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], almost 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

$$
\pi(x) \propto \exp\left\{ \beta J \sum_{<i,j>} x_i x_j \right\} \\
\propto \prod_{<i,j>} \exp\left\{ \beta J (1 + x_i x_j) \right\}.
$$

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

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

Then the marginal distribution of $x$ is the Gibbs distribution. And under this joint distribution, the conditional distribution $u|x$ is a product of uniform distributions with ranges depending on two neighboring spins. Conversely, the conditional distribution $x|u$ is: if $u_{ij} > 1$, then $x_i = x_j$; otherwise there is no constraint on $x_i$’s. Thus $u$ affects $x$ only through the event $I[u_{ij} > 1]$. Based on the configuration $u$, 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.
• **Step 2.** Conditional on the bond variable $u$, update the spin variable $x$ by drawing from $\pi(x|u)$, which is uniform on all compatible spin configurations; that is, clusters are produced by connecting neighboring sites with bond value 1. Each cluster is flipped with probability 0.5.

![Image](image.png)

Figure 2. The Swendsen–Wang Algorithm for the Ising Model on the $8 \times 8$ Lattice. (a) Initial image $x$ and Markov random field graph for $x(x)$. (b) Given the current image $x$, the bond variables $u$ are generated uniformly over the interval $(0, \varphi_{\text{SW}}(n \times n))$. If $u > 1$ (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 $x|x$ 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 0.5.

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 $x$, 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 $C^{(\text{old})}$; add a bond between a neighboring site and $C^{(\text{old})}$ the same way as in the Swendsen-Wang algorithm.
  
  – Add those newly bonded neighboring sites to $C^{(\text{old})}$ so as to form a new set $C^{(\text{new})}$.
  
  – 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 the cluster $C$ we have grown has $m + n$ neighboring links among which $m$ are linked with +1 spins and $n$ with −1 spins. The probability ratio of flipping $C$ is $\exp(2\beta J(n - m))$. One can also verify that the ratio of proposal transition (to build the block $C$) is $\exp(2\beta J(m - n))$. These two cancel each other and, thus, the proposed change is accepted with probability one.

2
3 Simulated tempering

To sample the distribution

\[ p(x) \propto \exp \left( -\frac{U(x)}{T} \right) \]

or compute the ensemble average with the type

\[ \langle H \rangle = \int H(x) \frac{1}{Z} \exp \left( -\frac{U(x)}{T} \right) dx, \]

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.

![Sketch of the Gibbs distribution at low and high temperature.](image)

In order to let a MCMC scheme move more freely in the state 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

\[ \pi_{\text{st}}(x, i) \propto \pi_i \exp \left( -\frac{U(x)}{T_i} \right), \]

where \( \pi_i \) is called pseudopriors which is set up a priori.

From this form, we know the conditional distribution

\[ f(x|i) \propto \exp \left( -\frac{U(x)}{T_i} \right) \]
which is the standard Gibbs distribution. The marginal distribution

\[ f(i) \propto \int \pi_i \exp \left( -\frac{U(x)}{T_i} \right) dx = \pi_i Z_i. \]

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 \( u \sim U[0, 1] \).
- **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 transition \( i \rightarrow j \), from a transition function \( \alpha(i, j) \), and let \( i_{n+1} = j \) with probability

\[ \min \left( 1, \frac{\pi_{st}(x, j)\alpha(j, i)}{\pi_{st}(x, i)\alpha(i, j)} \right). \]

Otherwise let \( i_{n+1} = i \).

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

\[ \alpha(i, i+1) = 1/2, \quad i = 2, \ldots, L - 1 \]

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 parallel 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

\[ (x_1, \ldots, x_L) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_L, \]

we define the stationary distribution

\[ \pi_{st}(x_1, \ldots, x_L) = \prod_{i \in I} \pi_i(x_i) \]

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 \((x_{1}^{(n)}, \ldots, x_{L}^{(n)})\). Draw \(u \sim U[0, 1]\).
- **Step 2:** If \(u \leq \alpha_0\), we conduct the parallel step. That is, we update each \(x_i^{(n)}\) to \(x_i^{(n+1)}\) via their respective MCMC scheme.
- **Step 3:** If \(u > \alpha_0\), we conduct the swapping step. 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 \((1/T_i - 1/T_{i+1})|\Delta U| \approx -\log p_a\), 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.

5 Equi-Energy sampler

In Monte Carlo statistical inference one crucial task is to obtain samples from a given distribution, often known up to a normalizing constant. Let \(\pi(x)\) denote the target distribution and let \(h(x)\) be the associated energy function. Then \(\pi(x) \propto \exp(-h(x))\). For simple problems, the famous Metropolis-Hastings (MH) algorithm, which employs a local Markov chain move, could work. However, if \(\pi(x)\) is multimodal and the modes are far away from each other, which is often the case for practical multidimensional distributions, algorithms relying on local moves such as the MH algorithm or the Gibbs sampler can be easily trapped in a local mode indefinitely, resulting in inefficient and even unreliable samples.

Define the Boltzmann average, also known as the thermal average in the physics literature,
\[
\mu_g(T) = \int g(x) \exp(-h(x)/T)/Z(T)dx
\]
and the conditional expectation of a state function \(g(x)\) given an energy level \(u\) is called its microcanonical average:
\[
\nu_g(u) = \mathbb{E}(g(X)|h(X) = u).
\]

Without loss of generality, we assume that the minimum energy of the system \(u_{\text{min}} = 0\). The following result follows easily from these definitions.

**Lemma 1.** Let \(\beta = 1/T\) denote the inverse temperature so that the Boltzmann averages and partition function are indexed by \(\beta\) as well as by \(Z\); then
\[
\mu_g(\beta^{-1})Z(\beta^{-1}) = \int_{0}^{\infty} \nu_g(u)\Omega(u)e^{-\beta u}du.
\]
In particular, the partition function \(Z(\beta^{-1})\) and the density of states \(\Omega(u)\) form a Laplace transform pair.
This lemma suggests that the Boltzmann averages and the partition function can be obtained through Monte Carlo algorithms designed to compute the density of states and microcanonical averages. We hence refer to such algorithms as energy-domain algorithms.

The EE sampler aims to overcome this difficulty by working on the energy function directly. First, a sequence of energy levels is introduced:

$$H_0 < H_1 < H_2 < \cdots < H_K < H_{K+1} = \infty,$$

such that $H_0$ is below the minimum energy, $H_0 \neq \inf_x h(x)$. Associated with the energy levels is a sequence of temperatures

$$1 = T_0 < T_1 < \cdots < T_K.$$ 

The EE sampler considers $K + 1$ distributions, each indexed by a temperature and an energy truncation. The energy function of the $i$-th distribution $\pi_i$ ($0 \leq i \leq K$) is $h_i(x) = 1/T_i(h(x) \lor H_i)$, that is, $\pi_i(x) \propto \exp(-h_i(x))$. Clearly $\pi_0$ is the initial distribution of interest. The EE sampler employs the other $K$ chains to overcome local trapping, because for large $i$ the energy truncation and the high temperature on $h_i(x)$ flatten the distribution $\pi_i(x)$, making it easier to move between local modes. The quick mixing of chains with large $i$ is utilized by the EE sampler, through a step termed the equi-energy jump, to help sampling from $\pi_i$ with small $i$, where the landscape is more rugged.

We partition the state space $\mathcal{X}$ according to the energy levels, $\mathcal{X} = \bigcup_{j=0}^K D_j$, where

$$D_j = \{ x : h(x) \in [H_j, H_{j+1}) \}, 0 \leq j \leq K,$$

are the energy sets, determined by the energy sequence (4). For any $x \in \mathcal{X}$, let $I(x)$ denote the partition index such that $I(x) = j$, if $x \in D_j$, that is, if $h(x) \in [H_j, H_{j+1})$.

The EE sampler begins from an MH chain $X^{(K)}$ targeting the highest-order distribution $\pi_K$. After an initial burn-in period, the EE sampler starts constructing the $K$-th order energy rings $\hat{D}_j^{(K)}$ by grouping the samples according to their energy levels; that is, $\hat{D}_j^{(K)}$ consists of all the samples $X_n^{(K)}$ such that $I(X_n^{(K)}) = j$. After the chain $X^{(K)}$ has been running for $N$ steps, the EE sampler starts the second highest-order chain $X^{(K-1)}$ targeting $\pi_{K-1}$, while it keeps on running $X^{(K)}$ and updating $\hat{D}_j^{(K)}$ ($0 \leq j \leq K$). The chain $X^{(K-1)}$ is updated through two operations: the local move and the equi-energy jump. At each update a coin is flipped; with probability $1 - p_{ee}$ the current state $X_n^{(K-1)}$ undergoes an MH local move to give the next state $X_{n+1}^{(K-1)}$, and with probability $p_{ee}$, $X_n^{(K-1)}$ goes through an equi-energy jump. In the equi-energy jump, a state $y$ is chosen uniformly from the highest-order energy ring $\hat{D}_j^{(K)}$ indexed by $j = I(X_n^{(K-1)})$ that corresponds to the energy level of $X_n^{(K-1)}$ [note that $y$ and $X_n^{(K-1)}$ have similar energy level, since $I(y) = I(X_n^{(K-1)})$ by construction]; the chosen $y$ is accepted to be the next state $X_{n+1}^{(K-1)}$ with probability

$$\min\left(1, \frac{\pi_{K-1}(y)\pi_K(X_n^{(K-1)})}{\pi_{K-1}(X_n^{(K-1)})\pi_K(y)}\right);$$

if $y$ is not accepted, $X_{n+1}^{(K-1)}$ keeps the old value $X_n^{(K-1)}$. After a burn-in period on $X^{(K-1)}$, the EE sampler starts the construction of the second highest-order [i.e., $(K - 1)$-st order] energy rings $\hat{D}_j^{(K-1)}$ in much the
same way as the construction of $\tilde{D}_j^{(K)}$, that is, collecting the samples according to their energy levels. Once the chain $X^{(K-1)}$ has been running for $N$ steps, the EE sampler starts $X^{(K-2)}$ targeting $\pi_{K-2}$ while it keeps on running $X^{(K-1)}$ and $X^{(K)}$. Like $X^{(K-1)}$, the chain $X^{(K-2)}$ is updated by the local MH move and the equi-energy jump with probabilities $1 - p_{ee}$ and $p_{ee}$, respectively. In the equi-energy jump, a state $y$ uniformly chosen from $\tilde{D}^{(K-1)}_{I(X^{(K-2)})}$, where $X_n^{(K-2)}$ is the current state, is accepted to be the next state $X_{n+1}^{(K-2)}$ with probability

$$\min\left(1, \frac{\pi_{K-2}(y)\pi_{K-1}(X_n^{(K-2)})}{\pi_{K-2}(X_n^{(K-2)})\pi_{K-1}(y)}\right).$$

The EE sampler thus successively moves down the energy and temperature ladder until the distribution $\pi_0$ is reached. Each chain $X^{(i)}, 0 \leq i < K$, is updated by the equi-energy jump and the local MH move; the equi-energy move proposes a state $y$ uniformly chosen from the energy ring $\tilde{D}^{(i+1)}_{I(X_n^{(i)})}$ and accepts the proposal with probability

$$\min\left(1, \frac{\pi_i(y)\pi_{i+1}(X_n^{(i)})}{\pi_i(X_n^{(i)})\pi_{i+1}(y)}\right).$$

At each chain $X^{(i)}$, the energy rings $\hat{D}^{(i)}_j$ are constructed after a burn-in period, and will be used for chain $X^{(i-1)}$ in the equi-energy jump.

6 Homeworoks

For 2D Ising model, choose one algorithm below to implement and compare it with the classical Metropolis algorithm:

- Swendsen-Wang algorithm
- Wolff’s modification
- Simulated tempering

References


