Lecture 8 Multilevel sampling and KMC [*]

Tiejun Li

1 Swendsen-Wang algorithm

For the numerical computation of the Ising model, a commonly used approach is the Gibbs sampling to flip single site at each step. However, this single-site update algorithm slows down rapidly once the temperature is approaching or below the critical value T_0, the so-called "critical slowing down". Swendsen and Wang [6] introduced a powerful clustering algorithm which together with an implementation modification by Wolff [8], amost completely eliminates the critical slowing down. Below explanation to Swendsen-Wang algorithm is from data augmentation viewpoint by Higdon [7].

We have the Gibbs distribution for Ising model

$$\begin{aligned}\pi(\boldsymbol{x}) &\propto \exp\Big\{\beta J \sum_{<i,j>} x_i x_j\Big\} \\ &\propto \prod_{<i,j>} \exp\Big\{\beta J(1 + x_i x_j)\Big\}.\end{aligned}$$

Note that $1 + x_i x_j$ is equal to either 0 or 2. Hence if we introduce an auxiliary variable $\boldsymbol{u}$ on each edge such that

$$\pi(\boldsymbol{x}, \boldsymbol{u}) \propto \prod_{<i,j>} I\Big[0 \leq u_{ij} \leq \exp\{\beta J(1 + x_i x_j)\}\Big].$$

Then the marginal distribution of $\boldsymbol{x}$ is the Gibbs distribution. And under this joint distribution, the conditional distribution $\boldsymbol{u}|\boldsymbol{x}$ is a product of uniform distributions with ranges depending on two neighboring spins. Conversely, the conditional distribution $\boldsymbol{x}|\boldsymbol{u}$ is: if $u_{ij} > 1$, then $x_i = x_j$; otherwise there is no constraint on x_i's. Thus $\boldsymbol{u}$ affects $\boldsymbol{x}$ only through the event $I[u_{ij} > 1]$. Based on the configuration $\boldsymbol{u}$, we cluster those lattice sites according to whether they have a mutual bond ($u_{ij} > 1$). We formulate the following algorithm

Algorithm 1. *Swendsen-Wang algorithm:*

- *Step 1. For a given configuration of the spins, form the bond variable by giving every edge of the lattice $< i, j >$, between two "like spins" ($x_i = x_j$) a bond value of 1 with probability $1 - \exp(-2\beta J)$, and a bond value of 0 otherwise.*

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

- *Step 2. Conditional on the bond variable $\boldsymbol{u}$, update the spin variable $\boldsymbol{x}$ by drawing from $\pi(\boldsymbol{x}|\boldsymbol{u})$, which is uniform on all compatible spin configurations; that is, clusters are produced by connecting neighboring sites with bond value 1. Each cluster is the flipped with probability 0.5.*

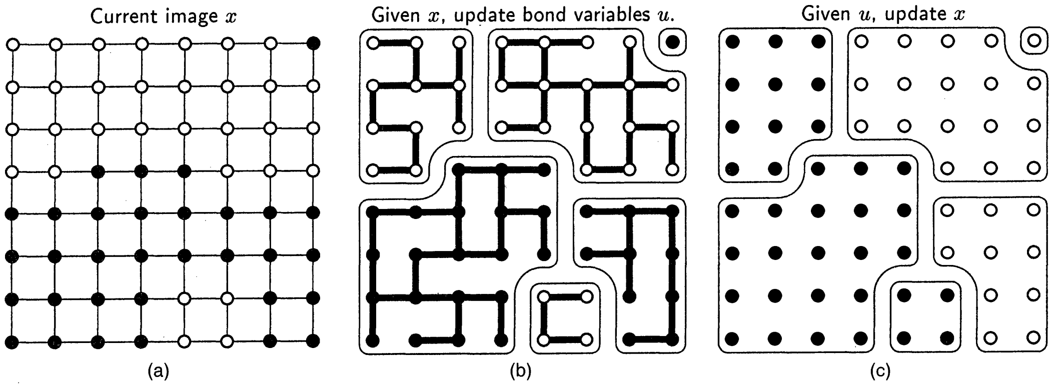

Figure 2. The Swendsen–Wang Algorithm for the Ising Model on the 8 × 8 Lattice. (a) Initial image x and Markov random field graph for $\pi(x)$. (b) Given the current image x, the bond variables u are generated uniformly over the interval (0, $e^{\beta I[x_i=x_j]}$). If $u_{ij} > 1$ (marked by the thick lines), x_i is constrained to equal x_j. These constraints partition the image into clusters of like-colored sites. Clusters induced by this realization of u|x are outlined. The Markov random field graph for x|u differs from that of x, marginally; the auxiliary vector u strengthens the dependence between some neighboring sites, while completely removing it from others. (c) Given the bond variables u, x is now a coarse image of independent clusters. Because there is no external field in this example, each cluster is recolored black or white with probability .5.

2 The modification by Wolff

Wolff introduced a modification for the Swendsen-Wang algorithm, which, although both conceptually and operationally simple, significantly outperforms the SW algorithm.

Algorithm 2. *Wolff's algorithm:*

- *Step 1. For a given configuration $\boldsymbol{x}$, one randomly picks a site, say x_i, and grow recursively from it a "bonded set" C as follows:*
  - *Check all the unchecked neighboring sites of a current set $C^{(old)}$; add a bond between a neighboring site and $C^{(old)}$ the same way as in the Swendsen-Wang algorithm.*
  - *Add those newly bonded neighboring sites to $C^{(old)}$ so as to form a new set $C^{(new)}$.*
  - *Stop the recursion when there is no unchecked neighbor to add; name the final set C.*
- *Flip all the spins corresponding to the sites in set C to their opposites.*

The only difference between Wolff's algorithm and SW is that in each iteration, only one cluster is constructed and all spins in that cluster are changed to their opposite value. This algorithm actually offers a new insight which is different from the one based on the data augmentation. Suppose all of the states in cluster C has spin $+1$ and it has $m+n$ neighboring links among which m are linked with $+1$ spins and n with -1 spins. We have the acceptance probability

$$A_{\text{old}\to\text{new}} = \min\left\{1, \frac{Q_{\text{new}\to\text{old}}\pi_{\text{new}}}{Q_{\text{old}\to\text{new}}\pi_{\text{old}}}\right\}$$

and

$$\frac{\pi_{\text{new}}}{\pi_{\text{old}}} = \frac{e^{\beta J(n-m)}}{e^{\beta J(m-n)}} = e^{2\beta J(n-m)}, \quad \frac{Q_{\text{new}\to\text{old}}}{Q_{\text{old}\to\text{new}}} = \frac{e^{-2\beta Jn}}{e^{-2\beta Jm}} = e^{2\beta J(m-n)}$$

Thus these two probability cancel each other and the proposed change is accepted with probability one.

3 Simulated tempering

To sample the distribution

$$p(x) \propto \exp\Big(-\frac{U(x)}{T}\Big)$$

or compute the ensemble average with the type

$$\langle H\rangle = \int H(x)\frac{1}{Z}\exp\Big(-\frac{U(x)}{T}\Big)dx,$$

one usually apply the Metropolis-Hastings MCMC algorithm. But when the temperature T is very low, that is, we have many high peaks in the pdf $p(x)$, which may cause the acceptance probability small thus decrease the mixing.

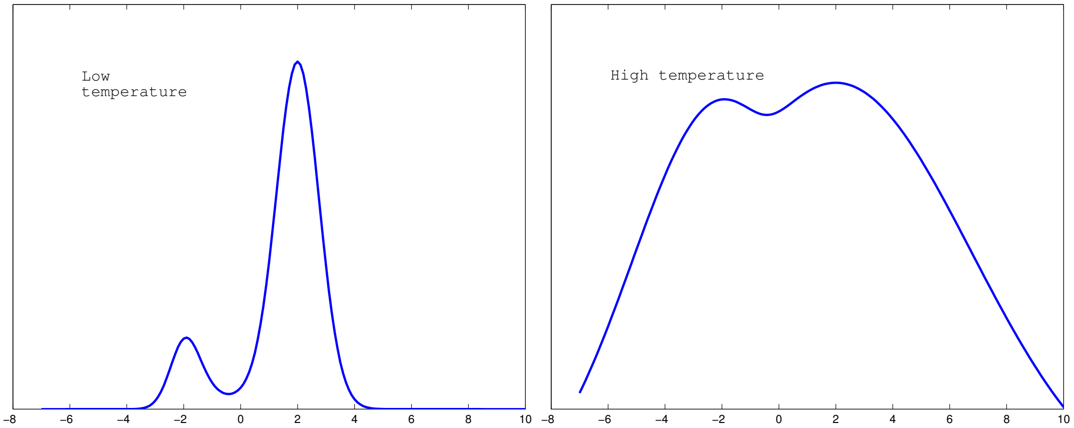

Figure 1: Sketch of the Gibbs distribution at low and high temperature.

In order to let a MCMC scheme move more freely in the sate space, Marinari and Parasi [4] and Geyer and Thompson [3] proposed a data augmentation strategy to increase the mixing, which is called *simulated tempering*. Algorithmically, their basic idea is to extend the state space $x \in \mathcal{X}$ into $(x, i) \in \mathcal{X} \times I$ and perform conditional sampling in this extended space. Physically, to approach the low temperature case, they consider the pdf at the heated temperature, which can give high acceptance ratio for traversing the state space $\mathcal{X}$, and then jump in the different ensembles.

Mathematically, they let

$$I = \{1, 2, \ldots, L\}, \quad T_1 < T_2 < \ldots < T_L$$

and $T_1 = T$, $T_L = T_{\text{high}}$. Then they ask the stationary distribution in the extended space as

$$\pi_{\text{st}}(x, i) \propto \pi_i \exp\Big(-\frac{U(x)}{T_i}\Big),$$

where π_i is called pseudopriors which is set up *a priori.*

From this form, we know the conditional distribution

$$f(x|i) \propto \exp\Big(-\frac{U(x)}{T_i}\Big)$$

which is the standard Gibbs distribution. The marginal distribution

$$f(i) \propto \int \pi_i \exp\Big(-\frac{U(x)}{T_i}\Big)dx = \pi_i Z_i.$$

To make the transition in different ensembles more uniformly, the best choice for the parameter $\pi_i \propto 1/Z_i$. But in the computations, it is not feasible and only updated with the time.

To do the conditional sampling in the extended space, we list a mixture-type transition kernel here.

Algorithm 3 (Simulated tempering)**.** *Mixture type of the simulated tempering.*

- *Step 1. With the current state* $(x_n, i_n) = (x, i)$*, we draw* $u \sim \mathcal{U}[0, 1]$.
- *Step 2. If* $u < \alpha_0$*, we let* $i_{n+1} = i$ *and let* x_{n+1} *be drawn from a MCMC transition* $T_i(x, x_{n+1})$ *that leaves* $f(x|i)$ *invariant (this is also Metropolis-Hastings strategy).*
- *Step 3. If* $u > \alpha_0$*, we let* $x_{n+1} = x$ *and propose a level transtion* $i \to j$*, from a transition function* $\alpha(i, j)$*, and let* $i_{n+1} = j$ *with probability*

  $$\min\Big(1, \frac{\pi_{st}(x, j)\alpha(j, i)}{\pi_{st}(x, i)\alpha(i, j)}\Big).$$

  Otherwise let $i_{n+1} = i$.

A commonly used strategy for $\alpha(i, j)$ is the random walk proposal with reflecting barrier, that is,

$$\alpha(i, i \pm 1) = 1/2, \quad i = 2, \dots, L-1$$

and $\alpha(1, 2) = \alpha(L, L-1) = 1$.

The idea of simulated tempering is further generalized by Liu and Sabatti [5] into the so called "simulated sintering" scheme.

4 Parallel tempering

The pararellel tempering is first proposed by Geyer [2] in 1991. Instead of augmenting $\mathcal{X}$ into $\mathcal{X} \times I$, Geyer suggested directly dealing with the product space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_L$, where the $\mathcal{X}_i$ are identical copies of $\mathcal{X}$, suppose

$$(x_1, \dots, x_L) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_L,$$

we define the stationary distribution

$$\pi_{st}(x_1, \dots, x_L) = \prod_{i \in I} \pi_i(x_i)$$

where $\pi_i(x_i) = 1/Z_i \exp(-U(x_i)/T_i)$ the Gibbs distribution at $T = T_i$. The parallel tempering is run on all of the $\mathcal{X}_i$. An "index swapping" operation is conducted in place of the temperature transition. The algorithm is defined as follows:

Algorithm 4 (Parallel tempering algorithm)**.** *Mixture type transition kernel.*

- *Step 1: Let the current state be* $(x_1^{(n)}, \dots, x_L^{(n)})$*. Draw* $u \sim \mathcal{U}[0,1]$*.*
- *Step 2: If* $u \le \alpha_0$*, we conduct the parallel step. That is, we update each* $x_i^{(n)}$ *to* $x_i^{(n+1)}$ *via their respective MCMCM scheme.*
- *Step 3: If* $u > \alpha_0$*, we conduct the swapping setp. That is, we randomly choose a neighboring pair, say* i *and* $i+1$*, and propose "swapping"* $x_i^{(n)}$ *and* $x_{i+1}^{(n)}$*. Accept this swap with probability*

$$\min\left\{1, \frac{\pi_i(x_{i+1}^{(n)})\pi_{i+1}(x_i^{(n)})}{\pi_i(x_i^{(n)})\pi_{i+1}(x_{i+1}^{(n)})}\right\}.$$

In computations, $T_1 < T_2 < \ldots < T_L$*, and it is very important to choose a proper number of temperature levels. A rough guideline is to choose* T_i *such that*

$$\Big(\frac{1}{T_i} - \frac{1}{T_{i+1}}\Big)|\Delta U| \approx -\log p_a,$$

where $|\Delta U|$ *is the typical energy difference (e.g., the mean energy change under the target distribution) and* p_a *is the lower bound for the acceptance rate.*

Remark 1. *The rationale behind the choice of temperature* T_i *is to make the acceptance probability is relatively large since*

$$\frac{\pi_i(x_{i+1}^{(n)})\pi_{i+1}(x_i^{(n)})}{\pi_i(x_i^{(n)})\pi_{i+1}(x_{i+1}^{(n)})} \sim \exp\left(-\left(\frac{1}{T_i} - \frac{1}{T_{i+1}}\right)\Delta U\right).$$

5 Kinetic Monte Carlo

Kinetic Monte Carlo is also called BKL algorithm [9]. It is widely used in simulating crystal growth.

- Drawbacks of standard MC:

  At the metastable state σ_m, suppose the proposal state is σ', then

  $$r = e^{-\beta \Delta H}, \qquad \Delta H = H(\sigma') - H(\sigma_m).$$

  If $r \ll 1$, rejection occurred very often! The sample sequence will be like

  $$\sigma_m, \sigma_m, \dots, \sigma_m, \sigma_{new} \dots.$$

  That's very inefficient!

  KMC aims to setup a **rejection free** algorithm.

- Generation of new state:

  Consider 2D Ising model: (ten-fold way in BKL algorithm)

  For a given state σ, there are 10 kinds of flips(single flip proposal):

Class	Spin	Number of spins up (nearest neighbors)
1	↑	4
2	↑	3
3	↑	2
4	↑	1
5	↑	0
6	↓	4
7	↓	3
8	↓	2
9	↓	1
10	↓	0

Table 1: Classification of spins in the 10-fold way

There are 10 kinds of flipping probability $P_j = \min(1, \exp(-\beta \Delta H_j)),\ j = 1, \ldots, 10$. Suppose there are n_j sites at j class $j = 1, \ldots, 10$. Define

$$Q_i = \sum_{j=1}^{i} n_j P_j, \quad i = 1, \ldots, 10,$$

then the BKL algorithm is as follows:

Algorithm 5. *(BKL Algorithm)*

Step1 Generate $R \sim \mathcal{U}[0, Q_{10})$;

Step2 Identify $Q_{i-1} \leq R < Q_i, \quad (Q_0 = 0)$;

Step3 Randomly choose one site to flip in class i.

- Time increment between two flips:

  Suppose on the average there is one attempted flip per lattice site in time τ (physical time), (# of sites $= N = M^2$) then

$$\frac{Q_{10}}{N} : \text{ Probability of flipping for a spin (only one) on a given attempt.}$$

Note that the above procedure has homogenized the successful flipping probability to each site. We have the successful flipping probability for one site in unit time

$$\frac{Q_{10}}{N} \Big/ \frac{\tau}{N} = \frac{Q_{10}}{\tau} : \text{ Flip one spin unit time.}$$

Define $P(\Delta t)$ is the probability that no flip occurs before time Δt has elapsed since the previous flip, then

$$P(\Delta t) - P(\Delta t + dt) = P(\Delta t) \cdot \frac{Q_{10}}{\tau} dt,$$

so one has

$$P(\Delta t) = \exp(-\frac{Q_{10}\Delta t}{\tau}), \quad P(0) = 1.$$

i.e. the time increment

$$\Delta t = -\frac{\tau}{Q_{10}} \ln R, \quad R \sim \mathcal{U}[0,1],\ 0 \leq R \leq 1.$$

Remark 2. *Essence: The Markov chain in Metropolis algorithm is some skeleton of a continuous time Q-process with Q-matrix*

$$q_{ij} = 1_{Q_{ij}} A_{ij} \tag{1}$$

in KMC, where $1_{Q_{ij}}$ is defined as $1_{Q_{ij}} = 1$ if $Q_{ij} > 0$ and $1_{Q_{ij}} = 0$ otherwise. A_{ij} is the acceptance probability P shown above.

Remark 3. *If one applies KMC to compute the ensemble average, the time increment occurs as a weight for different states.*

Remark 4. *KMC can simulate the non-equilibrium process such as crystal growth, but the connection between the process and the real physics is not clear!*

6 Homeworks

1. Write down the transition kernel of simulated tempering method (transition probability matrix in the case of discrete state Markov chain).

2. Write down the transition kernel of parallel tempering method (transition probability matrix in the case of discrete state Markov chain).

References

[1] J. Liu, Monte Carlo strategies in scientific computing, Springer-Verlag, 2001.

[2] C.J. Geyer, Markov chain Monte Carlo maximum likelihood, in E. Keramigas (ed.), COmputing Science and statistics: The 23rd symposium on the interface, Interface Foundation, Fairfax, 156-163, 1991.

[3] C.J. Geyer and E.A. Thompson, Annealing Markov Chain Monte Carlo with applications to ancestral inference, J. Amer. Stat. Assoc. 431 (1995), 909-919.

[4] E. Marinari and G. Parisi, Simulated tempering: a new Monte Carlo scheme, Europhys. Lett. 19 (1992), 451-458.

[5] J. Liu and C. Sabatti, Simulated sintering: Markov chain Monte Carlo with spaces of varying dimensions, in J. Bernardo, J. Berger, A. Dawid and A. Smith (eds.), Bayesian statistics, 6, 386-413, Oxford University Press, New York, 1998.

[6] R.H. Swendsen and J.S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, Phys. Rev. Lett. 58 (1987), 86-88.

[7] D.M. Higdon, Auxiliary variable methods for Markov Chain Monte Carlo with applications, J. Amer. Stat. Assoc. 93 (1998), 585-595.

[8] U. Wolff, Collective Monte Carlo updating for spin systems, Phys. Rev. Lett. 62 (1989), 361-364.

[9] A.B. Bortz, M.H. Kalos and J.L. Lebowitz, J. Comp. Phys. 17(1975), 10-18.