

The two-dimensional Ising model

Simulation using two Monte Carlo algorithms
Computational Physics, Universität Potsdam

Julius Eberhard

March 6, 2019

1 Introduction

1.1 Ising model

The Ising model, in general, is a model of discrete two-valued variables located on a lattice and interacting with each other under the possible influence of an external field. It is commonly used as a model of ferromagnetism. In its simplest two-dimensional form, spin variables s_i that can take the values ± 1 lie on a square lattice and interact with their direct neighborhood independent of the location within the lattice and without the presence of an external magnetic field. The Hamiltonian of the system then reads

$$H = -J \sum_{\langle ij \rangle} s_i s_j,$$

where i, j denote lattice sites and the first sum is only conducted for distinct pairs of nearest neighbors ($\langle ij \rangle = \langle ji \rangle$ occurs only once in the sum). J is the interaction energy between two spins that are nearest neighbors.

In the present form of the model, a ferromagnetic behavior of the lattice—lowest interaction energy for globally aligned spins, implying the ability for spontaneous magnetization—is observed when $J > 0$.

1.2 Monte Carlo methods

The outline described here essentially follows the explanations in [1].

The simulation of an equilibrated physical system using Monte Carlo methods—as done here—relies on the statistical description of an equilibrium state of the system. Assuming the Ising model to be in thermal equilibrium with a heat reservoir at temperature T , the probability of finding the system in a state with Hamiltonian H_μ is

$$p_\mu = \frac{1}{Z} e^{-\beta E_\mu}, \quad (1)$$

where Z is the canonical partition function, $\beta = (kT)^{-1}$, and E_μ the eigenvalue of H_μ . Here, probability refers to the share of time steps with state μ in an infinitely long simulation, $p_\mu = \lim_{t \rightarrow \infty} w_\mu(t)$, where $w_\mu(t)$ is the weight associated to the state μ at time t . The time evolution of the weights w_μ , which can be understood as non-equilibrium probabilities, follows

$$\dot{w}_\mu = \sum_\nu \left(w_\nu(t) P(\nu \rightarrow \mu) - w_\mu(t) P(\mu \rightarrow \nu) \right).$$

This is a simple rate equation (difference of arrival rate toward μ and departure rate from μ) and depicts a Markov process in which the transition from state μ to state ν occurs with probability $P(\mu \rightarrow \nu)$. Having an *equilibrium* state requires that $\dot{w}_\mu = 0$ and allows w_μ to be replaced by the equilibrium probability p_μ , which implies

$$\sum_\nu p_\mu P(\mu \rightarrow \nu) = \sum_\nu p_\nu P(\nu \rightarrow \mu) \quad \xrightarrow{\sum_\nu P(\mu \rightarrow \nu)=1} \quad p_\mu = \sum_\nu p_\nu P(\nu \rightarrow \mu). \quad (2)$$

The additional condition of *detailed balance*,

$$p_\mu P(\mu \rightarrow \nu) = p_\nu P(\nu \rightarrow \mu), \quad (3)$$

ensures that the method converges to actual equilibrium instead of unintentionally reaching a limit cycle, for which (2) may also be true.

A Monte Carlo method aims at simulating a Markov chain of states which ultimately realizes the equilibrium probability distribution (1). It therefore needs to obey conditions (2) and (3), with the detailed balance relation specified by (1),

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{p_\nu}{p_\mu} = e^{-\beta(E_\nu - E_\mu)}.$$

In order to actually create an algorithm consistent with this condition, it is helpful to split each transition probability into a probability which the algorithm creates a new state κ from ι , $g(\iota \rightarrow \kappa)$, and a probability of accepting the new state, the *acceptance ratio* $A(\iota \rightarrow \kappa)$:

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{g(\mu \rightarrow \nu) A(\mu \rightarrow \nu)}{g(\nu \rightarrow \mu) A(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)}.$$

This distinction allows for tweaking the probabilities associated with the two processes of generation of a state and acceptance of this state separately.

1.3 Metropolis algorithm

First suggested by [2], the Metropolis algorithm comprises the following specifications of the Monte Carlo method:

- (a) A new state is created by flipping only a single spin.
- (b) The probability of the algorithm to create a new state ν from state μ is independent of the choice of ν , thus $g(\mu \rightarrow \nu) = 1/N = g(\nu \rightarrow \mu)$, where N is the number of possible new states created according to (a). This implies that

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)}.$$

- (c) Consistent with (b), the acceptance ratio is chosen to obey

$$A(\mu \rightarrow \nu) = \begin{cases} e^{-\beta(E_\nu - E_\mu)} & \text{if } E_\nu - E_\mu > 0, \\ 1 & \text{else.} \end{cases}$$

2 Implementation

We set $k = 1[E]/[T]$, $J = 1[E]$, where $[\cdot]$ refers to the unit of the respective observable. The Hamiltonian is thus

$$H = - \sum_{\langle ij \rangle} s_i s_j.$$

References

- [1] Newman, M.E.J., Barkema, G.T., 1999. Monte Carlo methods in statistical physics. *Oxford University Press*.
- [2] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E., 1953. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**(6), 1087.