Lecture 7 Metropolis Algorithm *

Tiejun Li

1 Introduction

Metropolis algorithm is one of the Top10 algorithms in 20th century [1]. The aim is to compute the thermodynamic quantities through ensemble average, such as

$$\langle A \rangle = \sum_{\sigma} A(\sigma) P(\sigma) \quad (\text{or } \int_S A(\sigma) P(d\sigma)).$$

where σ represents all possible configurations, and

$$P(\sigma) = \frac{1}{Z_\beta} e^{-\beta H(\sigma)} \qquad \beta = (k_B T)^{-1}.$$

$H(\sigma)$ is the energy function $Z_\beta = \sum_\sigma e^{-\beta H(\sigma)}$ is the partition function, $\beta = (k_B T)^{-1}$.

Intuitively, we could approximate $\langle A \rangle$ by

$$\langle A \rangle \approx \langle A \rangle_N := \frac{1}{N} \sum_{i=1}^{N} A(\sigma_i), \tag{1}$$

where $\sigma_i \sim P(\sigma)$ *i.i.d.*

But the problem is **how to generate** σ_i!

2 Ising Model

Example 1. *(1D Ising model) The magnetization of a ferromagnet may be described essentially by a spin model as shown in Figure 1.*

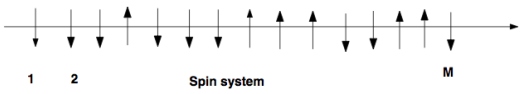

Figure 1: Sketch of 1D Ising model

The macroscopic magnetization $\bar{H}$ can be obtained from ensemble average of all the possible microscopic spin configurations in statistical physics. Consider one microscopic sate with M sites $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_M)$, $\sigma_i = +1$ or -1. If the i-th spin is $\uparrow$, then $\sigma_i = +1$; If the i-th spin is $\downarrow$, then $\sigma_i = -1$.

*School of Mathematical Sciences, Peking University, Beijing 100871, P.R. China

Physically the following quantities are of interest:

1. *Hamiltonian:*
$$H(\sigma) = -J \sum_{<ij>} \sigma_i \sigma_j - h \sum_i \sigma_i,$$
where $\sigma_i = \pm 1$, $< ij >$ means to take sum w.r.t all neighboring spins $|i-j| = 1$, and h is the strength of external magnetic field.

2. *Free energy:*
$$F_M = -\beta^{-1} \ln Z_M, \qquad f_M = \frac{1}{M} F_M$$
where $Z_M = \sum_\sigma \exp\{-\beta H(\sigma)\}$ is the partition function.

3. *Internal energy:*
$$U_M = \langle H(\sigma) \rangle = \sum_\sigma H(\sigma) \frac{\exp\{-\beta H(\sigma)\}}{Z_M} = -\frac{\partial \ln Z_M}{\partial \beta}, \tag{2}$$
Correspondingly define the internal energy per site: $u_M = \frac{1}{M} U_M$.

4. *Specific heat:*
$$C_M = \frac{\partial U_M}{\partial T} = k_B \beta^2 \{\langle H^2(\sigma) \rangle - \langle H(\sigma) \rangle^2\} = k_B \beta^2 \frac{\partial^2 \ln Z_M}{\partial \beta^2}. \tag{3}$$
Correspondingly define the specific heat per site: $c_M = \frac{1}{M} C_M$.

5. *Magnetization:*
$$G_M = \left\langle \sum_i \sigma_i \right\rangle, \qquad g_M = \frac{1}{M} G_M$$

6. *Magnetic susceptibility:*
$$\Phi_M = \frac{\partial G_M}{\partial h} = \beta \mathrm{Var}(G_M), \qquad \phi_M = \frac{1}{M} \Phi_M.$$

For 2D Ising model, if $h = 0$ and $M \to \infty$, the second order phase transition exists for the internal energy u when the temperature varies. If one takes $J = 1$, and the periodic boundary condition is assumed, the critical temperature can be obtained exactly

$$k_B T_c = (\beta_c)^{-1} = \frac{2}{\ln(1 + \sqrt{2})}.$$

And there exists so called spontaneous magnetization for g_M as $M \to \infty$, we have

$$\begin{aligned}
0 < \beta < \beta_c, \quad & g(\beta, h) \to g(\beta, 0) = 0 \ \text{as } h \to 0+, \\
& g(\beta, h) \to g(\beta, 0) = 0 \ \text{as } h \to 0-, \\
\beta > \beta_c, \quad & g(\beta, h) \to g(\beta, +) > 0 \qquad\qquad \text{as } h \to 0+, \\
& g(\beta, h) \to g(\beta, -) = -g(\beta, +) < 0 \ \text{as } h \to 0-.
\end{aligned}$$

For refs, see [2, 3].

How to compute the approximate $k_B T_c$ with computer?

3 Metropolis Algorithm

1. Basic idea:

   The central problem for standard MC(equation (1)) is that it is difficult to generate $\sigma_i \sim P(\sigma)$ *i.i.d.*. To overcome this difficulty, Metropolis algorithm takes an "**iteration**" procedure to produce these random variables. This is very similar with finding a root of a nonlinear equation $f(x) = 0$. There isn't direct method to do this, but if one uses the iteration $x_{k+1} = g(x_k)$ such that $x^* = g(x^*)$ is one root of the equation $f(x) = 0$, one approximate value of x^* will be obtained.

   Metropolis algorithm is also called Markov Chain Monte Carlo method. It sets up a Markov chain by defining a suitable transition probability matrix P, such that the probability density $\frac{1}{Z_M} e^{-\beta H(\sigma)}$ is the only equilibrium distribution of this Markov Chain. If we define π the final Gibbs distribution, then we need

   $$\pi P = \pi.$$

   The single step transition $\nu_{n+1} = \nu_n P$ satisfies similar contraction mapping property as the fixed point iteration. One could obtain the Gibbs distribution from any initial state ν_0. That's the essence of Metropolis algorithm!

2. Physical interpretation(a heuristic observation):

   In statistical physics, the equations (2) and (3) are called ensemble average, which means that the macroscopic state is an average result of all possible microscopic states. Suppose we image large amount of gas molecules in a container, and they are in equilibrium, we can measure the temperature, pressure etc. In the viewpoint of statistical physics, this system corresponds lots of microscopic systems. This is shown schematically in Figure 2.

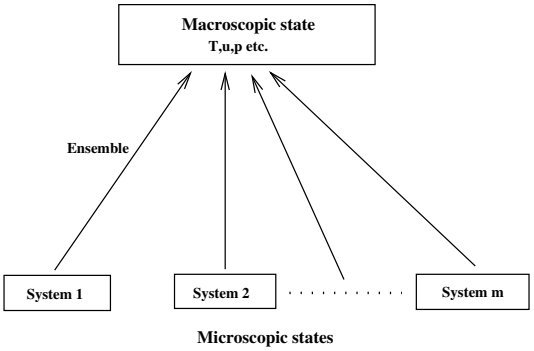

Figure 2: Sketch of ensemble

Though system $1, 2$ and system m are independent evolving systems, the probability distribution anytime for all the m systems keeps the same. Metropolis algorithm takes the following viewpoint: since the macroscopic quantity T, p, ρ is invariant all the time, and the systems are in a dynamic

equilibrium, this means

$$\text{Ensemble average} = \text{Time average},$$

i.e.

$$\langle H(\sigma)\rangle \approx \frac{1}{N}\sum_{i=1}^{N} H(\sigma_i) \tag{4}$$

If we can find an appropriate collision rule for the molecules to sample the state sequence $\{\sigma_i\}_{i=1}^{N}$, we can simulate this process in computer, and find an approximate value of $\langle H(\sigma)\rangle$. This collision rule is the setup of Markov Chain stated as before.

3. Mathematical description:

From the **equilibrium condition** $\pi P = \pi$, where

$$\pi(\sigma) = \frac{1}{Z_M}\exp\{-\beta H(\sigma)\},$$

the choice of matrix P is infinite. One must add more constraints to reduce the degrees of freedom.

Detailed Balance Condition(DBC):

$$\pi(\sigma)P(\sigma \to \sigma') = \pi(\sigma')P(\sigma' \to \sigma).$$

Furthermore one has

$$\frac{P(\sigma \to \sigma')}{P(\sigma' \to \sigma)} = \frac{\pi(\sigma')}{\pi(\sigma)} = e^{-\beta\Delta H},$$

where $\Delta H = H(\sigma') - H(\sigma)$.

Intuitively, if $\Delta H > 0$, set $P(\sigma \to \sigma') = 1$; else set $P(\sigma \to \sigma') = e^{-\beta\Delta H}$. This choice satisfies the DBC. But **how about** $P(\sigma \to \sigma') = 1$**?**

We lose a proposal process!

In general, set

$$P(\sigma \to \sigma') = Q(\sigma \to \sigma')A(\sigma \to \sigma'), \quad \sigma' \neq \sigma$$

and

$$P(\sigma \to \sigma') = 1 - \sum_{\tau \neq \sigma} P(\sigma \to \tau), \quad \sigma' = \sigma$$

where $Q(\sigma \to \sigma')$ is the choosing probability corresponds to **Proposal Matrix**, and $A(\sigma \to \sigma')$ is the acceptance probability corresponds to **Decision Matrix**. In many cases, we take $Q(\sigma \to \sigma') = Q(\sigma' \to \sigma)$, which is a symmetric matrix, and the following two decision strategies:

A. Metropolis algorithm:

$$A(\sigma \to \sigma') = \min(1, e^{-\beta\Delta H}).$$

B. Glauber dynamics:

$$A(\sigma \to \sigma') = (1 + e^{\beta\Delta H})^{-1}.$$

Remark 1. *The proposal matrix Q can also be unsymmetric, then the reversibility is realized in the decision step, which is called Metropolis-Hastings algorithm in statistical learning theory.*

Example 2. *(2D Ising model - $M \times M = M^2$ sites)*

Step1 Generate a proposal σ' from current state σ;

A. Equi-probability proposal:

$$Q(\sigma \to \sigma') = \begin{cases} \frac{1}{N_t - 1} & \sigma \neq \sigma' \\ 0 & \sigma = \sigma' \end{cases}$$

where $N_t = 2^{M^2}$ is the total number of microscopic states.

B. Single flip proposal(usually used):

$$Q(\sigma \to \sigma') = \begin{cases} \frac{1}{M^2} & \sigma \neq \sigma' \text{ only at one site} \\ 0 & \text{otherwise} \end{cases}$$

Step2 Decide to accept or reject the proposal:

Compute $\Delta H(\sigma) = H(\sigma') - H(\sigma)$. If $\Delta H < 0$, accept; else, accept with probability $e^{-\beta \Delta H}$.

4. Algorithm:

Algorithm 1. *(Metropolis Algorithm)*

Step1 Generate the proposal sate σ' from state σ_n according to some strategy;

Step2 Define $\Delta H(\sigma) = H(\sigma') - H(\sigma)$, compute

$$\begin{aligned} A(\sigma, \sigma') &= \min\left\{1, \frac{\pi(\sigma')}{\pi(\sigma)}\right\} \\ &= \begin{cases} 1 & H(\sigma') \leq H(\sigma), \\ \exp(-\beta \Delta H(\sigma)) & \text{otherwise}; \end{cases} \end{aligned}$$

Step3 Generate R.V. $r \sim \mathcal{U}[0, 1]$;

Step4 If $r \leq A(\sigma, \sigma')$, then $\sigma_{n+1} = \sigma'$; else, $\sigma_{n+1} = \sigma_n$, turn to Step1.

5. Theorems:

The proof of the theorems presented below can be found in [4].

Definition 1. *The total variation of a vector is defined as*

$$\|v\| = \sum_i |v_i|.$$

This is the commonly known L^1-norm in matrix theory.

Definition 2. *If there is a natural number τ such that the transition probability matrix P satisfies $P^\tau > 0$(where $A > 0$ means each element of A is greater than 0), this Markov chain is called primitive.*

Theorem 1. *(Ergodic theorem for time-homogeneous Markov Chain)*

There is only one invariant distribution π for a primitive Markov chain, and for any initial distribution ν, one has

$$\lim_{n\to+\infty} \|\nu P^n - \pi\| = 0.$$

Theorem 2. *(WLLN for time-homogeneous Markov Chain)*

Suppose $\{\xi_i\}_{i\geq 1}$ is a realization of a primitive Markov chain, then one has

$$\text{Prob}\left(\left|\frac{1}{n}\sum_{i=1}^{n} f(\xi_i) - \langle f\rangle_\pi\right| > \epsilon\right) \leq \frac{C}{n\epsilon^2},$$

where $\langle f\rangle_\pi$ means the expectation of f w.r.t. the probability π. C depends on R.V. f and Markov chain P.

6. The improved Metropolis-type algorithm in different extensions such as the Swendsen-Wang algorithm, parallel tempering etc. can be referred to [5].

4 Homeworks

- HW1. You want to compute
$$\frac{\int_{-10}^{10} e^{-x^2/a} dx}{\int_{-10}^{10} e^{-x^2/b} dx},$$
where $a,\ b$ are positive constants. Describe a Metropolis Monte Carlo algorithm which does the job; use matlab to test it when $a = 10, b = 12$.

- HW2. Check the detailed balance condition for Metropolis and Glauber dynamics.

- HW3. Check the Markov chain setup by Metropolis and Glauber dynamics for Ising model are both primitive.

References

[1] I. Beichl and F. Sullivan, The Metropolis algorithm, Computing in Science and Engineering 2(2000), 65-69.

[2] L. Onsager, Crystal Statistics. I. A Two Dimensional Model with an Order-Disorder Transition, Phys. Rev. 65(1944), 117-149.

[3] C.N. Yang, The Spotaneous Magnetization of a Two-Dimensional Ising Model, Phys. Rev. 85(1952), 808-816.

[4] G. Winkler, Image Analysis, Random Fields and Dynamic Monte Carlo Methods, Springer-Verlag, Berlin and New York, 1995.

[5] J.S. Liu, Monte Carlo strategies in scientific computing, Springer-Verlag, 2005.