Markov Chain Monte Carlo
The Metropolis-Hastings Algorithm

Anthony Trubiano

April 11th, 2018

1 Introduction

Markov Chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from a probability distribution. A handful of methods already exist for this very purpose, such as the inverse transform method or rejection sampling. In addition, MCMC is typically referred to as an extremely bad method, to be used "only when all alternative methods are worse" by leading Monte Carlo researcher Alan Sokal. Given this information, one may ask why is MCMC so prolific in the mathematical sciences? Here we will introduce Markov Chains and one common MCMC method and see what makes these methods so powerful.

1.1 Some Motivation - A Trip to the Past

We'll start with an example of when "all alternative methods are worse". To do this, we go back to some of the physics of the first half of the 20'th century and consider the 2-d Ising model, which is still widely used to model ferromagnetic materials as well as other systems [2]. The model is relatively simple, but still accurately captures significant phenomena that occurs in many complex physical systems.

We will model our material as a lattice of magnetic moments, a quantity that has a direction and determines how much torque a particle will feel in a magnetic field. A material becomes magnetic when there is a bulk alignment of these magnetic moments. We represent the magnetic moments with an $L \times L$ lattice of spins, σ_i for $i = 1, ..., L^2$, that take values of ± 1 for spin up and spin down, respectively. Nearby spins will interact with each other, trying to make their spins parallel. Wanting alignment of spins to be energetically favorable, we let each configuration have the associated potential energy (called the Hamiltonian)

$$E(\sigma) = -\sum_i \sum_{j \in N_i} \sigma_i \sigma_j, \tag{1}$$

where N_i is the set of nearest neighbors of spin i.

From statistical mechanics, the equilibrium distribution, which gives the probability of observing a particular configuration in equilibrium, is given by the Boltzmann distribution

$$\pi(\sigma) = \frac{1}{Z} e^{-\beta E(\sigma)}, \tag{2}$$

where Z is a normalization constant (which is unknown and difficult to compute in most cases), β represents the inverse temperature, $\beta = \frac{1}{k_b T}$, where k_b is Boltzmann's constant, and T is temperature. For low temperature the system tends to be magnetized, which corresponds to all spins being aligned. For high temperature, the system tends to be disordered with nearly the same number of up and down spins, which leads to no bulk magnetization.

One question we can ask is for what value of β does the system go from being magnetized to not? An analytic treatment of this question does exist in 2 dimensions due to Onsager [3], but it is quite complicated and involves a lot of new physics. Is there another way to gather some insight? One possibility is to compute the average magnetization as a function of temperature, where the magnetization is $M(\sigma) = \sum_i \sigma_i$. The average magnetization would then be given by

$$\langle M \rangle = \sum_{\sigma} M(\sigma)\pi(\sigma), \tag{3}$$

where this sum is over every configuration σ. Note that there are 2^{L^2} elements in the state space, which is unfathomably large even for L as small as 10. We cannot hope to enumerate each term in the sum, analytically or in a computer. There is also the fact that Z is unknown and difficult to compute (we need to sum the Boltzmann distribution over the entire state space).

Instead, we can turn to Monte Carlo methods, which use random numbers to compute deterministic answers that are the expected value of some random variable. We can compute an estimator for the average magnetization as

$$\hat{M} = \frac{1}{n}\sum_{i=1}^{n} M(\sigma_i), \tag{4}$$

where σ_i is a sample configuration sampled from $\vec{\pi}$. The Law of Large Numbers ensures this will converge to the true value as $n \to \infty$. Computing this requires us to sample from $\vec{\pi}$. What are the issues with this? Again, we do not know Z and our state space is too large to use the usual sampling methods. We will see how we can use MCMC to get around these issues.

2 Markov Chains

The first thing we ask is how do we go about modeling a random process? Using Markov Processes is a simple way to do so. The main idea is that the future state of a system depends only on the current state, not the entire history of the process. We can make an analogy with deterministic processes in the form of Newton's laws. If we know the position and velocity of a ball in the air at a specific time, Newton's laws tell us where it will be at a later time; it doesn't matter how the ball got to its initial position. Let's see how such processes are constructed.

2.1 Definition

We will stick to the case of discrete time Markov Chains, and the case where the state space is also discrete. Continuous random variables can be handled in the same framework by replacing conditional probability distributions with conditional densities.

We will be working in the state space S. A *discrete time Markov Chain* is a collection of random variables $\{X_i\}_{i=0}^{\infty}$ where $X_i \in S$, that satisfies the Markov Property, which states that the probability of moving to the next state only depends on the current state, not the past. Formally, we have

$$P(X_n = x | X_{n-1} = x_{n-1}, ..., X_1 = x_1, X_0 = x_0) = P(X_n = x | X_{n-1} = x_{n-1}). \tag{5}$$

When the state space is finite, we say that the Markov Chain has a *Probability Transition Matrix*, denoted by $\boldsymbol{P}$, where the (i, j) component gives the probability to transition to state

j given the current state is i. That is

$$P_{ij} = P(X_n = j | X_{n-1} = i). \tag{6}$$

Note that $\boldsymbol{P}$ is a stochastic matrix, i.e. all of its entries are non-negative since they are probabilities, and each row sums to 1, since starting at state i, the chain must go somewhere at the next time step.

In most cases, the transition probabilities are independent of the current time step, n. If this is the case, the process is called a *Time-homogeneous Markov chain*, whose transition probabilities can be characterized as

$$P_{ij} = P(X_1 = j | X_0 = i). \tag{7}$$

In this case, the transition matrix is constant in n, and thus the k-step transition probabilities can be computed as the k-th power of the transition matrix $\boldsymbol{P}$. We will discuss what we can say about what happens when $k \to \infty$, but first we must introduce some other properties of Markov Chains.

2.2 Irreducibility

A Markov chain is said to be *irreducible* if it is possible to eventually reach any state from any other state. More formally, this means that for all pairs of states (i, j), there exists some $m \in \mathbb{Z}$ such that $(\boldsymbol{P}^m)_{ij} > 0$. As visible in Figure 1, irreducibility ensures that there are no two regions in the state space that are not connected, which will play a role in determining the long time behavior of the Markov chain.

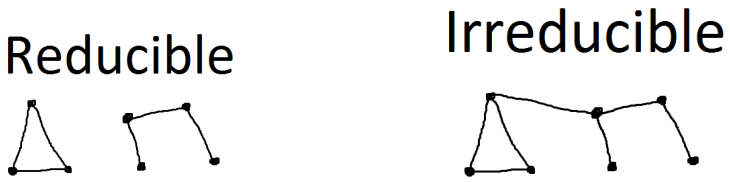

Figure 1: Examples of a reducible and irreducible Markov chain. Lines between nodes correspond to non-zero transition probability, forwards and backwards.

2.3 Periodicity

A state i is said to have *period* k if any return to state i must occur in multiples of k time steps. Formally we have

$$k = \gcd\{n > 0 | P(X_n = i | X_0 = i) > 0\}. \tag{8}$$

If $k = 1$ for state i, then we say state i is *aperiodic*. The Markov chain is aperiodic if all states are aperiodic. It can be shown that an irreducible Markov chain is aperiodic if just one state is aperiodic. A simple example of a periodic Markov chain can be seen in Figure 2.

3 Stationary Distribution and Long Time Behavior

What happens if we run a Markov chain for a long time? Even though the states will jump around indefinitely, maybe the probability distribution for what state the chain is in will converge to a steady-state distribution. This distribution, called the *limiting distribution*, can be characterized by a row vector $\vec{\Pi}$ such that $\vec{\Pi} = \lim_{n\to\infty} \vec{p_0}\boldsymbol{P}^n$ for any initial distribution $\vec{p_0}$. From this definition, we see that $\vec{\Pi}\boldsymbol{P} = \vec{\Pi}$, which will motivate our next definition.

Consider a Markov chain over a finite state space, so the transition probabilities can be represented by the matrix $\boldsymbol{P}$. A *stationary distribution* for $\boldsymbol{P}$, denoted by $\vec{\pi}$, is a row vector that satisfies the following properties:

i. $0 \leq \pi_i \leq 1$, ii. $\sum_i \pi_i = 1$, iii. $\vec{\pi}\boldsymbol{P} = \vec{\pi}$.

Note that property (iii.) can be written component-wise as $\pi_j = \sum_i \pi_i P_{ij}$. From this definition we can see that $\vec{\pi}$ is a left eigenvector of $\boldsymbol{P}$ with eigenvalue 1, normalized such that the sum of its entries is 1. Intuitively, this means that a Markov chain that starts in the stationary distribution, stays in the stationary distribution for all time.

What about a chain that does not start in the stationary distribution? From the definition of limiting distribution, we see that the limiting distribution is indeed a stationary distribution. Is the converse true? The answer is yes, under certain assumptions, as can be seen from the following theorem, proven in [4].

Theorem 1. *Assume a Markov chain over a finite state space is irreducible, aperiodic, and there exists a stationary distribution $\vec{\pi}$. Then $\vec{\pi}$ has $\pi_i > 0$, is unique, and is also the limiting distribution.*

Thus, to determine the limiting distribution, it suffices to find a stationary distribution. We now discuss a special type of stationary distribution.

4 Detailed Balance

A Markov chain is said to be in *detailed balance* (also called reversible) if its stationary distribution $\vec{\pi}$ satisfies

$$\pi_i P_{ij} = \pi_j P_{ji} \quad \text{for all } i, j. \tag{9}$$

Intuitively, you can think of these products as probability fluxes. Detailed balance says that the amount of probability going from state i to j is the same as the amount of probability going from j to i. Thus running the chain backwards is indistinguishable to running it forwards, and we say the system is in statistical equilibrium. How is this related to stationary distributions? We can show that any distribution satisfying detailed balance is a stationary distribution as follows. Let $\vec{\pi}$ satisfy the detailed balance equations (10). Then

$$\sum_i \pi_i P_{ij} = \sum_i \pi_j P_{ji} = \pi_j \sum_i P_{ji} = \pi_j$$

. Thus if we can find a distribution that satisfies detailed balance, we have found the stationary distribution. Now we can introduce the most common MCMC method.

5 Markov Chain Monte Carlo

Remember that we wanted to sample from some probability distribution $\vec{\pi}$. What we can do is construct a Markov chain whose stationary distribution is $\vec{\pi}$, and then simulate the Markov chain. From what we have just discussed, the limiting distribution will be $\vec{\pi}$. The question is, how do we construct this Markov chain?

5.1 The Metropolis-Hastings Algorithm

Assume the Markov chain is in some state $X_n = i$. Let $\boldsymbol{H}$ be the transition matrix for *any* irreducible Markov chain on the state space. We generate X_{n+1} via the following algorithm: [1]

1. Choose a proposal state j according to the probability distribution in the i-th row of $\boldsymbol{H}$.
2. Compute the acceptance probability $\alpha_{ij} = \min\left(1, \frac{\pi_j H_{ji}}{\pi_i H_{ij}}\right)$.
3. Generate a uniform random number $U \sim \text{Uniform}(0, 1)$. If $U < \alpha_{ij}$, accept the move and set $X_{n+1} = j$. Otherwise, reject the move and keep $X_{n+1} = X_n$.

This Markov chain has transition probabilities $P_{ij} = \alpha_{ij} H_{ij}$. Why does this Markov chain have $\vec{\pi}$ as its stationary distribution? We can show it satisfies detailed balance. First, assume that $\pi_j H_{ij} \leq \pi_i H_{ij}$. Then we have

$$P_{ij} = \min\left(1, \frac{\pi_j H_{ji}}{\pi_i H_{ij}}\right) H_{ij} = \frac{\pi_j}{\pi_i} H_{ji} \tag{10}$$

$$P_{ji} = \min\left(1, \frac{\pi_i H_{ij}}{\pi_j H_{ji}}\right) H_{ji} = H_{ji} \tag{11}$$

Thus the detailed balance condition is satisfied since

$$\pi_i P_{ij} = \pi_i \frac{\pi_j}{\pi_i} H_{ji} = \pi_j H_{ji} = \pi_j P_{ji}$$

We can do the same calculation for $\pi_j H_{ij} > \pi_i H_{ij}$, so the chain satisfies detailed balance with respect to $\vec{\pi}$, and thus has $\vec{\pi}$ as its stationary and limiting distribution. Let's see how to apply this algorithm to our Ising Model example.

5.2 Simulating the Ising Model

The first thing the Metropolis-Hastings algorithm calls for is a proposal transition matrix, $\boldsymbol{H}$. Any such matrix over the state space of the Ising Model will be much too large to store or work with, so what can we do? The simplest option is to pick a spin uniformly at random and swap its direction, which is what we will use here. There is an equal probability to move to any accessible state, so the terms involving the matrix $\boldsymbol{H}$ will cancel. Another option is to pick two spins at random and swap them. In general, any symmetric proposal, one that satisfies $H_{ij} = H_{ji}$, will be much easier to work with.

Now we just need an expression for the acceptance probability. Recall the Boltzmann distribution given in Equation (2). Substituting this into the formula, we have

$$\frac{\pi_j}{\pi_i} = \frac{Z}{Z} \frac{e^{-\beta E_j}}{e^{-\beta E_i}} = e^{-\beta(E_j - E_i)} = e^{-\beta \Delta E}. \tag{12}$$

The acceptance probability is then given by the minimum of 1 and this quantity. This is everything we need to run the algorithm and generate a sequence of samples, $\{X_i\}_{i=1}^N$, drawn from the distribution $\vec{\pi}$. Note that the normalization constant, Z, cancels out in this computation. We do not need to know it in order to simulate the Markov chain! Another important thing to note is that to compute ΔE, the change in energy between states i and j, only neighbors of the flipped spin need to be considered, as every other term in the sum remains the same

under our single flip proposal. The sum can be computed efficiently in $O(1)$ time for each proposal, which is crucial to making the algorithm run in a reasonable amount of time.

Now we can try to answer our initial question, for which value of T does the material go from being magnetized to not magnetized? We will choose some values of T, then use Metropolis-Hastings to collect N samples to use in our estimator of the magnetization, given in Equation (4). The results can be seen in Figure 3.

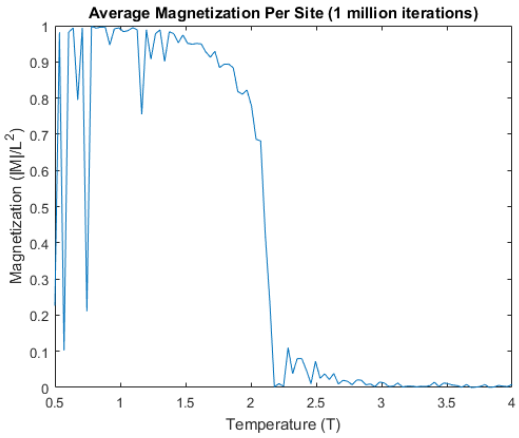

Figure 2: Monte Carlo estimate of average magnetization as a function of temperature.

We do indeed see a sharp transition around the $T = 2.25$ area, separating regions of approximately 0 or 1 magnetization. This is called a phase transition. The fact that the Ising Model produces this behavior is why it is so widely used in modeling systems known to exhibit these transitions. We also see large spikes for low temperature, where we expect to see a magnetization of 1. What happened here? To gain some insight, the magnetization as a function of iteration is plotted in Figure 4.

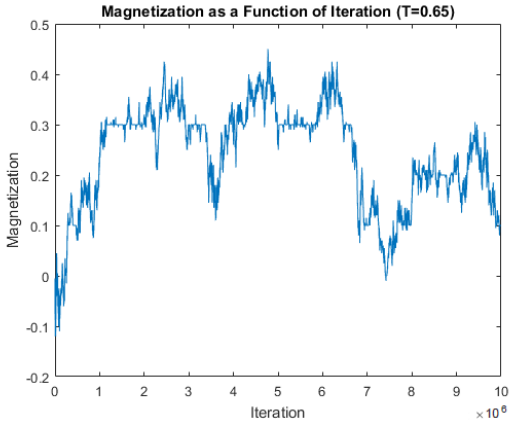

Figure 3: Magnetization as a function of iteration number for the above run of the Markov chain.

We see the magnetization hardly changes over the course of a million iterations. We can imagine the configuration is not changing very much in this case. If we consider the Boltzmann distribution, we see that as temperature decreases, which means β increases, the acceptance probability becomes exponentially small, unless ΔE is negative. Thus only changes that reduce the overall energy get accepted, which can take a long time to be proposed if we are choosing spins to flip uniformly at random. This is a negative consequence of the simplicity of

our choice of proposal distribution. Our samples are highly correlated, which gives a number of "effective samples" much less than the number of total samples generated. Thus our error is much higher than we would expect if we had independent samples. An alternative proposal that works well at low temperatures is called a clustering update. In these methods, we identity connected clusters of the same spin and propose to swap the entire cluster at once. It can be seen that correlations decay rapidly in comparison with the single flip proposal, making these methods much more effective, but more complicated to implement.

5.3 General Properties of MCMC

We have seen how to use MCMC on a particular example and some of the issues that arise. Here we discuss some of these issues and others in a more general case.

- Consider an arbitrary probability distribution $p(x) = \frac{1}{Z} f(x)$, where Z is a normalization constant. We see that Z cancels out in the formula for the acceptance probability, so we don't need to know it to sample from the distribution.
- The proposal distribution can be chosen to be anything that is convenient. The distribution is typically chosen such that roughly some percentage, like 50%, of the proposals are accepted. Choosing a good proposal is more of an art than a science, though there are heuristics in some cases.
- It takes the Markov chain some time to settle into its stationary distribution. This is called the burn in time, and usually these iterations are discarded when computing statistics from the Markov chain.
- The error is $O(N^{-1/2})$ by the Central Limit Theorem, where N is the number of samples. This gives very slow convergence and is the main reason the method is considered "bad". To improve a result by a factor of 10, we need 100 times more samples. This is compounded by the next point.
- Successive samples are correlated. The autocorrelation time, τ, can be computed, which tells you how long until the chain generates an approximately independent sample. For this reason, the effective number of samples is given by N/τ. We want τ to be as small as possible, so we can run the chain for a shorter time. This also warns us to be wary of how Monte Carlo data is presented; giving an N without telling us τ means the data is meaningless.

6 Conclusion

We have seen how to sample from a probability distribution by using Markov Chain Monte Carlo, and more specifically, the Metropolis-Hastings algorithm. This relied on constructing a Markov chain that satisfies detailed balance with respect to the desired distribution out of an arbitrary proposal transition matrix. We saw how choosing a symmetric, or otherwise simple, proposal simplifies the simulation of the Markov chain, but has a trade-off in how effectively the chain explores the state space. If the moves are small, our samples are highly correlated, and there is a large autocorrelation time that reduces our effective number of samples. If the moves are too large, there is a very small probability of accepting the move and we get stuck for long periods of time. Most of the work done in MCMC is to try to find a proposal distribution that is "just right". In fact, work has been done in choosing a proposal such that our probability distribution is the stationary distribution, without invoking detailed balance.

We also applied the Metropolis-Hastings algorithm to the 2-dimensional Ising Model to compute an estimator for the average magnetization. This showed the existence of a phase transition, without having to analytically solve the problem, a daunting task. This is just one example of how MCMC is a relatively easy way to gain insight into a challenging problem when "all other methods are worse".

References

[1] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[2] E. Ising. Beitrag zur theorie des ferromagnetismus. *Physik, Z*, 31(1):253–258, 1925.

[3] L. Onsager. A two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65(1):117, 1944.

[4] W. Winkler. Kai lai chung, markov chains with stationary transition probabilities. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift fr Angewandte Mathematik und Mechanik*, 40(12):574–574.