Monte Carlo Methods - Sheet 2

Ising Model: Sample generation with the Metropolis algorithm

Tobias Sizmann

May 30, 2022

1 Introduction

Due to the convergence properties Monte-Carlo methods have been proven very useful in estimating high-dimensional integrals and sums. Common discretising integrators tend to fail at high dimensions because the number of volume elements to be calculated and the resulting number of calculation steps N for a fixed allowed error of estimate ϵ increases exponentially with the dimension d, i.e. $\epsilon = \alpha N^d$. Monte Carlo methods always converge with $\epsilon = \alpha N^{-1/2}$. It should be noted, that for most problems α is so large, that a Monte-Carlo approach is not feasable. Only when importance sampling is possible α can be small enough. Usually, importance sampling is enables through a exponential factor in the sum or integral.

Expectation values of a Boltzmann distribution $\sum_i f(i)e^{-\beta E_i}$ in the case of the Ising model suffer a similar illness. The configuration space to be summed over increases exponentially with the number of spin sites. Therefore, small volumes of 100 spin sites are already out of reach of a straight forward numerical caluclation. The Boltzmann factor, however, weights only few configurations with a decently non-zero factor. This enables importance sampling and allows construction of a sample generation algorithm such that α is feasably small. Such an algorithm is the Metropolis algorithm. It allows generation of a Markov chain distributed according to the Boltzmann weights $\exp -\beta E_i$. In the following we will implement this algorithm for a 2D finite quadratic Ising model with periodic boundary conditions.

2 Configuration space and grid layout

The first task will be to find a suitable way to describe configurations. The inital idea is straight forward: Since we are considering a 2D quadratic grid of spins, a configuration may be described by a 2D matrix with entries 1, -1 representing spin up or down. Matrices must be flattened to 1D on the memory and to this one can use the row-major memory layout. In the following i denotes an index of the flattened matrix representation and x, y of the 2D matrix. With L being the extend of the grid and // being integer division, one gets the following transformations between the flattened and 2D representation:

$$i = yL + x$$

$$x = i \bmod L \quad \text{and} \quad y = i//L$$

This construction allows us to represent a configuration in memory. Since the Hamiltonian of the Ising model depends heavily on nearest neighbor interaction it is reasonable to create a lookup table $i \to i_{\text{top}}, i_{\text{right}}, i_{\text{bottom}}, i_{\text{left}}$ to quickly access nearest neighbors of site i. With the prior transformations this can be done in a simple manner. First transform i to (x, y), then calculate the nearest neighbors of site (x, y) according to:

$$(x, y)_{\text{top}} = (x, (y + L - 1) \bmod L)$$

$$(x, y)_{\text{bottom}} = (x, (y + 1) \bmod L)$$

$$(x, y)_{\text{left}} = ((x + L - 1) \bmod L, y)$$

$$(x, y)_{\text{right}} = ((x + 1) \bmod L, y)$$

which respects the periodic boundary conditions. Finally, transform back to the flattened representation. Such a lookup table speeds up sample generation quite a lot in contrast to calculation on the fly, since for each Metropolis sweep neighbors have to be accessed $4L^2$ times.

3 Measurements

Since the memory layout of a configuration has been dealt with, the next step is to implement measurements on such a configuration. In particular, we are interested in energy and magnetisation density. Interestingly we will see that the change in both these quantities can be calculated very efficently during the Metropolis steps. Therefore, an absolute measurement of energy and magnetisation will only be required for initialising these values. For a configuration c in flattened representation, these are given by

$$e(c) = \frac{1}{L^2} \sum_i (c(i)c(i_{\text{right}}) + c(i)c(i_{\text{bottom}})$$

$$m(c) = \frac{1}{L^2} \sum_i c(i)$$

Two remarks are necessary here. First, for the energy only the interaction with two neighbors must be considered, since the interactions in the other directions are dealt with by other center sites. Secondly, we are not yet considering the absolute value of the magnetisation. Otherwise the performance enhancing trick would not work. It is important to note that statistical momenta like the mean should always be calculated over the absolute value of the magnetisation to have a physical interpretation.

4 Metropolis algorithm

The Metropolis algorithm builds the Markov chain. It works in the following way:

1. Consider the last configuration in the Markov chain c and a site i. Calculate the energy difference $\Delta E = E_{\text{New}} - E_{\text{Old}}$ if one would switch the spin on this site. One can quickly see, that ΔE only takes the values -8, -4, 0, 4, 8.

2. Calculate $p = e^{-\beta \Delta E}$. This can be sped up by creating a lookup table for the five possible values of ΔE. Draw a random number d uniformly distributed from 0 to 1.
   - If $d < p$ then reject the change by adding a copy of the prior configuration to the Markov chain: $c_{\text{next}} = c$.

- If $d > p$ then accept the change by adding a copy of the prior configuration to the Markov chain but with flipped spin on site i: $c_{\text{next}} = c$ and $c_{\text{next}}(i) * = -1$ The new energy is then $e + = \Delta E / L^2$ and the new magnetisation is $m + = -2c(i)/L^2$

Since the resulting chains for energy and magnetisation are the only quantities we are interested in, the prior configuration c can be discarded and only the new configuration c_{next} must be saved for the next Metropolis step. One can save the magentisation state internally and append the absolute value of the chain.

5 Specific heat and magnetic susceptibility

After a chain has been generated and thermalized, one can compute second order quantities like specific heat and magnetic susceptibility. In the following t_M denotes the Markov time and $e(t_M)$ and $m(t_M)$ the energy density and absolute magnetisation density at this time step. The specific heat density c and magnetic susceptibility density χ are then given by:

$$c =$$

$$\chi =$$