7

Cluster Algorithms for the Ising Model

7.1 Ising and Potts Model Revisit

In Section 1.3, we introduced the Ising model, which is used by physicists to model the magnetization phenomenon and has been studied extensively in statistical physics literature. A closely related model is the *Potts model*. Similar to the Ising model, the Potts model is also defined on the lattice space $\mathcal{L}$ with configurations $\mathbf{x} = (x_l,\ l \in \mathcal{L})$, where

$$\mathcal{L} = \{l = (l_1, l_2) \quad \text{for} \quad l_1, l_2 = 1, \ldots, N\}.$$

Different from the Ising model, each x_l in the Potts model can take values in an alphabet of size q: $\{1, \ldots, q\}$. The potential energy function for the Potts model can be written as

$$H(\mathbf{x}) = -J \sum_{i \sim j} \delta_{x_i x_j} - \sum_j h_j(x_j),$$

where δ_{ab} is the Kronecker δ-function, equaling 1 when $a = b$ and 0 otherwise. The symbol $i \sim j$ indicates that i and j are neighbors in the lattice space. The function $h_j(\ )$ represents an outside magnet field. The distribution of interest is then the Boltzmann distribution

$$\pi(\mathbf{x}) \propto \exp\{-\beta H(\mathbf{x})\}.$$

Here, $\beta = (kT)^{-1}$ with the Boltzmann constant k and the absolute temperature T. When $q = 2$, the Potts model is equivalent to the Ising model

in that

$$H(\mathbf{x}) = -\frac{1}{2}J\sum_{i\sim j} 2\left(\delta_{x_i x_j} - \frac{1}{2}\right) - \sum_{i\sim j}\frac{1}{2}J - \sum_j h_j x_j.$$

For brevity of presentation, in this section we focus on the simulation of the Ising model under the setting $h_j \equiv 0$, which is supposedly the most interesting setting (Newman and Barkema 1999).

An important problem to physicists is the *phase transition phenomena* for such a model; that is, it is observed that when the temperature is high, all the spins behave nearly independently (no long-range correlation), whereas when temperature is below a *critical temperature* c_0, all the spins tend to stay the same (i.e., cooperative performance).

The standard Metropolis algorithm can be easily applied to simulate from this model: First, we randomly pick a spin and turn its value x_l to the opposite $-x_l$; then, we compute the Metropolis ratio to decide whether to accept such a move. However, this single-site update algorithm slows down rapidly once the temperature is approaching or below the critical value c_0, the so-called "critical slowing down." Swendsen and Wang (1987) introduce a powerful clustering algorithm which, together with an implementation modification of Wolff (1989), almost completely eliminates the critical slowing down.

7.2 The Swendsen-Wang Algorithm as Data Augmentation

Conceptually, we can think of the Swendsen-Wang algorithm as a data augmentation scheme (Edwards and Sokal 1988, Higdon 1998, Tanner and Wong 1987). To be precise, we consider augmenting the space of spins by a "bond variable" $\boldsymbol{u} = (u_{l\sim l'})$ with its component variable $u_{l,l'}$ sitting on every edge of the lattice and taking values in $[0, e^{2\beta J}]$:

$$\begin{aligned}\pi(\mathbf{x}) &\propto \exp\left\{\beta J\sum_{l\sim l'} x_l x_{l'}\right\}\\ &\propto \prod_{l\sim l'}\exp\{\beta J(1+x_l x_{l'})\}\end{aligned}$$

Note that $1 + x_l x_{l'}$ is equal to either 0 or 2. Hence, if we introduce an auxiliary variable $\boldsymbol{u}$ such that

$$\pi(\mathbf{x},\boldsymbol{u}) \propto \prod_{l\sim l'} I[0 \le u_{l,l'} \le \exp\{\beta J(1+x_l x_{l'})\}],$$

then the marginal distribution of $\mathbf{x}$ is the desirable distribution. Clearly, under this joint distribution, the conditional distribution $[\boldsymbol{u} \mid \mathbf{x}]$ is a product

of uniform distributions with ranges depending on two neighboring spins. Conversely, the conditional distribution of $\mathbf{x}$ given $\boldsymbol{u}$ is also easy to figure out: If $u_{l,l'} > 1$, then x_l must be equal to $x_{l'}$; otherwise there is no constraint on x_l's. Thus, $\boldsymbol{u}$ affects $\mathbf{x}$ only through the event $I[u_{l,l'} > 1]$. Based on the configuration of $\boldsymbol{u}$, we "cluster" those lattice sites according to whether they have a "mutual bond" [i.e., whether $u_{l,l'} > 1$]. Then, all the x_l whose site l belongs to a common cluster should take identical value. Conditional on the clusters, every configuration that does not violate the cluster homogeneity is equally likely.

Therefore, we can produce another augmented model that only uses an auxiliary *bonding* variables, $\boldsymbol{b} = (b_{l\sim l'})$, to indicate whether $u(l,l') > 1$ holds. More precisely, we define

$$b_{l,l'} = I[u_{l,l'} > 1].$$

Then, the corresponding augmented model is

$$\pi(\mathbf{x},\boldsymbol{b}) \propto \prod_{x_l = x_{l'}} \left\{1 + b_{l,l'}(e^{2\beta J} - 1)\right\},$$

and $\boldsymbol{b}_{l,l'} = 0$ whenever $x_l \neq x_{l'}$. The clustering of the spins can be achieved by connecting all those neighboring sites whose bond value is 1. Conditional on the realization of $\boldsymbol{b}$, the spin value of one cluster is independent of those of other clusters. The algorithm of Swendsen and Wang (1987) is then a data augmentation scheme that iterates between sampling from $\pi(\boldsymbol{b} \mid \mathbf{x})$ and $\pi(\mathbf{x} \mid \boldsymbol{b})$.

Swendsen-Wang (SW) Algorithm

- For a given configuration of the spins, form the bond variable by giving every edge of the lattice, $\langle l, l'\rangle$, between two "like spins" (i.e., $x_l = x_{l'}$) a bond value of 1 (i.e., $b_{l,l'} = 1$) with probability $e^{-2\beta J}$, and a bond value of 0 otherwise.
- Conditional on the bond variable $\boldsymbol{b}$, update the spin variable $\mathbf{x}$ by drawing from $p(\mathbf{x} \mid \boldsymbol{b})$, which is uniform on all compatible spin configurations; that is, clusters are produced by connecting neighboring sites with a bond value 1. Every cluster is then flipped with probability 0.5.

7.3 Convergence Analysis and Generalization

Based on Theorem 6.6 and theory in 13, the convergence rate of this algorithm is characterized by the *maximal correlation* between $\mathbf{x}$ and $\boldsymbol{b}$ (Liu et al. 1994, Liu, Wong and Kong 1995), which is defined as

$$\rho = \sup_{h\in L^2} \frac{\mathrm{var}[E\{h(\boldsymbol{b}) \mid \mathbf{x}\}]}{\mathrm{var}\{h(\boldsymbol{b})\}} \tag{7.1}$$

under the equilibrium distribution. Thus, we can obtain a lower bound of the algorithm's convergence rate by computing $\text{var}[E\{h(\boldsymbol{b}) \mid \mathbf{x}\}]$ for a particular test function. As in Li and Sokal (1989), we can choose $h(\boldsymbol{b}) = \sum b_{l,l'}$. Then, by denoting $p_a = 1 - e^{-2\beta J}$, it is easy to see that

$$E[h(\boldsymbol{b}) \mid \mathbf{x}] = p_a \sum_{l\ l'} \delta_{x_l = x_{l'}},$$

$$\text{var}[h(\boldsymbol{b}) \mid \mathbf{x}] = p_a(1-p_a) \sum_{l\ l'} \delta_{x_l = x_{l'}}.$$

If we let $U(\mathbf{x}) = \sum_{l\ l'} \delta_{x_l = x_{l'}}$, then $E[U(\mathbf{x})]$ is related to the mean energy and $\text{var}[U(\mathbf{x})]$ is proportional to the specific heat. From definition (7.1) and the theory in Section 13.2.1, we have

$$\lambda_2 = \rho \geq \frac{\text{var}\{E[h(\boldsymbol{b}) \mid \mathbf{x}]\}}{\text{var}[h(\boldsymbol{b})]} = \frac{p_a^2 \text{var}(U)}{p_a^2 \text{var}(U) + p_a(1-p_a)E(U)},$$

where λ_2 is the second largest eigenvalue of the transition operator of this data augmentation chain and ρ is as defined in (7.1). This is the key inequality for Li and Sokal (1989) to derive an approximate bound on the critical exponents of the SW algorithm.

A more general formulation of the SW algorithm is given by Edwards and Sokal (1988) and described also in Higdon (1998). In particular, Higdon (1998) successfully applied this approach to tackle a class of image analysis problems. Suppose the target distribution has a form

$$\pi(\mathbf{x}) \propto \pi_0(\mathbf{x}) \prod_k f_k(\mathbf{x}).$$

One can introduce an augmented model with $\boldsymbol{u} = (u_k)$ so that

$$\pi(\mathbf{x}, \boldsymbol{u}) \propto \pi_0(\mathbf{x}) I[0 \leq u_k \leq f_k(\mathbf{x})]. \tag{7.2}$$

Then, a data augmentation scheme can be formally implemented by iterating (a) draw $\boldsymbol{u} \sim [\boldsymbol{u} \mid \mathbf{x}]$ and (b) draw $\mathbf{x} \sim [\mathbf{x} \mid \boldsymbol{u}]$. Although step (a) is trivial, step (b) may not be achievable in problems other than the Ising or Potts models. This strategy is also referred to as the *slice sampler* (Section 6.3.1).

A *partial decoupling* method was also given by Higdon (1998) in which he replaces the expression (7.2) by

$$\pi(\mathbf{x}, \boldsymbol{u}) \propto \pi_0(\mathbf{x}) \prod_k f_k(x)^{1-\delta_k} I[0 \leq u_k \leq f_k(\mathbf{x})^{\delta_k}].$$

One can iterate $[\boldsymbol{u} \mid \mathbf{x}]$ and $[\mathbf{x} \mid \boldsymbol{u}]$ as in the previous case. The partial decoupling method is potentially useful when one does not have the nice symmetry as in the Ising or Potts models, which is typically the case in statistical image analysis (a likelihood term will destroy symmetry in the model).

7.4 The Modification by Wolff

Wolff (1989) introduced a modification for the Swendsen-Wang algorithm, which, although both conceptually and operationally simple, significantly outperforms the SW algorithm.

Wolff's Algorithm

- For a given configuration $\mathbf{x}$, one randomly picks a site, say x_l, 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 (no random sampling here).

The only difference between this algorithm and the SW algorithm is that in each iteration, only *one* cluster is constructed and *all* spins in that cluster are changed to their opposite value. However, the algorithm provides a new insight that is different from the one based on data augmentation. Consider the "move" in the Wolff algorithm from a Metropolis algorithm viewpoint. 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. Thus, if the current state of C is all $+1$, by flipping the whole cluster of C to -1, the probability ratio (of the new to old) is $e^{2\beta J(n-m)}$. Now, consider the process of building C (i.e., the proposal). This proposal, in comparison with the reverse proposal, can be viewed as "breaking bonds" along the edge of the cluster C. Since C starts with all $+1$, the probability of breaking m bonds is $e^{-2\beta Jm}$. To propose back, one needs to break n bonds which has a probability $e^{-2\beta Jn}$. Thus, the ratio of the proposal transitions is

$$\frac{T(\mathbf{x}^{(\text{new})} \to \mathbf{x}^{(\text{old})})}{T(\mathbf{x}^{(\text{old})} \to \mathbf{x}^{(\text{new})})} = \frac{\exp\{-2\beta Jn\}}{\exp\{-2\beta Jm\}} = \exp\{2\beta J(m-n)\}.$$

This ratio cancels the probability ratio and, thus, the proposed change is accepted with probability one.

7.5 Further Generalization

There is no essential reason for restricting the growth of the cluster to be among those "like spins." More generally, Niedermayer (1988) suggests

that one can allow "bonds" to link neighboring spins of opposite values. Let p_a be the probability of growing a bond between l and l' when $x_l = x_{l'}$, and let p_b be that when $x_l \neq x_{l'}$. After growing the cluster C, we can flip every spin in C to its opposite. First, it is not difficult to see that such a "transition rule" is completely symmetric for the *interior* or the cluster; that is, the energy difference between the two states, $\mathbf{x}_{\text{old}} = (\mathbf{x}_C, \mathbf{x}_{[-C]})$ before the move and $\mathbf{x}_{\text{new}} = (-\mathbf{x}_C, \mathbf{x}_{[-C]})$ after the move, is equal to the energy difference of the two states at the boundary. Now, suppose there are m same-spin links and n different-spin links between C and its complement $\bar{C}$. Then, the transition ratio is

$$\frac{T(\mathbf{x}^{(\text{new})} \to \mathbf{x}^{(\text{old})})}{T(bx^{(\text{old})} \to \mathbf{x}^{(\text{new})})} = \frac{(1-p_a)^m(1-p_b)^n}{(1-p_a)^n(1-p_b)^m} = \left(\frac{1-p_a}{1-p_b}\right)^{m-n}.$$

Therefore, we can choose p_a and p_b so as to cancel the transition ratio with the probability ratio, $e^{2\beta J(n-m)}$, achieving a no-rejection transition in the Metropolis algorithm framework. It is of interest to see if this type of cluster-growth method can be used more generally in statistical computation. One area that might benefit from the method is the statistical image analysis, as shown in Higdon (1998).

7.6 Discussion

The auxiliary variable approach discussed in this chapter seems to be at odds with the theory presented in Section 6.7. This "apparent" conflict seems to suggest that adding "decoupling" variables does not necessarily help in improving convergence rate of the sampler unless the system possesses a special symmetry structure. For example, in the bivariate Gaussian inference problem (Sections 2.2 and 6.7), parameter Σ serves as a "decoupling" variable; that is, given Σ, all the missing components are mutually independent. On the other hand, all the missing components are dependent of each other if we integrate out Σ, as suggested by the collapsing theorem of Section 6.7. Our numerical results clearly showed that this "decoupling" really did not improve convergence rate and is more time-consuming for each iteration. A similar phenomenon was also observed for the Gibbs motif finding algorithm (Section 6.5).

Generally, the Gibbs sampler itself does not specify exactly how the random variable should be augmented or partitioned. This is a decision that users have to make and is where they can apply their ingenuity. There are two conflicting criteria that a good Gibbs sampler algorithm has to meet: (a) Drawing one component conditional on the others is computationally simple; (b) the Markov chain induced by the Gibbs sampler with such partitioning components has to converge reasonably fast to its equilibrium distribution. For example, drawing the variables jointly with no partitioning

at all is optimal for convergence, but it is formidable and is the reason why the Gibbs sampler was invented. The above theorem provides a theoretical confirmation of such a conflict. It seems to be a reasonable strategy to "group" or "collapse" when it is computationally feasible. But as a whole, it is left to the reader to make compromises to balance all factors mentioned.

7.7 Problems

1. Implement the SW algorithm for simulating a 64×64 Ising model near the critical temperature.
2. Implement Wolff's algorithm for the above task.
3. Implement the Niedermayer's generalization for the above.
4. Experiment with different choices of p_a and p_b in the Niedermayer's algorithm. Can you find a pair of p_a and p_b so that the resulting algorithm outperforms Wolff's algorithm?
5. Is it possible to generalize Wolff's or Niedermayer's algorithms along the line of Li and Sokal (1989) described in Section 7.3?