Classical Monte Carlo Simulations

Hyejin Ju

April 17, 2012

1 Introduction

Why do we need numerics? One of the main goals of condensed matter is to compute expectation values

$$\langle O \rangle = \frac{1}{Z} \mathrm{Tr}\{O\, e^{-\beta \hat{H}}\} \tag{1}$$

and its correlation functions

$$\langle O(r_1) O(r_2) \rangle = \frac{1}{Z} \mathrm{Tr}\{O(r_1) O(r_2)\, e^{-\beta \hat{H}}\}. \tag{2}$$

If the Hamiltonian is some gaussian thing, this is "easy". But in practice, Hamiltonians are much more complicated. While there are some analytical tools to compute these observables, these methods are not exact and need numerical solutions to make the solution complete. In this section, I will concentrate on the classical Monte Carlo methods, in particular, focusing on the 2D classical Ising model.

2 Methods of classical Monte Carlo

Now, let's dive into how Eq. (2) is computed in classical Monte Carlo. Usually, O is some thermodynamic quantity and the "Tr" is some thermal average of the states in the system. This method is based on importance sampling, ergodicity, and detailed balance. I will outline these three things and then go through the 2D classical Ising model.

2.1 Importance sampling

In short, importance sampling is choosing a different probability distribution to minimize the fluctuations of O. Normally, the expectation value, Eq. (2) is taken by integrating O with some probability distribution function P, where P is sharply peaked function in a small region,

$$\langle O \rangle = \int_{-L}^{L} P(x) O(x) dx, \qquad \int_{-L}^{L} P(x) dx = 1. \tag{3}$$

One can discretize this sum by randomly generating a set of M points, $x_1, ..., x_M$ in the range $[-L, L]$, in which case,

$$\langle O \rangle \approx \frac{2L}{M} \sum_{i=1}^{M} P(x_i) O(x_i). \tag{4}$$

The idea is to sample according to some other probability distribution $W(x)$ to minimize the variance of O. Then, Eq. (4) changes to

$$\langle O \rangle \approx \frac{1}{M} \sum_{i=1}^{M} \frac{P(x_i)}{W(x_i)} O(x_i). \tag{5}$$

The variance of O with the distribution $W(x)$ is

$$\sigma_W^2(O) = \int_{-L}^{L} \left(\frac{P(x)}{W(x)} O(x) - \langle O \rangle \right)^2 W(x) dx, \tag{6}$$

which means that we can, in theory, choose a $W(x)$ to minimize the variance. In general, it is very hard to optimize this integral over $W(x)$, but if $P(x)$ has much larger fluctuations than $O(x)$, it is a good solution to use $W(x) = P(x)$. Then, the expected fluctuation of the thermal quantity, O, is

$$\sigma_P^2(O) = \int_{-L}^{L} \left(O(x) - \langle O \rangle\right)^2 W(x) dx. \tag{7}$$

In statistical physics, P is described by the Boltzmann weight, $e^{E/k_B t}$ (from here and on, we'll set $k_B = 1$ and work in units of temperature) while O is typically some low-order polynomial function described by the degrees of freedom. Then, we can take the above prescription of *importance sampling* and sample against the Boltzmann weight.

2.2 Ergodicity and detailed balance

We use some stochastic process to describe the Monte Carlo steps. Let $X_1, X_2, ...$ describe some states in our system. These states constitute a *Markov chain*, where

$$P(X_{i+1}|X_i, X_{i-1}, ..., X_0) = P(X_{i+1}|X_i), \tag{8}$$

or the probability of making a transition to X_{i+1} depends only on its immediate past X_i and not on its history. We also want the sampling to be *ergodic*, which means that any two states must be connected by some path through the state space. This means that the time evolution of all the individual configurations in the ensemble would have the *same* distribution, P, over time.

In classical Monte Carlo, we are typically interested in transition probabilities, i.e. $P(X_i \to X_{i+1})$. Detailed balance governs these probabilities. Let's start by imagining an ensemble of a large number of configurations. If this ensemble is distributed according to P, then the number of configurations $N_0(X_i)$ of configurations X_i in the ensemble is proportional to $P(X_i)$. Then, the number of configurations X_i after updating all the configurations is

$$N_1(X_i) = N_0(X_i) + \sum_{j \neq i} \left[N_0(X_i)P(X_j \to X_i) - N_0(X_i)P(X_i \to X_j)\right], \tag{9}$$

where the second term is the number of configurations changed into and out of X_i. Then, to enforce ergodicity, or to allow the ensemble to remain distributed according to P,

$$\sum_{j \neq i} \left[N_0(X_i)P(X_j \to X_i) - N_0(X_i)P(X_i \to X_j)\right] = 0. \tag{10}$$

Since $N \propto P$, we obtain our equation for *detailed balance*:

$$P(X_j)P(X_j \to X_i) = P(X_i)P(X_i \to X_j). \tag{11}$$

Since, in statistical physics, $P(X_j) = W(X_j)/Z$,

$$\frac{P(X_i \to X_j)}{P(X_j \to X_i)} = \frac{W(C_j)}{W(C_i)}. \tag{12}$$

To get the Metropolis acceptance probability, it is useful to decompose the transition probability:

$$P(X_i \to X_j) = P^{\text{attempt}}(X_i \to X_j)P^{\text{accept}}(X_i \to X_j), \tag{13}$$

where P^{attempt} is equivalent to "proposing" a change from X_i to X_j in your simulation. This is usually a constant (e.g. randomly (by a uniform distribution) picking one out of the four directions in a square lattice). Then, rewriting Eq. (12),

$$\frac{P^{\text{accept}}(X_i \to X_j)}{P^{\text{accept}}(X_j \to X_i)} = \frac{W(C_j)}{W(C_i)}. \tag{14}$$

The Metropolis acceptance probability is:

$$P^{\text{accept}}(X_i \to X_J) = \min\left[\frac{W(C_j)}{W(C_i)}, 1\right]. \tag{15}$$

Another frequently used acceptance probability is a variant of a *heat bath* equation:

$$P^{\text{accept}}(X_i \to X_J) = \frac{W(X_j)}{W(X_i) + W(X_j)}. \tag{16}$$

Now we're ready to study the 2D classical Ising model.

2.3 Example: 2D classical Ising model

Using the convention, $\sigma_i = \pm 1$, the Hamiltonian for the classical Ising model on the 2D square lattice is

$$H = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j - h \sum_i \sigma_i. \tag{17}$$

Usually, a Monte Carlo update from state m to state n occurs via a single spin flip. Let's work out the detailed balance equations.

$$\begin{aligned} E_m &= -J \sum_{\langle ij \rangle} \sigma_i^{(m)} \sigma_j^{(m)} - h \sum_i \sigma_i^{(m)} && (18)\\ W(m) &= e^{-\beta E_m} && (19)\\ P(m \to n) &= \min\left[\frac{W(n)}{W(m)}, 1\right] && (20)\\ &= \min\left[e^{-\beta(E_n - E_m)}, 1\right] && (21) \end{aligned}$$

So, letting $\Delta E = E_n - E_m$, to go from state m to state n, if $\Delta E > 0$ (energy gain), the move occurs with probability $P(m \to n)$ and if $\Delta E < 0$ (state n is in lower energy than m), then the move occurs with probability 1. This code is very simple to write! For the Ising model, Eq. (17), on the square lattice with $N = L \times L$ sites,

- create an array Neighbors of size N, where each entry of Neighbors contains a list of its neighbors. This means that size(Neighbors[i]) = 4 and Neighbors[i] contain a list of i's neighbors.
- create another array, Spin, that contains the spins, ± 1, at each site i.
- Pseudo-code:
- for i = $0 : M$, where M is the number of Monte Carlo steps
  - site = random int($0, N-1$)
  - prob = random real($0, 1$)
  - energy = compute energy(Spin)
  - if(prob < compute weight(site, energy)) Spin[site] = -Spin[site]

This is the basic code to implement the single spin flips. You'll have to implement the functions compute energy (returns double) and compute weight (returns bool), as well as some other functions to measure the observables, but it's quite simple. You can use a different method to implement your own lattice, instead of using this "Neighbors" array.

Now, the question is, are single spin flips ergodic? The answer is: yes, at least above the critical temperature $T_c = 2/\ln(1+2\sqrt{2})$ (yes, this is exact - computed by Onsager). I won't go through this in detail but the basic idea is to look at the autocorrelation function,

$$A_O(\tau) = \frac{\langle O_k O_{k+\tau} \rangle - \langle O_k \rangle^2}{\langle O_k^2 \rangle - \langle O_k \rangle^2} \tag{22}$$

which in principal, measures the number of Monte Carlo steps needed for two states to be considered statistically independent. Above the critical temperature, this autocorrelation time converges. However, near T_c, because of the diverging correlation length (and the system separates into "domains" of up spins and down spins), the autocorrelation time is proportional to $\sim L^2$ with single spin flip updates. This is called *critical slowing down*.

2.4 Cluster updates

We can fix this problem by implementing a method called cluster updates. (N.B. This only works for *second order* transitions. For first order transitions, one needs to implement a method called Wang-Landau algorithm, which creates a "flat histogram" in the energy.) These methods flip entire clusters (of same spins) with some transition weight. There are two algorithms, Swendsen-Wang and Wolff (Phys. Rev. Lett. **62** 361), which are pretty similar. Here's the Wolff algorithm. Consult the references given at the end for more details.

- Choose a site, i, randomly.
- Calculate $p_j = 1 - e^{\beta J \delta_{\sigma_i \sigma_j}}$ for each nearest neighbor. The δ is a Kronecker delta. It only gives a non-zero probability to neighbors with parallel spins.
- For each j, if $p_j < 1$, generate a random number $r \in (0, 1)$.
- if $r < p$, place a "bond" between i and j.
- repeat the previous 3 steps recursively for all bonds in the cluster of spins
- flip the cluster with probability **one**.

In terms of data structures, the best thing to do is to keep information about clusters in some other array. You can use some algorithm, like the Hoshen-Koppelmann or just a simple coloring algorithm (like stacks), to find the clusters. With this modification, Monte Carlo simulations of the 2D Ising model becomes ergodic: all of configuration space will be sampled effectively. Also, larger clusters of spins will be predominantly chosen. There are other methods (loop updates in dimer models/spin ice) that perform some non-local updates. Won't talk about them here!

3 Other numerical methods

More complicated numerical methods are needed to study quantum systems. Here, I give a few examples of other methods available.

3.1 QMC

This method, quantum Monte Carlo (QMC), has been formulated after Feyman's path integral formulations. There are many different variants of QMC, each tailored to some class of problems that one might want to study. *World line methods*– Discretization of imaginary time (Suzuki-Trotter decompositions). *Methods to sample more efficiently with non-local updates*– loop-cluster methods, worm algorithms, etc. *Stochastic series expansion (SSE)*– This is probably the most popular variant of QMC, which is a method of expanding the Hamiltonian to some high power,

$$Z = \sum_{\alpha} \sum_{n=0}^{\infty} (-1)^n \frac{\beta^n}{n!} \sum_{S_n} \langle \alpha | \prod_{p=0}^{n-1} H_p | \alpha \rangle, \tag{23}$$

and is capable of simulating very large systems (for 2D, up to $L = 192$ and sometimes bigger. Depends on how complex the Hamiltonian is...) for $T > 0$. Notice the $(-1)^n$ inside the expansion. This comes from expanding $Z = \mathrm{Tr}\{e^{-\beta \hat{H}}\}$ and is the source of the infamous *sign* problem. For $T = 0$, some projector methods are often used (valence-bond Monte Carlo, etc). For more information, one should consult Anders Sandvik's very comprehensive review: *Computational Studies of Quantum Spin Systems*, arXiv:1101.3281.

3.2 Exact diagonalization

This method is used to exactly diagonalize the Hamiltonian matrix to get the energy spectrum and their corresponding eigenstates. It is powerful because one obtains a complete knowledge of the Hamiltonian. However, it is limited to studying small systems because, for instance, a $s = 1/2$ system contains a total of 2^N states, where N is the number of sites in the system. If, in theory, we had limitless computer power, why bother doing

anything but exact diagonalization? But the fact of the matter is that computer sources are limited and this method is frequently used to test the correctness of other simulations (e.g. QMC) and to learn, in particular, the symmetries of a given Hamiltonian.

As an example, take the $s = 1/2$ XXZ-model

$$H = \sum_{\langle ij \rangle} J_{xy} \left(S_i^x S_j^x + S_i^y S_j^y \right) + J_z S_i^z S_j^z. \tag{24}$$

Then, in the S^z-basis, if we have N sites in our lattice, we can write a state as

$$|v_k\rangle = |1_k\rangle \otimes |2_k\rangle \otimes ... \otimes |N_k\rangle, \tag{25}$$

where $|i_k\rangle = |\uparrow, \downarrow\rangle$. It is easy to see that we have a total of 2^N states. Then, your state space contains a linear combination of all these states, i.e.

$$|\psi_\alpha\rangle = \sum_{k_\alpha} a_{k_\alpha} |v_k\rangle. \tag{26}$$

Then the matrix elements of the Hamiltonian is

$$H_{\alpha\beta} = \langle \psi_\alpha | H | \psi_\beta \rangle, \tag{27}$$

where $\alpha, \beta = 1, ..., 2^N$ and the spin operators can be written as

$$S_i^{x,y,z} \to I \otimes ... \otimes S^{x,y,z} \otimes I \otimes ..., \tag{28}$$

where the operator S is placed in the ith position and $S^{x,y,z} = \frac{\hbar}{2} \sigma^{x,y,z}$. Then, one can build the Hamiltonian through some sparse matrix in MatLab or whatever program you like, and diagonalize this matrix using Lanczos, etc.

3.3 DMRG

The density-matrix renormalization group (DMRG) is very powerful for 1D systems and can simulate 2D systems in moderate sizes. This is a numerical method that efficiently truncates the Hilbert space in such a way to minimize the entanglement. The statistical error is only generated by this truncation error. There are more sophisticated methods that build off of this method, to generalize to larger 2D systems. For more information, one should consult the literature.

- S. White, Phys. Rev. Lett. **69**, 2863 (1992)
- U. Schollwock, Rev. Mod. Phys. **77**, 259 (2005)
- F. Verstraete, et al, Adv. Phys. **57**, 143 (2008)
- ... and references therein.

3.4 And others...

There are many more numerical techniques, such as series expansion techniques, specifically, high-temperature expansions. There are a lot of references... consult J. Oitmaa, et al., *Series Expansion Methods for Strongly Interacting Lattice Models* (Cambridge University Press, Cambridge 2006).

4 References

The following are a list of references I've used to build this summary. Some of the material covered here are directly from Sandvik's notes. These (except for Roger's notes) are all available online!

- Anders Sandvik's notes found on his website. This is an excellent reference!!!

- Matthias Troyer's notes found on his website.
- Roger Melko's notes (via private communication).
- Notes from 2010 Boulder Summer School: *Computational and Conceptual Approaches to Quantum Many-Body Systems*.