Study guide

We return to the Ising model to study how the proposal distribution of the Metropolis-Hastings method can be modified to sample much more efficiently than flipping one spin at a time, as before. You experiment with a technique called simulated annealing, which is generally applicable to MCMC, in the pre-class work, and with the Wolff cluster algorithm, which is specific to the Ising model, during the two activities in class.

In order to be prepared for class, you need to be able to do the following.

- Fully understand the Ising model of ferromagnetism.
- Understand how the Wolff cluster algorithm works and why it moves through the state space of the Ising model much faster than the single spin flip model we looked at previously.

Review

- The way we defined and implemented the Ising model before is actually an instance of the Metropolis-Hastings (MH) algorithm. There is a probability distribution from which we are trying to generate samples is

  $$p(x) \propto \exp(-E(x) / T)$$

  $$E(x) = -\sum_{<i,j>} x_i x_j$$

  where E is the energy of the system in state x and T is the temperature. Note that low energy states have high probabilities, which is generally the case.

- Distributions based on energy functions, like the one above, are known as Boltzmann distributions. (Ludwig Boltzmann was a physicist and physicists like to think of systems in terms of their energy functions.)
- This MH algorithm we implemented for the Ising model also happens to follow the hypothesized physical process underlying ferromagnetism — interactions between directions of adjacent electron spins, and random thermal fluctuations. However, there is no reason we can't design more efficient MH algorithms for sampling from the same probability distribution, $p(x)$, even if these algorithms do not correspond to real physical processes.
- This is today's main focus — how to make generating samples from the equilibrium distribution of the Ising model a lot more efficient. We can then use the samples from the state space to calculate useful quantities like the average magnetization.

Shonkwiler & Mendivil

Key point for today: "The Metropolis algorithm will asymptotically approximate the Boltzmann distribution as the run proceeds."

Some important concepts from this reading:

- The microstate of a system is the specific configuration of all of its variables. In the Ising model, the microstate is the collection of all signs of all cells in the grid. We have just been referring to this as the state before.
- In contrast, a macrostate of a system is some quantity that we can calculate from the microstate and that represents an aggregate property of the system as a whole. The main one we have been studying is the average magnetization of the whole grid (or similarly the sum of all spins, which is known as the magnetic moment).
  - There can be many microstates that correspond to the same macrostate. Many different microstates result in the same average magnetization.
  - We are often interested in the distribution over macrostates — for example, the probability distribution over average magnetization at a given temperature.
- The Boltzmann factor describes the probability of observing a microstate with particular energy. The ratio between two Boltzmann factors gives you the acceptance probability of transitioning between two microstates in a Metropolis-Hastings algorithm.
- The magnetic moment is defined as the sum of all the spins in the Ising model — it tells us whether there is any residual positive or negative magnetism in the system. If the magnetic moment is 0, there are as many positive spins as negative spins, and the system as a whole is completely unmagnetized. In Session 11.2, we calculated this macrostate value using the microstates of the simple Ising model and estimated the distribution over the magnetic moment using samples from the Ising model state space.
- The text goes through a microscopic analysis, using a 2 x 2 Ising model, which is always a good idea in any model. Always check that you understand how a model works at small scales — this is also a good opportunity to check your code for bugs! — before looking at macroscopic properties or emergent properties of a simulation.

The Ising simulation as a Metropolis-Hastings algorithm

Our previous Ising model simulation is actually a type of Metropolis-Hastings algorithm. At each update step, the proposal function tells us to flip the sign of a randomly selected cell. The choice of cell is uniformly distributed across the grid.

Since the energy of the whole system is

$$E(x) = -\sum_{<i,j>} x_i x_j$$

where the sum is over all neighboring pairs of cells, and the target probability distribution is the Boltzmann distribution

$$p(x) \propto \exp(-E(x) / T)$$

the acceptance probability is

$$p(y) / p(x) = \exp((E(x) - E(y)) / T)$$

Since all cells remain the same, except for the change in sign in one cell, the energy difference above simplifies to

$$p(y) / p(x) = \exp((-2\, s_{i,j}\, (s_{i+1,j} + s_{i-1,j} + s_{i,j+1} + s_{i,j-1})) / T)$$

where the indexes i and j here represent the randomly selected cell from the model grid.

This is exactly the quantity we used in our Ising model simulation before. This derivation shows why we used this acceptance probability in the first place — because it is the acceptance ratio of a Metropolis-Hastings algorithm that samples from the target distribution, $p(x)$.

Remember the key point, "The Metropolis algorithm will asymptotically approximate the Boltzmann distribution as the run proceeds."

An important corollary is that if we want to sample from a target distribution more efficiently we are free to define a better proposal function (instead of flipping the sign of a uniformly chosen cell) and then using the MH algorithm again. The Wolff cluster algorithm described in today's video is an example of such an efficient proposal function, specialized for generating samples from the Ising model. We explore this algorithm further in class.