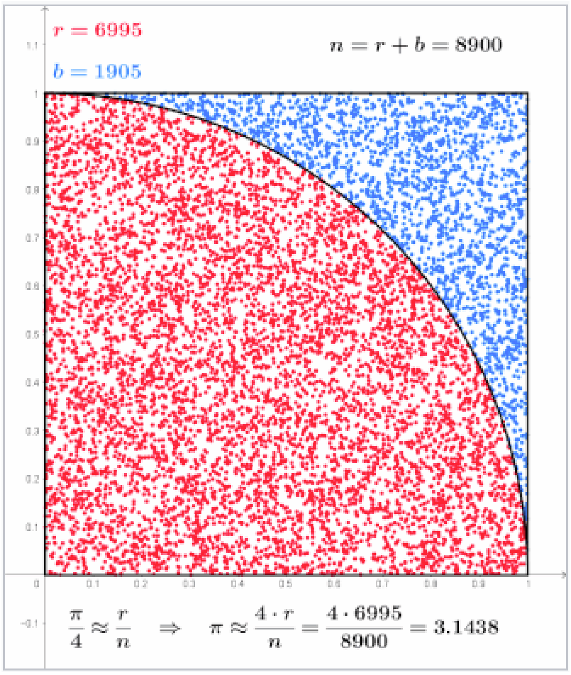


Figure 1: To estimate the value of π, we can project random points onto the space $[0, 1] \times [0, 1]$. Evaluating the ratio of the points inside the quarter circle and the total number of points gives an approximation of π, [2]

2 Acceptance-rejection sampling

We now focus on the problem of sampling from an arbitrary probability distribution. The simplest case is sampling from the uniform distribution. This is straightforward since we can just generate a random number and scale/manipulate accordingly, but how about arbitrary distributions? It turns out that even if we only know how to directly sample from the uniform distribution, we can still sample from any distribution as long as we know its probability density function $f(x)$ in closed form. One way to do this is via acceptance-rejection sampling.

The procedure of acceptance-rejection sampling goes like this:

1. Choose a proposal distribution $p(x)$ that we know how to directly sample from e.g. uniform
2. Choose $M \geq \sup_x \frac{f(x)}{p(x)}$
3. Generate random number Y from p and another random number U from $\text{Uni}(0, 1)$, the uniform distribution over $[0, 1]$.
4. If $U < \frac{f(Y)}{M \cdot p(Y)}$, accept the proposal. Otherwise start from 3 and try another pair of Y and U

To sample more than 1 point from our desired distribution, simply repeat the procedure until we have $X_0, X_1, ..., X_n$. Amazingly, X will have the same density function as p.

Intuitively, in the case of using the uniform distribution as the proposal p, we can think of this as placing an upper bound on $f(x)$. Whenever we generate a random point, we look at how close this upper bound and $f(x)$ are. The upper bound and $f(x)$ will be closer if $f(x)$ is larger, proportional to the original distribution, and hence more likely to be sampled since $\frac{f(Y)}{M \cdot p(Y)}$ is closer to 1 and more likely to be larger than a random number from 0 to 1.

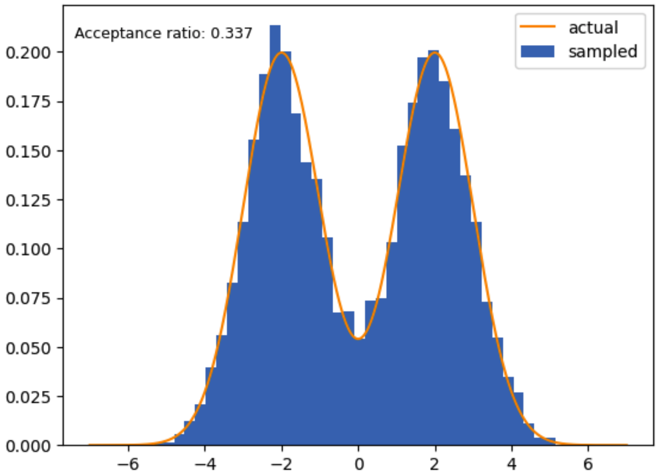


Figure 2: Sample from mixed Gaussian distribution $p(x) = 0.5 \cdot N(x, 2, 1) + 0.5 \cdot N(x, -2, 1)$ using the acceptance-rejection algorithm outlined above. More details can be found in the linked Jupyter notebook.

It is worth mentioning that we would like to choose a good proposal distribution p and M to be as small as possible, since both can impact the number of iterations we need to achieve a desired number of samples. If we choose a bad proposal function, M may need to be extremely large, in which case $\frac{f(Y)}{M \cdot p(Y)}$ will be small, meaning that we will likely reject more samples. However, this will still sample correctly, albeit not as efficiently.

3 Markov Chain Monte Carlo (MCMC)

3.1 Why MCMC

We now know that if we are given any density $f(x)$ in closed form, we can sample from it. But what if what we have is not exactly a density function, but a density function up to some unknown/unattainable constant? It might seem strange why this would ever be the case, but it turns out that this is common in practice.

3.1.1 Motivating Example, the Ising Model

The Ising model is a popular pedagogical and also practical model. In short, it is a mathematical model of ferromagnetism that represents the spins of the atoms as a lattice. For example, one possible configuration of a 3×3 periodic lattice could look like this:

$$\begin{pmatrix} 1 & 1 & -1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix}$$

The Hamiltonian (energy) of the Ising model is as follows:

$$H(s) = -J \sum_{<i,j>} s_i s_j - h \sum_i s_i \tag{2}$$

Where the first term sums over all nearest-neighbor pairs (up to four neighbors), and the second sums over all entries. J, h are user-specified constants that controls the strength of interactions in the model.

We are interested in calculating what is known as the partition function

$$Z = \sum_{s \in \{-1,1\}^{n^2}} e^{-\beta H(s)}. \tag{3}$$

Here, β denotes the inverse temperature of the system. The sum is taken over all possible configurations of the lattice. The partition functions is used to normalize the Boltzmann distribution that gives the probability of being in a certain state:

$$\pi(s) = \frac{e^{-\beta H(s)}}{Z}, \tag{4}$$

We would love to be able to sample from this distribution to study many interesting quantities such as the energy, magnetization, and phase behaviors of the model. For smaller systems, such as $n = 3$, this is reasonable, since there are in total only $2^{3^2} = 2^9$ possible configurations. However, for larger values n of practical interest, this is infeasible, as even taking $n = 10$ would give $2^{10^2} = 2^{100}$ configurations.

What is practical to calculate is the probability density function $\pi(s)$, to be normalized by the constant $\frac{1}{Z}$. This is where MCMC comes in.

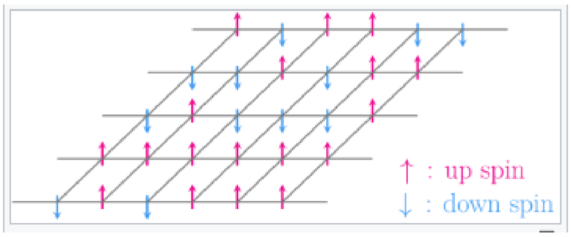


Figure 3: The 2D Ising model with up and down spins [1]

3.2 Metropolis-Hastings

[4]

The MCMC algorithm that I will introduce is the Metropolis-Hastings algorithm. There are many other MCMC algorithms such as Gibbs sampling and Hamiltonian Monte Carlo, we will focus on Metropolis-Hastings in this write up.

The Metropolis-Hastings algorithm goes like this:

1. Have an initial state $x_0 \in S$ and a proposal function $q(i|j)$, $i, j \in S$
2. Start with $X_0 = x_0$
3. repeat the following for $n = 0, 1, 2, ..., N$
   (a) Simulate candidate from proposal function $C \sim q(y|X_n = i)$, suppose that $C = j$
   (b) Calculate the Metropolis-Hastings acceptance probability:
   $$a = min\left\{\frac{\pi_j \times q(i|j)}{\pi_i \times q(j|i)}, 1\right\} \tag{5}$$
   (c) Generate $U \sim Uni(0, 1)$
   (d) Accept or reject the candidate
   $$X_{n+1} = \begin{cases} C & \text{if } U \leq a \\ X_n & \text{if } U > a \end{cases} \tag{6}$$

3.2.1 Why this works, and how this is helpful

The Metropolis-Hastings algorithm helps us correctly sample from our target distribution because of this remarkable property:

Proposition 3. ***Metropolis-Hastings and Stationary Distribution***

Assume that $\pi(i) > 0$ *and that*

$$p(i|j) > 0 \iff p(j|i) > 0 \quad \textit{for all } i, j \in S.$$

Then the Metropolis-Hastings algorithm generates a ***Markov chain*** $X_0, X_1, \ldots, X_n$ (3) *with* ***stationary distribution*** π. (8)

The proof of this proposition [4] involves showing that the Markov chain satisfies detailed balance (9), which is ensured by the Metropolis-Hastings acceptance probability. Also recall that the **Ergodic Theorem** (2) tells us that the behavior of an irreducible, homogeneous, positive recurrent Markov chain with a stationary distribution will converge to its stationary distribution regardless of the initial state. For this reason, when actually sampling, we might want to discard some samples at the start **(mixing phase)**, since they are not being sampled from the target distribution yet.

The other important feature of the Metropolis-Hastings algorithm happens when we take $\frac{\pi_j}{\pi_i}$ in the acceptance probability step. Although right here π denotes the actual probability density function, if π is in the form Cf, where f is a function and C is the normalizing constant, we simply have: $\frac{\pi_j}{\pi_i} = \frac{f_j}{f_i}$. In the case of the Ising model, assuming we have a symmetric proposal function $q(j|i) = q(i|j)$, the acceptance probability reduces to

$$a = \min\left\{\frac{\pi_j}{\pi_i}, 1\right\} = \min\left\{\frac{\frac{e^{-\beta H(j)}}{Z}}{\frac{e^{-\beta H(i)}}{Z}}, 1\right\} = \min\left\{\frac{e^{-\beta H(j)}}{e^{-\beta H(i)}}, 1\right\} = \min\left\{e^{-\beta\left[H(j)-H(i)\right]}, 1\right\} \tag{7}$$

We can see Z is canceled out and all that is required is evaluating $e^{-\beta H}$, which is exactly what we want.

3.3 Replica Exchange

3.3.1 Motivation

On distributions such as the Gaussian mixture that we've been experimenting with, Metropolis-Hastings works perfectly. However, when dealing with more intricate distributions, Metropolis-Hastings might get trapped, and fail to explore the entire distribution. One classic example is the double-well potential function $P(x)$:

$$\begin{aligned} V(x) &= a(x^2 - b^2)^2 \\ \beta &= \frac{1}{k_B T} \\ P(x) &\propto e^{-\beta V(x)} \end{aligned} \tag{8}$$

Here V has two minima at $x = \pm b$, hence the name. At higher temperatures, β becomes smaller, and therefore P will be more evenly distributed. On the other hand, at lower temperatures, β becomes larger, and the effects of non-zero $V(x)$ is more pronounced, causing P to be flat almost everywhere besides around the two minima.

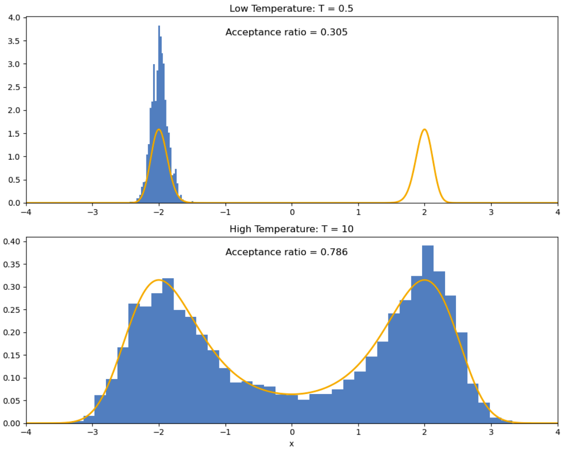


Figure 4: Running Metropolis-Hastings on the double-well potential (8) gives good results at higher temperatures but only samples from one well at lower temperatures. Note that the graph illustrates the probability distribution of double-well but not the energy, at the two peaks are actually low energy states, and the flat region correspond to high energy.

As we can see, Metropolis-Hastings performs badly at low temperature. This is because when we try to take a step j to escape the well, our acceptance probability is $\min\{e^{-\beta\left[V(j)-V(i)\right]}, 1\}$. Since β is large, any sizable, positive (escaping from low to high energy) $V(j) - V(i)$ will give us a vanishingly small acceptance probability, and Metropolis-Hastings almost always reject those proposals. On the other hand, when $V(j) - V(i) < 0$ or is very small, a much higher acceptance probability is attained, and we get to explore other states, but only in the low-energy vicinity.

Although in theory 3, Metropolis-Hastings would eventually sample correctly from distributions like the low-temperature double well if given enough time, in practice we should never expect to get good results in a practical time span from just using the standard algorithm.

3.3.2 The algorithm

The idea of Replica Exchange is to simulate multiple Markov chains at different temperatures and have the higher temperature chains explore the full distribution, reaching otherwise unreachable states for the low temperature chain to attempt to swap to. To guarantee that we are still sampling from the correct distribution, we add an extra acceptance-rejection step when attempting to swap to ensure detailed balance. (9)

1. Choose a ladder of temperatures $\{T_i\}$
2. Initialize configuration x_i for each replica
3. For each MCMC step:
    (a) Perform standard Metropolis updates independently on each replica with acceptance probability
    $$a = \min\{1, e^{-\beta_i \left[V(x_j) - V(x_i)\right]}\} \tag{9}$$
    (b) Every few steps, attempt to exchange between adjacent replicas $(x_n, x_{n+1}) \rightarrow (x_{n+1}, x_n)$ according to the acceptance probability:
    $$p_{\text{swap}} = \min\left\{1, \exp\left[(\beta_i - \beta_j)\Big(V(x_i) - V(x_j)\Big)\right]\right\} \tag{10}$$
4. Collect the sample from the replica corresponding to our target temperature

Once again, the extra exchange criterion was chosen to satisfy detailed balance (9) with respect to the joint distribution

$$P(\{x_i\}) \propto \prod_{i=1}^{N} e^{-\beta_i V(x_i)} \tag{11}$$

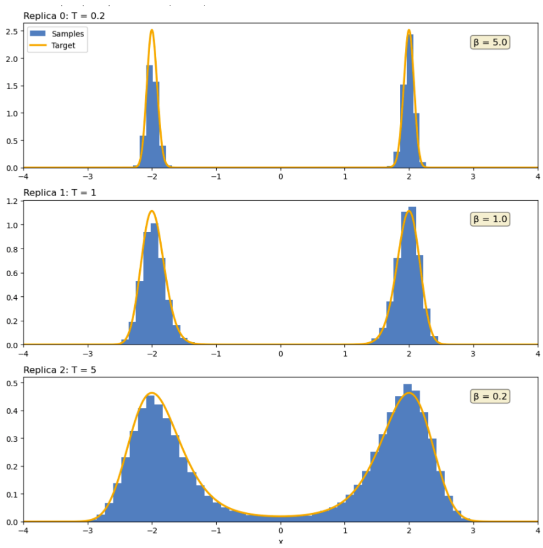


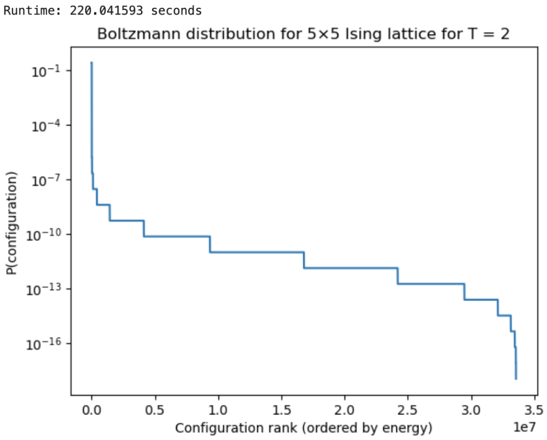
Figure 5: Sampling from the double-well potential (8) with Replica exchange, with $T = [0.2, 1, 5]$

4 Testing MCMC on the Ising Model

Let's now return to the Ising model (3) and see how Metropolis-Hastings helps us sample correctly from the Boltzmann distribution. (4) For simplicity, we will use $\beta = \frac{1}{T}, J = 1, h = 0$. (2) (4)

4.1 Brute-force approach

Recall from earlier that even for as small as $n = 5$, the number of configurations on a $n \times n$ lattice is $2^{5^2} = 2^{25}$. Let's first try the brute force way of summing over all possible configurations to obtain the Boltzmann distribution for this set up.



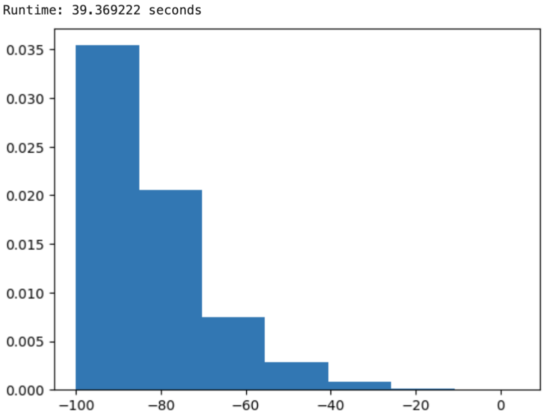
```
Runtime: 220.041593 seconds
```

Figure 6: Calculating the Boltzmann distribution for 5 by 5 lattice by directly summing over all possible configurations to obtain Z, (3) and ordering the configurations from lowest to highest energy.

As we can see, obtaining this distribution took several minutes. In fact, a 5×5 lattice is perhaps the limit for what a laptop could run in a reasonable time. Now lets try sampling using Metropolis-Hastings instead.

4.2 Using MCMC

The only thing that is not yet clear with using MCMC on the Ising model is how we should propose a new state, i.e. what $p(j|i)$ should be. (5) In this example we will attempt to flip one spin at a time which satisfies (7) (See more detail in code by [3]). And since the convergence of MCMC sampling is generally slow, we will only observe a sample after a certain number of samples.



```
Runtime: 39.369222 seconds
```

Figure 7: Sampling 200000 times from the Boltzmann distribution on the 5×5 Ising model at $T = 2$

As we can see, MCMC sampling was able to capture the ladder shaped distribution, sampling more from lower energy states than higher energy states. Also note that the samples that we see here are actually all low-energy states, the "steps" that we see here correspond to the very first few drops in 6, and we almost never sample higher energy states since they are so unlikely.

4.3 Bigger Lattice

To really demonstrate the necessity of MCMC, let's now try $n = 10$. This would be impossible if we try to use the brute-force approach.

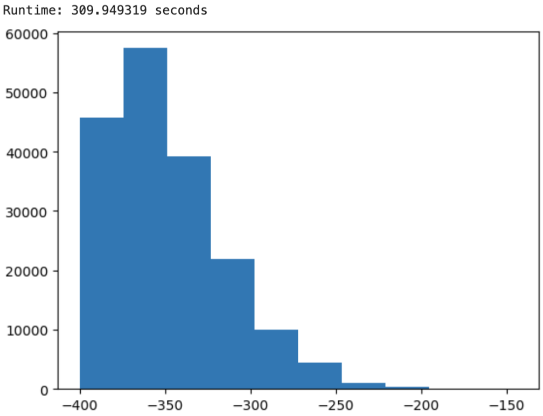


Figure 8: MCMC samples from a 10×10 Ising lattice in reasonable time

We can see that the distribution follows the similar right-skewed pattern 6, an indicator that our sampler is behaving as intended.

4.4 Comments

Mixing time is an important idea in MCMC, it denotes the time it takes for a Markov chain to behave like its stationary distribution and hence for MCMC to truly sample from the target distribution. In general, MCMC algorithms converge slowly. Therefore in the examples above, our samples does not look perfectly like the explicit Boltzmann distribution and that's okay, we might just need to take more samples.

5 Acknowledgement

I am very fortunate to have Lewis as my mentor and I thank him for his guidance and support.

References

[1] Ising model. https://en.wikipedia.org/wiki/Ising_model#References. Accessed: 2025-12-01.

[2] Monte Carlo method. https://en.wikipedia.org/wiki/Monte_Carlo_method. Accessed: 2025-12-01.

[3] Tanya Schlusser. Mcmc and the ising model. https://tanyaschlusser.github.io/posts/mcmc-and-the-ising-model/. Accessed: 2025-12-02.

[4] Yen-Chi Chen and University of Washington Department of Statistics. STAT 516: Stochastic Modeling of Scientific Data — Lecture Notes. https://faculty.washington.edu/yenchic/18A_stat516/. Accessed: 2025-12-01.