Ising Model: Notes

Hands-on Computational Physiscs
Seasons of Code-2023

Simran Sinha
Mentors: Varad Mahashabde, Kaustav Prasad
Indian Institute of Technology Bombay

1 Markov Chains

A Markov process is a process where evolution from the current state into the next one is dependent only on the current state, that is, it doesn't depend on how the system got there (its history). A quintessential example is the conventional random walk, contrasted with intricate scenarios like the self-avoiding random walk, where next steps depend on preceding events.

Our pursuit is to orchestrate a Markov chain that samples the states in such a way that it helps us solve the integral (or sum) by **importance sampling**, that is, the chain should transition from one state to another with the probability needed for solving the sum, sampling the states the same way as the g distribution. To particularize further, in statistical physics the distribution that is often needed is:

$$p_i = \frac{1}{Z} e^{-\beta \epsilon_i}$$

where $\beta = \frac{1}{k_B T}$ and Z is the partition function, $Z = \sum_i e^{-\beta \epsilon_i}$, where the sum takes into account degeneracy, too, by summing over states, not energy level. The expectation value for some macroscopic measurable value A is $\langle A \rangle = \frac{1}{Z} \sum_i A_i e^{-\beta \epsilon_i}$ and that's the kind of sums we want to solve with the simulations. If we manage to use the Boltzmann distribution for importance sampling, the partition function becomes equal with the number of samples and the calculation simplifies to: $\langle A \rangle = \frac{1}{N} \sum_i A_i$.

Very shortly, to obtain a way of getting the required distribution, one writes the **master equation** for the probability of being in a particular state and then demanding that the probability of being in a particular state is constant. This way one finds out that the probability of transitioning into a particular state during a period of time must be equal with the probability of transitioning out during the same period of time, that is, a **balance equation**. A particular solution that satisfies the equality is the detailed balance:

$$p(a)w(a \to b) = p(b)w(b \to a)$$

a, b are two particular states, p is the probability of being in a particular state and w gives the transition rate between them. Arranging the solution a little one finds:

$$\frac{p(a)}{p(b)} = \frac{w(b \to a)}{w(a \to b)}$$

This means that one can obtain a desired probability distribution just by having the proper transition probability between states in a Markov chain! The transition probability is then split in two, a choice probability and an acceptance ratio, $w(a \to b) = c_a(b)A(a \to b)$. The choice probability (also called a trial or selection probability) depends on the system being simulated and the implementation, but we want to have one with an acceptance ratio as high as possible. We don't want to choose a 'next state' many times because it is not accepted.

2 Metropolis Algorithm

the Metropolis algorithm which is a special case where the choice probability is symmetric, that is, the choice probability of picking b if the current state is a is equal with the one of picking a from the state b,

$c_a(b) = c_b(a)$. This gives:

$$\frac{p(a)}{p(b)} = \frac{A(b \to a)}{A(a \to b)}$$

There are many ways to choose the acceptance to obtain the desired ratio of probabilities, the Metropolis one is:

$$A(a \to b) = \min(1, \frac{p(b)}{p(a)})$$

That is, if the probability of the state b is higher than the probability of state a, the state is always accepted, if not, the state b is accepted according to the ratio of probabilities. Even if ignoring all the previous discussion it is quite obvious how this favors high probability states but also samples the low probability ones.

3 Monte Carlo Simulation

Let's consider a case that is expected to be met in many statistical physics calculations, a Boltzmann distribution. For this case the ratio of probabilities turns into:

$$\frac{p(a)}{p(b)} = e^{-\beta(E_a - E_b)}$$

Very shortly, here is the algorithm, I'll detail it when presenting an actual implementation:

- Pick a state at random. Sometimes it's not that random, the algorithm might start with a state corresponding to an infinite or zero temperature.
- 'Thermalize' by walking along the Markov chain. Since the initial state might be quite far from the states that need to be sampled, the algorithm should be run for a while to allow reaching that region of the state space. The 'walking' is detailed below, it is similar with what happens after warmup.
- Pick the next state according with the choice probability.
- If the energy of the next state is lower than the energy of the current state, accept the new state., if not, accept it with the probability $e^{-\beta \Delta E}$.
- Continue with step 3 until enough states were generated. What's 'enough' depends on the particular distribution and autocorrelation 'time' of the Markov chain.
- Use the current state for doing statistics and then go back to step 3.

Some brief implementation details: In many cases you don't have to calculate again and again the energy or other values. Just notice that going from one state to another means an easily calculated change (often that's the case) and use the difference to update the value. Also calculating some values involve exponentials. In many cases (for example when choosing a 'next state' involves some spin flip) the number of exponential values needed is limited. In such case it's better to calculate them in advance and put them in a table to be reused, calculation of the exponentials all over again might be expensive.

4 The model

The Ising model represents a fundamental yet intricately behaving system, noted for its significance in demonstrating universality—where properties span across diverse systems independently of their dynamic intricacies. In this context, we delve into the 2D Ising model on a square lattice, focusing on a specific scenario. The Ising model represents a lattice of magnetic spins, which can take values of either +1 or -1, interacting with their neighboring spins. Notably, we simplify the model by disregarding positional variations in magnetic fields/couplings and directional impacts on interaction strength. This refinement leads to the following Hamiltonian expression:

$$H = -J \sum_{i,j} s_i s_j - B \sum_i s_i$$

$J > 0$ for ferromagnetic (favoring alignment) interaction and $J < 0$ for antiferromagnetic (favoring anti-alignment) interaction. The summation over (i, j) encompasses adjacent pairs within the system.

4.1 1D Ising model

The introduction of a domain wall (a defect in the ordered state) results in an increase in energy by 4J, where J represents the interaction strength. Simultaneously, the change in entropy is denoted as $k_B \ln N$, reflecting the N possible choices for introducing the domain. Consequently, the net change in free energy, ΔF, can be expressed as $\Delta F = 4J - k_B T \ln N$. Importantly, as $N \to \infty$, the net change in free energy is invariably negative. This indicates a preference for a disordered state in the system. Therefore, for an infinite Ising chain in 1D, spontaneous symmetry breaking does not occur. This reasoning can be extended to domains of length L and higher dimensions.

4.2 2D Ising model

In the 2D Ising model and beyond, the scenario shifts. Defect islands can be introduced in two or higher dimensions, incurring energy costs primarily at their boundaries. These costs are proportional to the perimeter of the island, which can be approximated as $L = \epsilon N^2$, where $0 < \epsilon < 1$. In a two-dimensional context, the number of islands scales as $3^{\epsilon N^2}$, while the energy change is represented by $\Delta E = \epsilon 4 J N^2$. This, in turn, leads to a net change in free energy, ΔF, given by $\Delta F = \epsilon 4 J N^2 - k_B T ln(N^2 3^{\epsilon N^2})$. This estimation provides a rough approximation of the critical temperature, $T_c \sim J/k_B$. In essence, the 2D Ising model and higher dimensions exhibit a critical temperature beyond which significant changes in the system's behavior occur.

4.3 Metropolis Monte Carlo

The Metropolis Monte Carlo method involves iteratively updating the spins in the lattice while adhering to a specific acceptance criterion. This ensures that the system explores its energy landscape according to a probabilistic scheme. It is applied on a 2D square-lattice Ising model with periodic boundary conditions.

1. Initialization
   - Initialize a lattice of spins, where each spin is assigned a random value of +1 or -1.
   - Define the lattice size and temperature for the simulation.
2. Iterative Sweeps
   Perform a sweep over the entire lattice. For each spin:
   - Randomly choose a spin site (row, col).
   - Calculate the change in energy (ΔE) that would occur if the selected spin were flipped.
   - If ΔE is negative, accept the flip unconditionally.
   - If ΔE is positive, calculate the acceptance probability using the Metropolis criterion: $P = min(1, e^{(-\Delta E/kT)})$; where k is the Boltzmann constant and T is the temperature.
   - Generate a random number between 0 and 1. If it's less than P, accept the flip; otherwise, reject it.