

马氏链蒙特卡洛方法

RainEggplant¹

(1. 清华大学电子工程系 无 76 2017*****)

【摘要】

本文通过马尔科夫蒙特卡洛方法 (MCMC) 实现了对二维高斯相关系数的估值, 并分析了不同举荐分布对估值结果和性能的影响。在此基础上, 采用 MH 算法、Swendsen-Wang 算法和 Self-adjusted mixture sampling 对 Potts 模型归一化常数进行了估计, 绘制了不同温度下的能量直方图和典型样本。并且比较了不同算法的性能。

1 二维高斯分布相关系数估计

1.1 介绍

马氏链蒙特卡洛 (Markov Chain Monte Carlo, MCMC) 方法, 是马氏链理论的一个重要应用。从 1950 年萌芽, 马氏链蒙特卡洛方法在实践中不断发展, 广泛应用于各种学科领域 (如信息科学、物理、化学、生物学、金融、材料等) 的科学计算, 展示出越来越强大的威力。

马氏链蒙特卡洛方法的基本思想是通过生成一个易于实现的不可约遍历链 $X = \{X_n, n \geq 0\}$, 其平稳分布为 π , 来产生服从给定分布 π 的随机变量的样本^[1]。在下方我们将使用经典的 Metropolis-Hastings 算法进行二维高斯分布的相关系数的估计。

1.2 模型

给定如下二维高斯分布:

$$\mathcal{N}\left\{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \begin{pmatrix} 5 \\ 10 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}\right\} \quad (1)$$

有均值 $\mu = \begin{pmatrix} 5 \\ 10 \end{pmatrix}$, 协方差矩阵 $\Sigma = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}$ 。从而可以计算得两个分量的相

$$\text{关系系数 } \rho = \frac{\text{Cov}(X_1, X_2)}{\sqrt{\text{Var}(X_1)\text{Var}(X_2)}} = \frac{1}{\sqrt{1 \times 4}} = \frac{1}{2}。$$

下面, 我们使用 MH 算法对该二维高斯分布进行随机采样, 并利用采样结果估计相关系数。

1.3 基本理论和方法

Metropolis-Hastings 算法 (MH 算法) 是由 Metropolis 提出、Hastings 加以推广的一种经典的 MCMC 方法。该算法通过“提议”和“接受-拒绝”步骤进行迭代, 满足细致平衡方程, 从而生成服从给定分布的随机样本。

1.4 具体实现方法

设状态空间为 S , $\pi = (\pi(i), i \in S)$ 为任意给定的概率分布, 而 $Q(i, j), i, j \in S$ 为任选的易于实现的转移概率函数 (提议函数)。 $X = \{X_n, n \geq 0\}$ 由下列步骤生成:

- (1) 将初始状态任意 (随机) 取为 S 中的某个状态;
- (2) **提议:** 假设当前状态为 x , 按照转移概率函数 $Q(y|x)$, 随机地到达某个状态 y ;
- (3) **接受-拒绝:** 如果 $y \neq x$, 则以概率 $\alpha_{xy} = \min\{\frac{\pi(y)Q(x|y)}{\pi(x)Q(y|x)}, 1\}$ 接受系统
在下一个时刻跳转到状态 y ; 否则认为系统还是处在状态 x ;
- (4) 返回步骤 (2), 进行下一轮迭代。

特别地, 当提议分布 $Q(i|j)$ 满足对称性, 即 $Q(i|j) = Q(j|i)$ 时, Metropolis-Hastings 算法退化为 Metropolis 算法。

1.5 方法分析

由前一小节的描述, 可以看到 MH 算法的影响因素包括迭代初值、提议函数 $Q(i|j)$ 、迭代次数。

一般来说, 尽管 Markov Chain 最后都会收敛到目标分布, 但迭代初值的选取会直接影响收敛到目标分布所需的迭代次数。因此, 选择初值时, 最好加入一个 “burn-in period” (预烧期, 预选期) [2]。

而对于提议函数, 一般希望其满足以下要求[3]:

- (1) 对于固定的 j , 能够便捷地从 $Q(i|j)$ 中产生随机数;
- (2) 提议函数形式与目标分布越接近, 生成的随机样本效果越好。

1.6 算法

Algorithm 1 Metropolis-Hastings 算法

```

Initialize  $X(1)=(x_1, x_2)$ 
for  $i=1$  to  $N$  do
  Generate  $Y$  from proposal distribution  $Q(y|x)$ 
  Calculate the acceptance probability:
      
$$\alpha = \min\left\{\frac{\pi(Y)Q(X(t-1)|Y)}{\pi(X(t-1))Q(Y|x)}, 1\right\}$$

  Generate  $u$  from  $Uniform[0,1]$ 
  if  $u < \alpha$ 
    set  $X(t) = Y$ ;
  else
    set  $X(t)=X(t-1)$ 
  end if
end for
    
```

1.7 数值结果

对于题设情形，我们采用两种举荐分布进行随机采样。

第一种举荐分布为均匀分布。考虑到服从高斯分布的变量集中分布在均值附近，我们选择 $[-4\sigma_1 + \mu_1, 4\sigma_1 + \mu_1] \times [-4\sigma_2 + \mu_2, 4\sigma_2 + \mu_2]$ 区域下的均匀分布作为举荐分布。此时落在该区域外的取值概率总和非常小，可以忽略不计。

第二种举荐分布为条件二元高斯分布。我们希望其与目标分布相似，故选择分布 $\mathcal{N}(x, \Sigma)$ 。其均值 x 为当前样本点的值，协方差矩阵与目标分布的协方差矩阵一致。

设置预烧期为 迭代次数 $\times 3$ ，以下是不同迭代次数下的数值结果（蓝色点为使用 `mvnrnd` 函数生成的分布，红色点为 MH 采样点）：

迭代 10000 次：

举荐分布	相关系数	接受比例
均匀分布	0.5171	0.1620
二维高斯分布	0.5172	0.5607

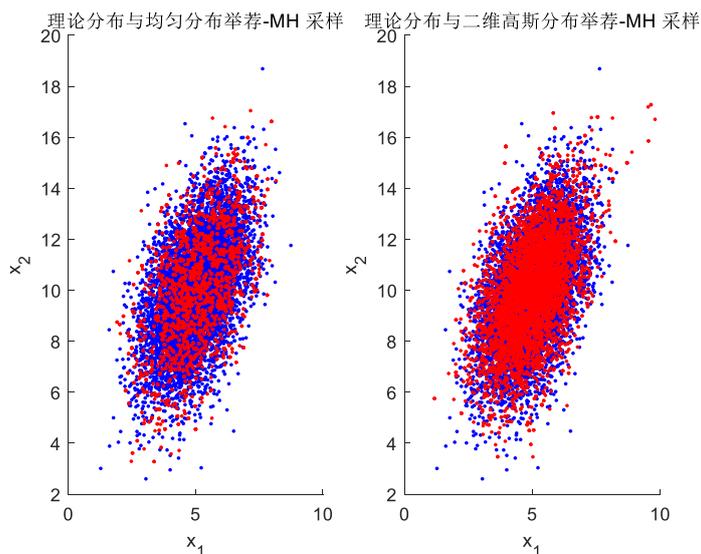


图 1 迭代 10000 次的采样情况

可以看到，由于二维高斯分布举荐有更高的接受比例，其采样点分布更广，遍历程度更高。

迭代 100000 次：

举荐分布	相关系数	接受比例
均匀分布	0.5019	0.1689
二维高斯分布	0.5007	0.5551

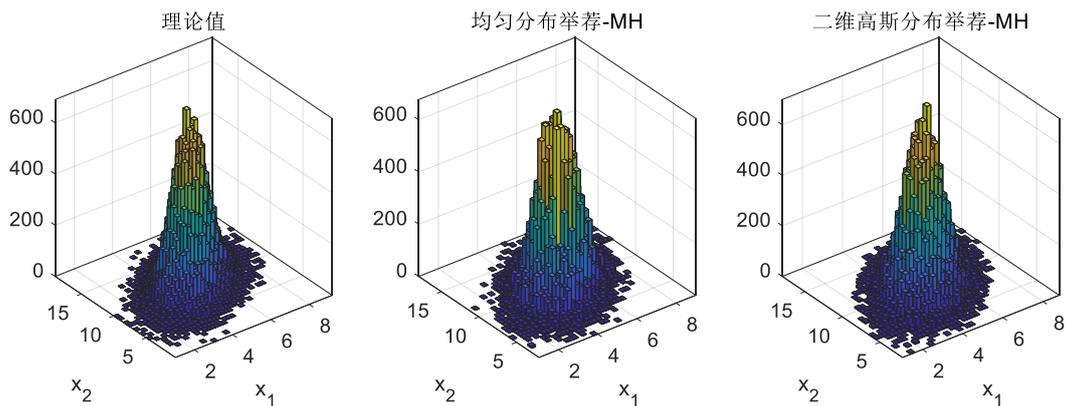


图 2 样本点直方图

可以看到，三者的分布基本一致。我们用 MH 算法完成了对目标二元高斯分布的采样。

1.8 讨论

此处我们关注一下两种不同的举荐分布下的接受比例和收敛速度。

分别在不同迭代次数下运行 MH 算法 100 次，记录相关系数结果与真实值的相对偏差的平均值，绘出如下图形。

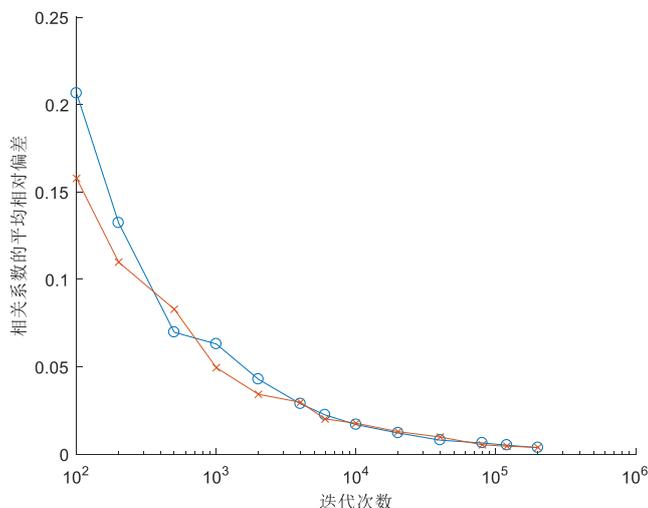


图 3 不同举荐分布下的相关系数的平均相对偏差

可以看到，当迭代次数较小时，二维高斯分布举荐明显地优于均匀分布举荐，更快地收敛到真实值附近。但随着迭代次数的增加，两者的平均相对偏差趋于一致。

平均接受比例与迭代次数无关。对于均匀分布举荐，平均接受比例为 0.1698；对于二维高斯分布举荐，平均接受比例为 0.5529。

2 Potts 模型的归一化常数的估计

2.1 介绍

Potts 模型是一种重要的概率模型。其起源于统计物理，是 Ising 模型的推广。简单地介绍其物理意义：Potts 模型网格上的点代表晶格中粒子的自旋。有两种力作用作用在粒子上，一种力来源与相邻粒子间的相互作用以及外场的影响，这种影响使得相邻的粒子状态趋于一致（有外场作用时，还趋于与外场一致），即尽量使得系统的总能量达到最小。另外一种力则来源于环境热噪声的扰动，它迫使粒子发生自旋的随机转变。涨落的大小由温度决定，温度越高，随机涨落干扰越强。

实际上，除去上面的物理意义外，Potts 模型可以具有其他丰富的内涵。例如可以用来建模社会科学中的投票模型、神经科学中的 Hopfield 网络等。

归一化常数是 Potts 模型的重要数值特征，对 Potts 模型的归一化常数的估计，代表了一大类科学计算问题，至今仍是非常活跃的一个研究课题。

2.2 模型

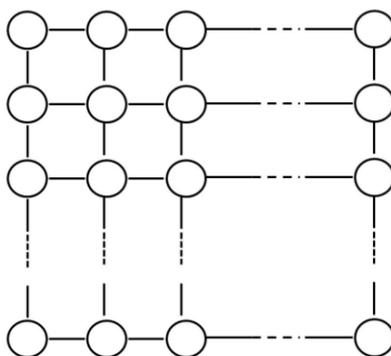


图 4 Potts 模型

考虑定义在上图周期边界网格上的 $K = 20 \times 20$ 个离散随机变量 $x = (x_1, \dots, x_{(20 \times 20)})$ ，每个变量有 q 个可能取值。 K 个变量的联合分布列为：

$$p(x) = \frac{1}{Z(T)} \exp\left[-\frac{u(x)}{T}\right] \quad (2)$$

$$u(x) = -\sum_{i \leftrightarrow j} 1(x_i = x_j) \quad (3)$$

$i \leftrightarrow j$ 表示格点 i 和 j 相邻。 $1(x_i = x_j)$ 是示性函数。在统计物理中，

$u(x)$ 称为能量， T 表示温度，归一化常数 $Z(T) = \sum_x \exp\left[-\frac{u(x)}{T}\right]$ 称为配分函数。

选取 $q = 10$, $T^{-1} = 1.4; 1.4065; 1.413; 1.4195; 1.426$ ，代表五个温度下的五个不同的 Potts 模型。接下来的研究针对该参数下的模型展开。

2.3 基本理论和方法

显然, 直接由定义式计算归一化常数 $Z(T)$ 是不可行的。因此, 通常的做法是通过 MCMC 方法进行采样, 然后利用生成的随机样本估计 $Z(T)$ 。

在几十年中, 研究者尝试或提出了多种采样算法对 Ising 模型或者 Potts 模型进行采样。最基础的做法是直接使用 Metropolis-Hastings 算法进行采样, 但是当温度接近或者低于临界温度时, MH 算法会表现地非常低效^[4]。针对该缺陷, 研究者们提出了一系列 cluster 算法。例如由 Robert Swendsen 和 Jian-Sheng Wang 在 1987 年提出的 Swendsen-Wang 算法^[5], 由 Ulli Wolff 在 1989 年对 SW 算法进行改进后提出的 Wolff 算法^[4] 等。

但是, 研究者们又发现了以上算法的局限和不足。如果需要对多个不同分布下的 Potts 模型进行随机采样, 以上算法只能一一分别地运行, 对于多模态性和反常轮廓的呈现是低效的^[6]。为了解决这些问题, 研究者们提出了两大类算法, 其指导思想是使不同分布下的样本能够相互作用。第一类是基于重叠的算法, 例如 parallel tempering 算法^[7]、serial tempering^[8] 算法、resample-move^[9] 算法等; 第二类是基于状态空间划分的算法, 例如 Wang-Landau 算法^[10] 等。此外, 还有 Zhiqiang Tan 在 adaptive serial tempering 算法和 generalized Wang-Landau 算法基础上提出的 self-adjusted mixture sampling 方法^[6]。

2.4 方法与算法实现

2.4.1 Metropolis-Hastings

MH 算法的基本思想见前文。对于本节的 Potts 模型, 其概率分布如 (2) 式。我们选取的举荐分布 $Q(y|x)$ 是这样的: y 与 x 有且只有一个分量有可能不同, 该分量的选取是均匀随机的, 且该分量的新值也是均匀随机的。显然, $Q(y|x)$ 满足对称性, 因此退化为 Metropolis 算法。

已知 Potts 模型的概率分布如 (2) 式, 因此不难计算得接受概率

$$\begin{aligned} \alpha &= \min \left\{ \frac{\frac{1}{Z(T)} \exp[-\frac{u(y)}{T}]}{\frac{1}{Z(T)} \exp[-\frac{u(x)}{T}]}, 1 \right\} \\ &= \min \left\{ \exp[-\frac{u(y)-u(x)}{T}], 1 \right\} \end{aligned} \quad (4)$$

其算法实现与 1.6 节相同, 不再赘述。

在运行该算法时, 为了提高采样的充分性, 我们每经过 $K = 20 \times 20 = 400$ 次迭代保存一次样本。

2.4.2 Swendsen-Wang

Swendsen-Wang 算法的提出是为了解决 MH 算法临界慢化的问题, 其作为 cluster 算法的代表, 核心思想是在每次迭代中:

1. 每一步创建一个尝试构型, 这个构型被创建的概率 (频率) 为对应的 Boltzmann 权重;
2. 该尝试构型被接受的概率为 1。

对应本节的 Potts 模型, 每次迭代的步骤为^[11]:

1. 对晶格上的每条边，如果其两端的自旋相同，则以概率 $p = 1 - e^{-\beta}$ 放一个棒；
2. 有棒相连的两个自旋被认为属于同一个集团，一个孤立的自旋也算一个集团；
3. 对于每个集团，其自旋（整体）随机的取 $1, 2, \dots, q$ ；
4. 返回 1，进行下一次迭代。

其中， q 为每个变量可能取值的个数， $\beta = \frac{1}{T}$ 。

其算法实现如下：

```

Algorithm 2 Swendsen-Wang 算法
Initialize  $x^{(0)} = \text{randi}([1, q], 20, 20)$ 
for  $t=1$  to  $N$  do
   $x^{(t)} = x^{(t-1)}$ 
   $x\_cluster = \text{get\_cluster}(x)$ 
  for each cluster in  $x\_cluster$  do
     $x^{(t)}[\text{cluster}] = \text{randi}([1, q])$ 
  end for
end for
    
```

其中，`get_cluster` 函数的作用是进行聚类。聚类算法的选择对于 SW 算法的执行效率至关重要，我们采用的是基于并查集的高效的 Hoshen–Kopelman 算法^[12]。

2.4.3 Self-adjusted mixture sampling

为了表述方便，我们定义如下符号：

- m : 需要仿真的温度个数，在本节中 $m = 5$ ；
- $P = (P_1, \dots, P_m)$: 在 m 个温度下的分布；
- L : 标号，取值 $1, 2, \dots, m$ ，对应不同的分布；
- $\beta = (\beta_1, \dots, \beta_m)^T = (T_1^{-1}, \dots, T_m^{-1})^T$: 温度的倒数；
- $\zeta = (\zeta_1, \dots, \zeta_m)^T$: 归一化常数的对数相对值， $\zeta_j = \log(Z_j/Z_1)$ ；
- $\pi = (\pi_1, \dots, \pi_m)^T$: 温度的分布，固定的正数，满足 $\sum_{j=1}^m \pi_j = 1$ ；在本节中，我们取 $\pi_1 = \dots = \pi_m = m^{-1} = 1/5$ ；

Self-adjusted mixture sampling 算法^[6] 同时对不同分布下的样本 (L, X) 进行采样。由 (2) 式可推出， (L, X) 的联合分布和条件分布分别为：

$$(L, X) \sim p(j, x; \zeta) \propto \frac{\pi_j}{e^{\zeta_j}} \exp(-\beta_j \cdot u(x)) \quad (5)$$

$$p(L = j|x; \zeta) = \frac{\pi_j e^{-\zeta_j \cdot \exp(-\beta_j \cdot u(x))}}{\sum_{l=1}^m \pi_l e^{-\zeta_l \cdot \exp(-\beta_l \cdot u(x))}} \propto \frac{\pi_j}{e^{\zeta_j}} \exp(-\beta_j \cdot u(x)) \quad (6)$$

在每次迭代中, 其实现步骤分为三步:

1. Labeled mixture sampling: 采用局部跳转或全局跳转, 从 L_{t-1} 跳转到 L_t ;
2. Markov move: 使用 MH 算法, 从 X_{t-1} 生成新的样本 X_t ;
3. Free energy update: 采用二元更新模式、局部更新模式或全局更新模式, 更新 ζ 。

在本文的实现中, 我们采取了与 Zhiqiang Tan 的 Optimally adjusted mixture sampling and locally weighted histogram analysis^[6] 文 7.1.1 节所提到的 4 个算法中的第 2 个算法完全一致的配置, 即采用局部跳转和两阶段的二元更新模式。下面详细叙述该配置下的每个阶段。

在 labeled mixture sampling 阶段, 我们使用提议分布 $\Gamma(k, j)$ 提议从 L_{t-1} 到 L_t 的跳转。其中, 当 j 与 k 相邻时, $\Gamma(k, j)$ 的取值为与 k 相邻的标号的个数的倒数; 否则 $\Gamma(k, j)$ 取值为 0。对于本节情形, 可用下图表示:

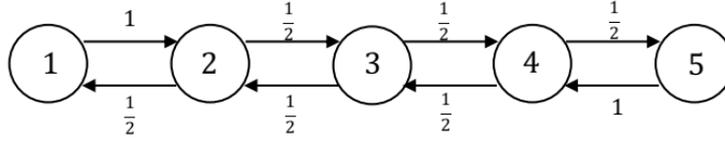


图 5 提议分布 $\Gamma(k, j)$

使用 $\Gamma(L_{t-1}, \cdot)$ 生成标号 j , 相应的接受概率

$$\begin{aligned} \alpha &= \min \left\{ \frac{\Gamma(j, L_{t-1})}{\Gamma(L_{t-1}, j)} \frac{p(j|X_{t-1}; \zeta)}{p(L_{t-1}|X_{t-1}; \zeta)}, 1 \right\} \\ &= \min \left\{ \frac{\Gamma(j, L_{t-1})}{\Gamma(L_{t-1}, j)} \frac{\pi_j}{\pi_{L_{t-1}}} \frac{e^{-\zeta_j}}{e^{-\zeta_{L_{t-1}}}} \exp[-(\beta_j - \beta_{L_{t-1}})u(x)], 1 \right\} \end{aligned} \quad (7)$$

若成功接受, 则置 $L_t = j$, 否则置 $L_t = L_{t-1}$ 。

在 markov move 阶段, 选择 β_{L_t} 作为温度的倒数, 当前采用与 2.4.1 节一致的方式, 迭代 K 次后, 产生新的样本 X_t 。

在 free energy update 阶段, 我们采用两阶段的二元更新模式。具体操作如下:

取对角矩阵 $\mathbf{D} = \text{diag}(d_1, \dots, d_m)$, 其中

$$d_j = \begin{cases} \min(\pi_j, t^{-\beta_0}) & \text{if } t \leq t_0 \\ \min\{\pi_j, (t - t_0 + t_0^{\beta_0})^{-1}\} & \text{if } t > t_0 \end{cases} \quad (8)$$

参数取值与该论文中一致, 即 $\beta_0 = 0.8$, $t_0 = 2 \times 10^5$ 。

ζ 的更新按照下式进行, 其中 $\zeta_1^{(t-1/2)}$ 为 $\zeta^{(t-1/2)}$ 的第 1 个元素。

$$\zeta^{(t)} = \zeta^{(t-1/2)} - \zeta_1^{(t-1/2)} \quad (9)$$

其中,

$$\zeta^{(t-1/2)} = \zeta^{(t-1)} + \mathbf{D}\{\delta_1(L_t)/\pi_1, \dots, \delta_m(L_t)/\pi_m\}^T \quad (10)$$

上式中 $\delta_j(L_t) = 1\{L_t = j\}$ 。

2.5 方法与算法分析

2.5.1 Metropolis-Hastings

前面提到, MH 算法会出现临界慢化的现象。这里我们针对这一现象产生的原因稍作分析。在 Potts 模型中, MH 算法的接受概率由 (4) 式给出。可以看到, 对于相同高度的势垒, 温度越低越难以逾越。因此当温度很低时, MH 算法的收敛速度非常慢。

并且, 可以看到, MH 算法在一次迭代中仅尝试对一个格点进行更新, 这也使得该算法的收敛速度较慢。

2.5.2 Swendsen-Wang

Swendsen-Wang 算法通过构建 cluster, 以 cluster 为单位进行整体更新, 从而在临界点附近能够显著提高效率, 以解决临界慢化问题。

SW 算法中的一个关键步骤是 cluster 的构建。怎样高效地将格点进行归类是非常重要的问题。最佳的办法是使用并查集这一数据结构, 在启用路径压缩和按秩合并并查集树的情况下, 并查集的 union 和 find 操作的时间复杂度几乎为常数^[13]。这能极大地降低 SW 算法的执行时间。

2.5.3 Self-adjusted mixture sampling

Self-adjusted mixture sampling 算法集中了 adaptive serial tempering 算法和 Wang-Landau 算法的优点, 使不同分布下的样本能够相互作用。从而提高算法的收敛速度, 高效呈现 Potts 模型的多模态性和反常轮廓。

2.5.4 归一化常数的对数相对值的估计方法

使用 self-adjusted mixture sampling 算法可以直接给出归一化常数的对数相对值的估计值, 但是 MH 算法和 SW 算法不能, 所以这两种算法需要采用额外的手段进行估计。

统计上, Potts 模型的分布属于指数分布族, 其充分统计量为 $-u(x)$, 自然参数 $\beta = T^{-1}$ 。令 $U = E\{u(x)\}$, 则有^[6]

$$U = -\frac{d}{d\beta} \ln Z(T) \quad (11)$$

令 $\zeta_j = \ln[Z(T_j)/Z(T_j)]$, 为了能够估计 ζ_j 的值, 我们对 (11) 式做如下近似:

$$-\bar{U} \Delta\beta = \Delta \ln Z(T) \quad (12)$$

其中, \bar{U} 为利用对应区间端点处的 U 计算所得平均值。

从而, 我们可以递推地利用 U_1, \dots, U_5 , 求出 ζ_j 的值。

2.6 数值结果

以下仿真环境为 Matlab 2017.a, Windows 10 x64.

2.6.1 Metropolis-Hastings

以下为迭代 200000×400 次的结果:

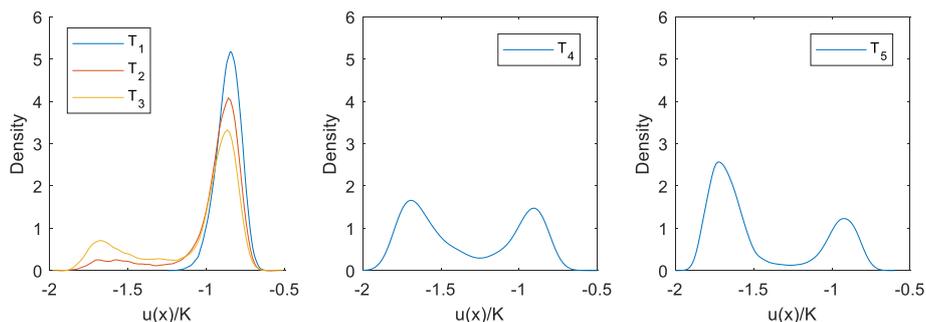


图 6 MH 算法迭代 200000×400 次的能量直方图

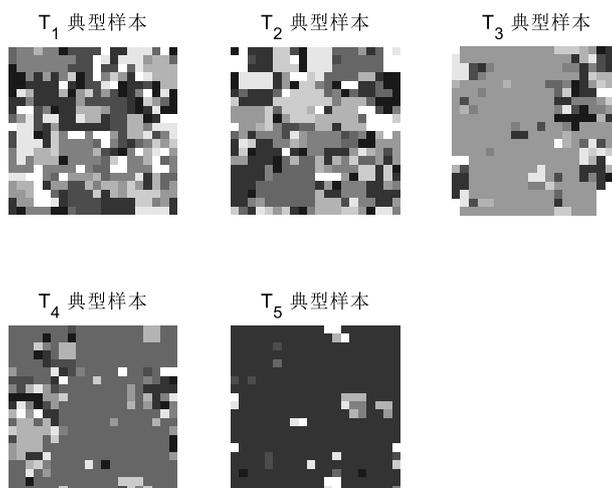


图 7 MH 算法迭代 200000×400 次的典型样本

表 1 归一化常数的对数相对值估计

T^{-1}	1.4	1.4065	1.413	1.4195	1.426
ζ	0	2.36834	5.01107	8.13381	11.7349

2.6.2 Swendsen-Wang

以下为迭代 200000 次的结果:

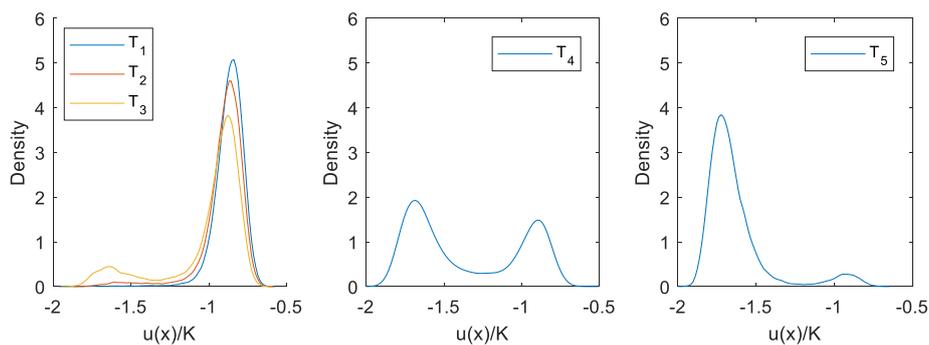


图 8 SW 算法迭代 200000 次的能量直方图

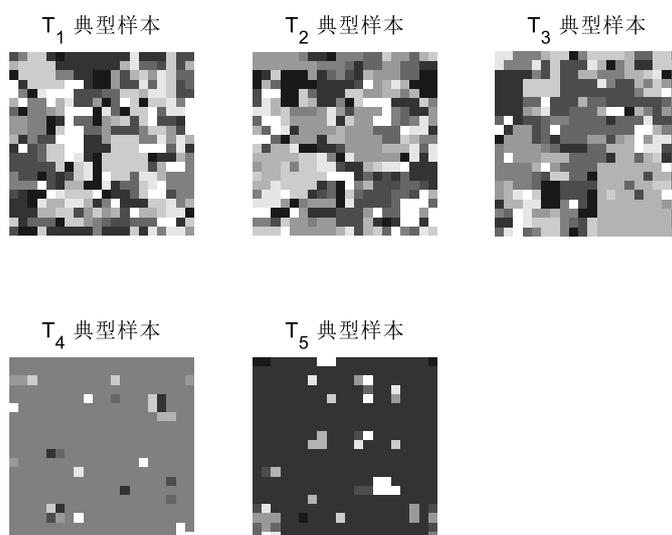


图 9 SW 算法迭代 200000 次的典型样本

表 2 归一化常数的对数相对值估计

T^{-1}	1.4	1.4065	1.413	1.4195	1.426
ζ	0	2.30757	4.79531	7.85886	11.7347

2.6.3 Self-adjusted mixture sampling

以下为按照 Zhiqiang Tan 论文配置获得的结果:

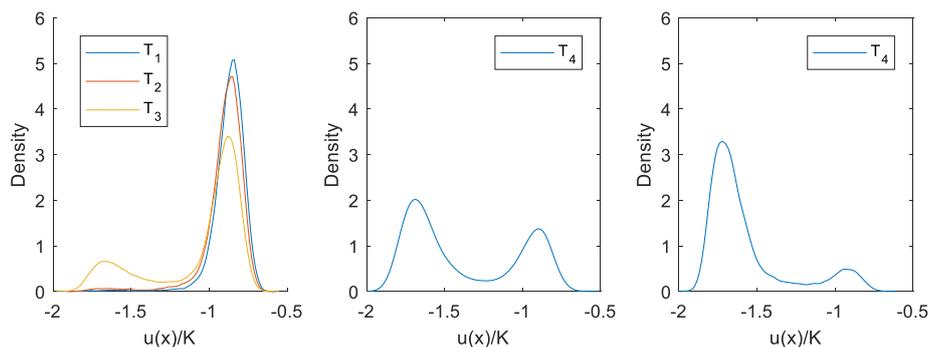


图 10 SAMS 能量直方图

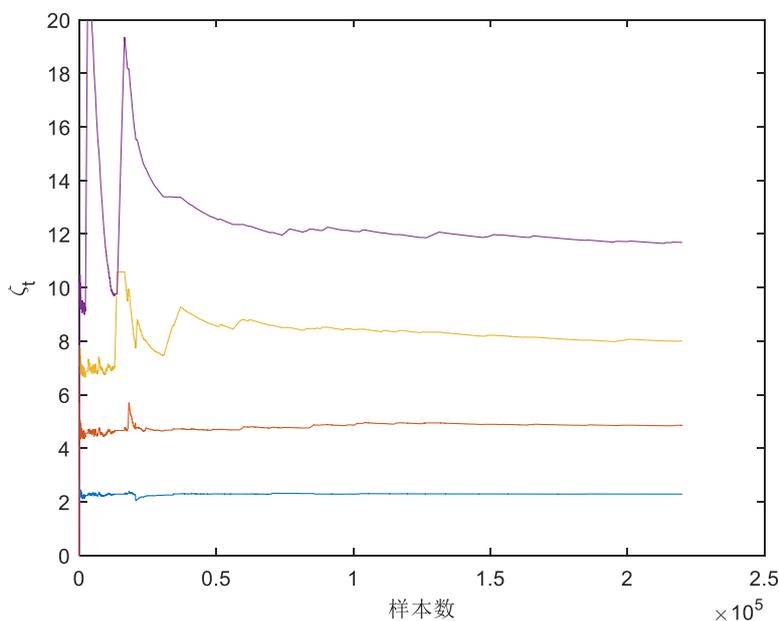


图 11 ζ 的估计值

表 3 归一化常数的对数相对值估计

T^{-1}	1.4	1.4065	1.413	1.4195	1.426
ζ	0	2.28719	4.85344	7.99955	11.682

2.7 讨论

可以看到，三种算法最后都获得了较为接近的估计值。并且随着温度降低，典型样本的各点取值逐渐趋于一致。其中，SAMS 接近完美地复现了 Zhiqiang Tan 的论文。并且，我们可以发现，从 MH、SW 到 SAMS，三种算法的收敛速度依次加快，收敛到真实值附近所需要的迭代次数依次减少。

3 总结

本次的大作业主要探讨了 MCMC 方法以及 Potts 模型。通过本次学习, 我初步掌握了 Potts 模型的基础概念, 学会了使用不同的算法对 Potts 模型的的归一化系数进行估计, 收获颇丰。

致谢

感谢我的室友***同学与我分享资料, 感谢***与我探讨算法。感谢欧智坚老师和助教的付出。

参考文献

- [1] 林元烈. 应用随机过程. 清华大学出版社有限公司, 2002.
- [2] Geyer, C.J., 2011. Introduction to markov chain monte carlo. *Handbook of markov chain monte carlo*, 20116022, p.45.
- [3] 陈平, 徐若曦. Metropolis-Hastings 自适应算法及其应用. 系统工程理论与实践, 2008, 28(1): 100-108.
- [4] Wolff, U., 1989. Collective Monte Carlo updating for spin systems. *Physical Review Letters*, 62(4), p.361.
- [5] Swendsen, R.H. and Wang, J.S., 1987. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical review letters*, 58(2), p.86.
- [6] Tan, Z., 2017. Optimally adjusted mixture sampling and locally weighted histogram analysis. *Journal of Computational and Graphical Statistics*, 26(1), pp.54-65.
- [7] Geyer, C.J., 1991. Markov chain Monte Carlo maximum likelihood.
- [8] Geyer, C.J. and Thompson, E.A., 1995. Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90(431), pp.909-920.
- [9] Gilks, W.R. and Berzuini, C., 2001. Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1), pp.127-146.
- [10] Wang, F. and Landau, D.P., 2001. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical review letters*, 86(10), p.2050.
- [11] 丁成祥. 统计物理中的蒙特卡罗算法简介. 2013
- [12] Hoshen, J. and Kopelman, R., 1976. Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. *Physical Review B*, 14(8), p.3438.
- [13] Tarjan, R.E., 1979. A class of algorithms which require nonlinear time to maintain disjoint sets. *Journal of computer and system sciences*, 18(2), pp.110-127.