

马氏链蒙特卡罗方法及其应用

陈誉博

(清华大学 电子工程系 无42 2014011058)

摘要 本文首先使用了Metropolis – Hastings算法对二维高斯分布的相关系数进行了估计,并且讨论了提议分布的选取对于采样结果以及估计结果的影响。接下来,本文使用了AIS算法、SAMC算法、TAP算法、改进的RTS算法估算RBM模型的归一化常数。我使用MATLAB对四种算法分别进行仿真,综合比较四种算法的计算精度、计算效率和数值性能,并评估了以上四种算法各自的优缺点。最后,本文使用AIS算法估计四个RBM模型的归一化常数,该方法在计算精确度上较改进的RTS方法和TAP方法有明显改善,在计算效率上较SAMC方法有明显提高。

关键词 RBM模型, 归一化常数, MCMC, M – H算法, AIS算法, SAMC算法, TAP算法, RTS算法

Markov Chain Monte Carlo Algorithm and Its Application

CHEN Yu-bo

(Electronic Engineering, Tsinghua University)

Abstract In this paper, the Metropolis-Hastings algorithm is used to estimate the correlation coefficient of the two-dimensional Gauss distribution, and then discuss the effect to the sample result and estimate result of different proposal distributions. Next, I use Annealed Importance Sampling (AIS) algorithm, Stochastic Approximation Monte Carlo (SAMC) algorithm, Thouless-Anderson-Palmer (TAP) algorithm and adjusted Rao-Blackwellized Tempered Sampling (RTS) algorithm to estimate the normalized constant of the RBM model. Using MATLAB for simulation, I compared the accuracy, time complexity and numerical performance of the four algorithms mentioned above and evaluate the advantages and disadvantages of the four algorithms. Eventually I use AIS algorithm to estimate the normalized constants of the four RBM model, which is better than adjusted RTS algorithm and TAP algorithm in accuracy, and better than SAMC algorithm in time complexity.

Key words RBM model, normalized constant, MCMC, MH, AIS, SAMC, TAP, RTS algorithm

1 引言(Introduction)

1.1 MCMC 方法

马尔科夫链 (Markov Chain) 是时间离散, 状态也离散的具有 Markov 性的随机过程。蒙特卡罗算法涉及到模拟观测、模拟采样和期望估计, 以及多元分布的归一化常数的估计, 其精髓在于使用随机数 (或伪随机数) 来解决计算问题。马氏链蒙特卡罗方法 (Markov Chain Monte Carlo Methods, MCMC), 是马氏链理论和蒙特卡罗算法相结合的一个重要的应用[3][7]。MCMC方法是由Metropolis提出的。他提出在以 π 为平稳分布的马氏链上产生相互依赖的样本, 进而马氏链的值可以看成是从分布 π 中抽取样本。之后Hastings对MCMC方法加以推广, 形成了Metropolis – Hastings (MH)算法[15]。但是, 普通的MH算法迭代次数较少时难以快速收敛到目标分布, 而且提议分布 (proposal distribution)的选取对采样结果有很大的影响。

本文使用了MH算法对二维高斯分布进行了采样, 并且用采样值估算了二维高斯分布的相关系数。除此之外, 本文使用了两种最常用的提议分布: 二维正态分布和二维均匀分布来比较讨论不同的提议分布对采样结果的影响。

1.2 估算 RBM 模型的归一化系数

受限玻尔兹曼机(Restricted Boltzmann Machine)是深度学习中最重要最基础的模型之一。它是一种常用的无向图模型。随着深度学习方法的快速发展, 估算RBM模型归一化系数的问题受到了广泛关注。

目前, RBM归一化系数的估算方法主要有四种, 分别是AIS方法[10][12], SAMC方法[14], TAP[9]方法和RTS方法[4]。其中, AIS算法是机器学习理论中最主要的估算归一化系数的方法。RTS算法和AIS算法的主要区别在于RTS算法将逆温度系数视为随机变量, 状态跳转时根据其 posterior 分布产生新的系数。TAP算法采用统计物理的思想, 将RBM模型的参量与统计物理中的熵、自由能等物理量联系起来, 用物理方法估计归一化系数。SAMC算法与上述算法都有一定的区别, 因为SAMC算法估计的并不是每一个RBM模型的确切值, 而是各个模型之间归一化系数的相对值。

但是, 上述算法均存在一定的不足。其中, AIS算法的计算精确程度很高, 但是计算效率远低于TAP算法; RTS算法的计算精确度也比较高, 但是计算效率略低于AIS算法; TAP算法的计算效率非常高, 但是计算结果存在一定的误差; SAMC方法的精确程度也很高, 但是计算效率以及计算结果的稳定程度均不如AIS方法。我使用MATLAB对四种模型分别进行仿真与改进, 并综合比较四种方法的计算效率和精确程度。最终, 我权衡计算效率和数值性能等多方面因素, 选择AIS算法估计四个模型的归一化系数, 并通过计算方差来估计结果的上下界。

2 二维高斯的相关系数估计(Correlation coefficient of 2D Gauss distribution)

2.1 概述

已知下述二维高斯分布:

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

使用概率论中的相关理论可以计算得到该分布的相关系数 $\rho = 0.5$ 。

2.2 Metropolis – Hastings 算法及其分析

使用 Metropolis – Hastings 算法对该二维高斯分布进行采样，并通过采样值估计该二维高斯分布的相关系数。Metropolis – Hastings 算法步骤如下[15]:

Algorithm 1 Metropolis-Hastings Algorithm

Initialize $X^{(1)}=(x_1, x_2)$

for $i = 1$ **to** N **do**

 Generate Y from proposal distribution $T(X)$

 Evaluate the acceptance probability

$$\alpha = \min \left\{ 1, \frac{\pi(Y)T(Y, X^{(t-1)})}{\pi(X^{(t-1)})T(X^{(t-1)}, Y)} \right\}$$

 Generate a u from Uniform[0,1] distribution

 if $u < \alpha$ then $X^{(t)} = Y$ else $X^{(t)} = X^{(t-1)}$

end for

可以证明，Metropolis – Hastings算法构造出来的链是一个Markov链，且其平稳分布为目标分布 $\pi(x)$ （证明方法见附录 A）。简单来说，M – H算法依次更新状态变量，并计算其接受概率。之后按照转移概率跳转到新产生的状态变量。该算法的复杂度为 $O(N)$ 。这是最原始，也是应用最为广泛的MCMC方法，此后出现的方法基本都是建立在Metropolis和Hastings提出的“构造转移核”的基础之上。

2.3 数值结果与分析

我使用了MATLAB对采样过程进行仿真。简单起见，提议分布采用均匀分布。取采样次数为100000，由此估计出来的相关系数估计值为0.5022。在提议分布为均匀分布的情况下，图 1 与图 2 显示了MH算法的采样结果与理论分布的对比效果。（红：MH采样值，蓝：理论分布值）

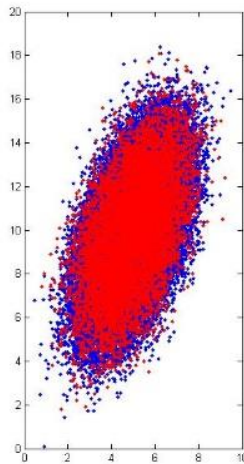


图 1 理论分布与 MH 采样对比图(T 为均匀分布)

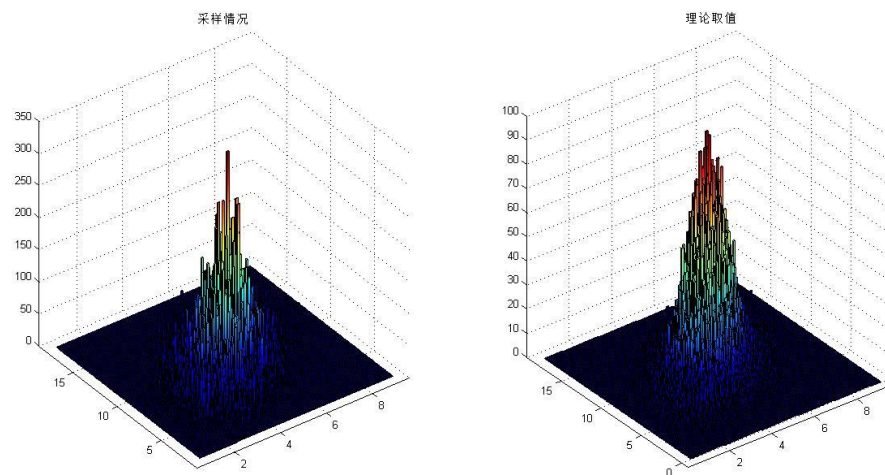


图 2 理论分布与 MH 采样直方图对比(T 为均匀分布)

以上结果均为提议分布为均匀分布情况下的采样对比。为了研究不同提议分布对采样结果与计算结果的影响，我对提议分布为二维正态分布（均值与目标分布相同，协方差矩阵为单位阵）的情况也做了仿真。迭代次数同样取100000，由此得到的相关系数估计值为0.4933。在提议分布为二维正态分布的情况下，图 3 与图 4 显示了 MH 算法的采样结果与理论分布的对比效果。（红：MH 采样值，蓝：理论分布值）

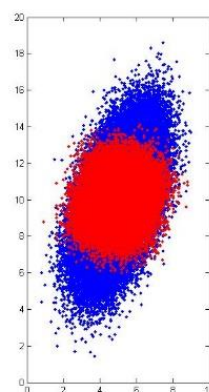


图 3 理论分布与 MH 采样对比图(T 为二维正态分布)

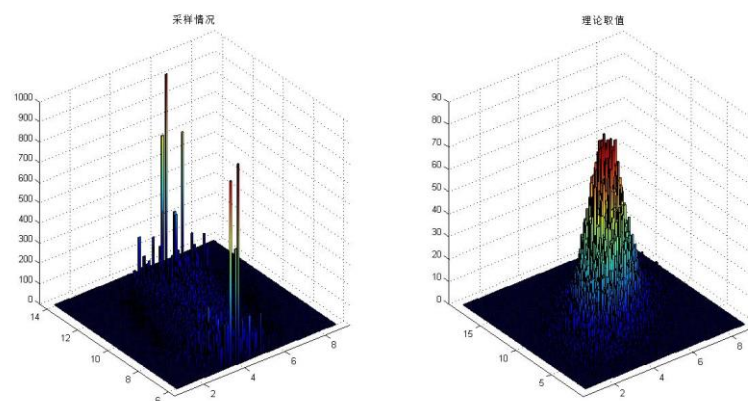


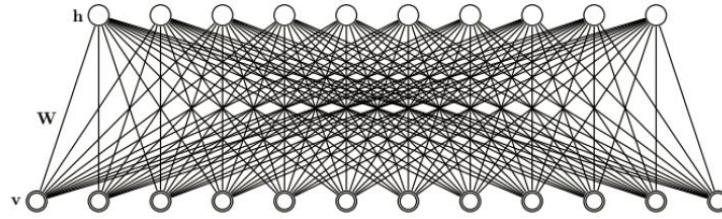
图 4 理论分布与 MH 采样直方图对比(T 为二维正态分布)

仿真结果表明，不同的提议分布对于最终的采样效果与估计结果有非常大的影响。当提议分布为均匀分布时，每次迭代产生的新的采样点在状态空间内均匀分布，因此采样的结果更加接近理论分布的结果。当提议分布为二维标准正态分布时，每次迭代产生的新采样点集中在以均值为圆心，标准差为半径的一个圆形区域内，位于圆形区域之外的概率极小，所以会导致MH采样的结果集中在均值附近的区域，相关系数的估计精度也随之下降。理论上来说，提议分布与目标分布越接近，得到的采样效果越好。

3 RBM模型归一化常数的估计(Normalized constants of RBM model)

3.1 概述

受限玻尔兹曼机（RBM）是深度学习中最重要最基础的模型之一，它是一种常见的无向图模型，具体结构如下：



RBM由一层观测变量和一层隐变量构成，变量均为 0,1 取值。模型的参数为 $\theta = \{W, b, a\}$ 。模型的能量为：

$$\begin{aligned} E(v, h; \theta) &= -v^T W h - b^T v - a^T h \\ &= -\sum_i \sum_j W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j \end{aligned} \quad (1)$$

RBM模型的概率分布即变量的联合分布为：

$$p(v, h; \theta) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta)) \quad (2)$$

其中

$$Z(\theta) = \sum_v \sum_h \exp(-E(v, h; \theta)) \quad (3)$$

即RBM模型的归一化常数。

目前，RBM归一化系数的估算方法主要有四种，分别是AIS方法，SAMC方法，TAP方法和RTS方法。

3.2 算法

3.2.1 AIS (Annealed Importance Sampling) 算法

3.2.1.1 基本理论

AIS算法是由Ruslan Salakhutdinov于2009年提出的一种算法[12]，是目前估算RBM模型归一化常数最主要的算法。AIS算法的基本思想是先构造一个容易计算的RBM模型，然后通过构造中间概率分布，以Markov链跳转的方法从一个分布跳转到下一个分布。最后以重要性采样(Importance Sampling)的思想计算两个RBM模型归一化系数的比值：

$$\frac{Z_B}{Z_A} \approx \frac{1}{M} \sum_{i=1}^M \omega_{AIS}^{(i)} = \hat{r}_{AIS} \quad (4)$$

3.2.1.2 方法

假设有两个RBM模型A和B，其能量分别为

$$E_A = E(v, h^A; \theta^A), E_B = E(v, h^B; \theta^B)$$

假设我们可以定义向量 $\beta = [\beta_1, \beta_2, \dots, \beta_K]$ ($0 = \beta_0 < \beta_1 < \beta_2 < \dots < \beta_K = 1$) 和与其一一对应的一系列中间概率分布： $P_0 \dots P_K$ ，其中 $P_0 = P_A$ ， $P_K = P_B$ 。则根据RBM模型的二部图结构可以计算出这些中间概率（未归一化）的值（详细推导过程见附录B）：

$$\begin{aligned} P_k^*(v) &= \sum_{h^A, h^B} e^{(1-\beta_k)E_A + \beta_k E_B} \\ &= e^{(1-\beta_k) \sum_i b_i^A v_i} \left[\prod_{j=1}^{F_A} \left(1 + e^{(1-\beta_k)(\sum_i W_{ij}^A v_i + a_j^A)} \right) \right] e^{\beta_k \sum_i b_i^B v_i} \left[\prod_{j=1}^{F_B} \left(1 + e^{\beta_k (\sum_i W_{ij}^B v_i + a_j^B)} \right) \right] \end{aligned} \quad (5)$$

从以上表达式可以给出以下三个条件分布：

$$\begin{aligned} p(h_j^A = 1|v) &= g\left((1 - \beta_k)(\sum_i W_{ij}^A v_i + a_j^A)\right) \\ p(h_j^B = 1|v) &= g\left(\beta_k(\sum_i W_{ij}^B v_i + a_j^B)\right) \\ p(v_i' = 1|h) &= g\left((1 - \beta_k)(\sum_j W_{ij}^A h_j^A + b_i^A) + \beta_k(\sum_j W_{ij}^B h_j^B + b_i^B)\right) \end{aligned} \quad (6)$$

其中 $g(x) = 1/(1 + \exp(-x))$ 。简单起见，定义 $\theta^A = \{0, b^A, a^A\}$ 。A模型的构造一般采用 base rate 方法[12]。则A模型的归一化系数和v的概率分布可由如下表达式给出：

$$Z_A = \prod_j (1 + e^{a_j}) \prod_i (1 + e^{b_i}) \quad (7)$$

$$P_A(v) = \prod_i 1/(1 + e^{-b_i}) \quad (8)$$

3.2.1.3 算法实现

Algorithm 2 Annealed Importance Sampling Algorithm

Select β_k with $0 = \beta_0 < \beta_1 < \beta_2 < \dots < \beta_K = 1$

Sample $v^{(1)}$ from P_A using (7)

for $k = 2: K$ **do**

 Evaluate $p(h_j^A = 1|v^{(k-1)})$ and $p(h_j^B = 1|v^{(k-1)})$ using (6)

 Sample h^A and h^B from $p(h_j^A = 1|v^{(k-1)})$ and $p(h_j^B = 1|v^{(k-1)})$

 Evaluate $p(v_i' = 1|h)$ using (6)

 Sample $v^{(k)}$ from $p(v_i' = 1|h)$

end for

Set $\omega_{AIS} = \prod_{k=1}^K P_k^*(v^{(k)}) / P_{k-1}^*(v^{(k)})$

每一条Markov链可以计算得到一个 ω_{AIS} 。运行多个平行的马氏链，就可以用(4)和(7)计算出B模型的归一化系数 Z_B 。通过构造一个易于计算的RBM，我们就可以估算出一个难以计算的RBM的归一化系数。

3.2.2 RTS (Rao-Blackwellized Tempered Sampling) 算法

3.2.2.1 基本理论

RTS算法是由David E. Carlson于 2016 年提出的在AIS算法的基础上进行一些改进而形成的另一种估算归一化系数的算法[4]。RTS算法的基本思想与AIS算法中

马氏链状态跳转的思想基本相同，但是在迭代过程中计算方法较AIS算法有一定的区别。首先，RTS算法中的逆温度系数(inverse temperature)是作为随机变量出现，而不是像AIS算法中作为确定性向量的形式出现。其次，RTS算法中对于逆温度系数边缘分布的估算并没有采用传统的统计出现次数的方法，而是使用了Rao – Blackwellized形式[11]的估计算法。最后，RTS迭代过程中对于逆温度系数的更新并不是采用了AIS的循环迭代的方式，而是从后验分布中采样得到新的逆温度系数。

此外，RTS方法另一种理解方式是Rao – Blackwellized形式下的似然值最大化算法。详细解释参见附录 C。

在实际仿真中，为了提高算法的运行速度，也为了让逆温度系数的取值更加接近先验分布，我对RTS算法进行了一些改进，就是将逆温度系数的采样分布从后验分布改成了先验分布。经过修改之后，在不同的迭代过程中逆温度系数的取值更加接近先验分布，算法的收敛效果与计算准确性也有一定的提升。

3.2.2.2 方法

RTS算法的基本假设与AIS算法基本相同，即：有两个RBM模型A和B，模型A的归一化系数容易计算，模型B的归一化系数为需要计算的量。规定逆温度系数的样本空间为 $\{\beta_k, k = 1, 2, \dots, K | 0 = \beta_0 < \beta_1 < \beta_2 < \dots < \beta_K = 1\}$ 。规定逆温度系数的先验分布为 $r(\beta_k) = r_k$ 。采取与AIS算法相同的处理方法，定义一系列与逆温度系数一一对应的中间分布 $f_0 \dots f_K$ ，其中 $f_0 \propto f_A$ ， $f_K \propto f_B$ 及其未归一化的函数 $f_0^* \dots f_K^*$ 。则 f_k^* 的表达式为：

$$f_k^*(v) = \frac{P_k^*(v)}{Z_A^{1-\beta_k}} \quad (9)$$

其中 $P_k^*(v)$ 和 Z_A 的值由(5)和(7)给出。

如果用 \hat{Z}_k 来表示 $f_k^*(v)$ 归一化系数的估计值，那么可以定义以下联合概率分布：

$$q(v, \beta_k) \propto f_k^*(v) r_k / \hat{Z}_k$$

由此容易计算出以下条件分布：

$$q(v | \beta_k) \propto f_k^*(v) \quad (10)$$

$$q(\beta_k | v) = \frac{f_k^*(v) r_k / \hat{Z}_k}{\sum_{k'=1}^K f_{k'}^*(v) r_{k'} / \hat{Z}_{k'}} \quad (11)$$

令 $\hat{Z}_1 \equiv Z_1 = 1$ ，则容易计算出 β_k 的边缘分布为：

$$q(\beta_k) = \frac{r_k Z_k / \hat{Z}_k}{\sum_{k'=1}^K r_{k'} Z_{k'} / \hat{Z}_{k'}} \quad (12)$$

传统的估计边缘分布的方法是统计 β_k 出现的次数取平均。在RTS算法中，采用Rao – Blackwellized形式的估计方法[11]。令 \hat{c}_k 为边缘分布的估计值，则：

$$\hat{c}_k = \frac{1}{N} \sum_{i=1}^N q(\beta_k | v^{(i)}) \quad (13)$$

由(12)和(13)两个表达式可以得到如下表达式：

$$\hat{Z}_k^{\text{RTS}} = \hat{Z}_k \frac{r_1 \hat{c}_k}{r_k \hat{c}_1} \quad (14)$$

(14)式就是用RTS方法得到的 Z_k 的估计值。

对于 Z_k 可以采用迭代的方法进行估计[13]：将所有 \hat{Z}_k 初始化为1；开始循环：使用MCMC方法，用较小长度的Markov链估计 \hat{c}_k 的值；用(14)式更新 \hat{Z}_k 的值。当 \hat{Z}_k 与 Z_k 足够接近的时候，证明估算结果已经基本正确，迭代过程结束。判断 \hat{Z}_k 与 Z_k 是否足够接近的方法是：当 \hat{Z}_k 与 Z_k 足够接近时， $q(\beta_k) \approx r_k$ ，而 \hat{c}_k 是 $q(\beta_k)$ 的估计值，

所以只要 \hat{c}_k 与 r_k 足够接近, 就可以判定 \hat{Z}_k 与 Z_k 足够接近, 迭代过程结束[4]。举例来说, 如果取先验分布为均匀分布, 则一种特定的迭代结束条件就是 $\max|r_k - \hat{c}_k| < 0.1/K$ 。但是在实际仿真中发现 $\max|r_k - \hat{c}_k|$ 的值虽然整体呈现收敛趋势但是当两者误差很小的时候, 其取值会有小的波动。因此有可能会出随机性导致的迭代次数过长, 降低计算效率的问题。因此, 我在此处对RTS算法做了一下改进, 将判断 $\max|r_k - \hat{c}_k| < 0.1/K$ 作为结束条件改为固定迭代次数, 迭代次数足够长, 保证估计结果趋于稳定。

3.2.2.3 算法实现

Algorithm 3 Rao-Blackwellized Tempered Sampling Algorithm

```

for run = 1:runtime do
  Initialize  $\hat{c}_k = 0$ 
  Initialize  $v^{(0)}$  from  $r_k$ 
  Initialize  $\beta \in \{\beta_1, \beta_2, \dots, \beta_K\}$ 
  for  $n = 1:N$  do
    Transition in  $v$  for numswep times using (6)
    Evaluate  $f_k^*(v)$  using (9)
    Evaluate  $q(\beta_k|v)$  using (11)
    Sample  $\beta$  from  $r_k$ 
    Update  $\hat{c}_k$  using (13)
  end for
  Update  $\hat{Z}_k^{RTS} = \hat{Z}_k \times (r_1/r_k) \times (\hat{c}_k/\hat{c}_1)$ 
end for

```

迭代结束之后, 取 $Z_B = \hat{Z}_K^{RTS}$ 可以得到待估计的归一化系数。

3.2.3 TAP (Thouless-Anderson-Palmer)算法

3.2.3.1 基本理论

TAP算法是由Marylou Garbié于 2015 年提出的一种基于统计物理的估算归一化系数的方法[5][9]。TAP算法的基本思想是将归一化系数视作RBM模型的自由能。使用统计物理中的标准方法对自由能进行勒让德变换, 即将自由能重新写成与一个新引入的辅助外场有关的函数。对含有外场作用形式的自由能进行高温展开 (high temperature expansion)[1], 可以得出含外场自由能的二阶近似。显然, RBM模型的自由能等于外场为0的情况下的自由能, 也就是其勒让德变换意义下的最小值。通过求最小值可以得到与自由能和外场有关的迭代表达式, 用迭代的方式可以计算出RBM模型的自由能, 也就是它的归一化系数。

3.2.3.2 方法

首先考虑一个一般的能量模型: 定义二进制单元 $\mathbf{s} = \{s_i, s_i \in \{0,1\}\}$ 以及各单元之间的耦合系数 W_{ij} , 则该模型的能量为 $E(\mathbf{s}) = -\sum_i a_i s_i - \sum_{(i,j)} W_{ij} s_i s_j$ 。考虑温度的作用, 还要在能量的表达式基础上乘以一个逆温度系数 β 。对自由能做勒让德变换, 将其写成以辅助外场 $\mathbf{q} = \{q_i\}$ 为参数的函数: $-\beta F[\mathbf{q}] = \ln \sum_{\mathbf{s}} e^{-\beta E(\mathbf{s}) + \beta \sum_i q_i s_i}$ 。最后把外场置成 0 就可以恢复出真实的自由能。于是自由能的勒让德变换可以通过以 \mathbf{q} 为限值取最大值得到一个共轭变量 $\mathbf{m} = \{m_i\}$ 的函数:

$$-\beta\Gamma[\mathbf{m}] = -\beta \max_{\mathbf{q}} [F[\mathbf{q}] + \sum_i q_i m_i] = -\beta (F[\mathbf{q}^*[\mathbf{m}]] + \sum_i q_i^*[\mathbf{m}] m_i) \quad (15)$$

其中共轭变量的函数 $\mathbf{q}^*[\mathbf{m}]$ 是 $\mathbf{m}[\mathbf{q}] \equiv -\frac{dF}{d\mathbf{q}}$ 的反函数。注意到自由能是 $\mathbf{q} = \mathbf{0}$ 时其勒让德变换的逆变换，所以

$$-\beta F = -\beta F[\mathbf{q} = \mathbf{0}] = -\beta \min_{\mathbf{m}} [\Gamma[\mathbf{m}]] = -\beta \Gamma[\mathbf{m}^*] \quad (16)$$

此处 \mathbf{m}^* 为使 Γ 取到最小值的 \mathbf{m} ，代表了磁化意义下自由能的表达形式。令 $A(\beta, \mathbf{m}) \equiv -\beta \Gamma[\mathbf{m}]$ ，在0附近做高温展开：

$$A(\beta, \mathbf{m}) = A(0, \mathbf{m}) + \beta \left. \frac{\partial A(\beta, \mathbf{m})}{\partial \beta} \right|_{\beta=0} + \frac{\beta^2}{2} \left. \frac{\partial^2 A(\beta, \mathbf{m})}{\partial \beta^2} \right|_{\beta=0} + \dots \quad (17)$$

根据二进制单元 $\{s_i, s_i \in \{0,1\}\}$ 的性质可以得出如下展开：

$$\begin{aligned} -\beta\Gamma[\mathbf{m}] = & -\sum_i [m_i \ln m_i + (1 - m_i) \ln(1 - m_i)] + \beta \sum_i a_i m_i \\ & + \beta \sum_{(i,j)} W_{ij} m_i m_j + \frac{\beta^2}{2} \sum_{(i,j)} W_{ij}^2 (m_i - m_i^2)(m_j - m_j^2) + \dots \end{aligned} \quad (18)$$

上述展开中的零阶项对应于具有受约束磁化值的非相互作用的自旋的熵。二阶项对应于TAP等式中的Onsager反应项[5]。

对(18)保留二阶项，并应用于RBM模型可以得到：

$$\begin{aligned} \Gamma(\mathbf{m}^v, \mathbf{m}^h) \approx & -S(\mathbf{m}^v, \mathbf{m}^h) - \sum_i a_i m_i^v - \sum_j b_j m_j^h \\ & - \sum_{(i,j)} W_{ij} m_i^v m_j^h + \frac{W_{ij}^2}{2} (m_i^v - (m_i^v)^2) (m_j^h - (m_j^h)^2) \end{aligned} \quad (19)$$

其中 S 是熵分量， \mathbf{m}^v 和 \mathbf{m}^h 是用于表示显变量与隐变量的磁化值而引入的量，而逆温度系数 $\beta = 1$ 。为了恢复RBM的自由能，需要对(19)的展开式求最小值。通过计算 $\frac{d\Gamma}{d\mathbf{m}} = 0$ 可以得到对磁化的自我一致约束。取二阶近似，可以得到显变量的约束：

$$m_i^v \approx g \left(b_i + \sum_j W_{ij} m_j^h - W_{ij}^2 \left(m_i^v - \frac{1}{2} \right) (m_j^h - (m_j^h)^2) \right) \quad (20)$$

同样地可以得到隐变量的偏置：

$$m_j^h \approx g \left(a_j + \sum_i W_{ij} m_i^v - W_{ij}^2 \left(m_j^h - \frac{1}{2} \right) (m_i^v - (m_i^v)^2) \right) \quad (21)$$

(20)和(21)两个表达式给出了TAP算法的迭代算子。可以证明，该过程是收敛的[6]。

3.2.3.3 算法实现

Algorithm 4 Thouless-Anderson-Palmer Algorithm

Initialize m_i^v, m_j^h

for $n = 1:N$ **do**

 Evaluate m_j^h using (20)

 Evaluate m_i^v using (21)

end for

Evaluate F using (19)

3.2.4 SAMC (Stochastic Approximation Monte Carlo) 算法

3.2.4.1 基本理论

SAMC算法是Zhiqiang Tan于2014年提出的一种估算归一化常数的方法[14]。与前面三种算法相比，SAMC算法有一个非常显著的特点：只能估计若干个模型之间归一化系数的相对值。SAMC算法的基本思想是采用蒙特卡罗方法进行随机近似。将需要估计的若干个模型的概率分布进行标记混合(labeled mixture)[8]，并通过Markov链的方法进行局部标记跳转(Local-Jump)，再通过混合采样(mixture sampling)的方法进行状态跳转。然后采用随机近似的方法更新RBM的自由能，即归一化系数。

3.2.4.2 方法

设目标分布的个数为 m ，则标记混合采样的基本想法是把目标分布 $P_1 \dots P_m$ 在状态空间 $\{1, \dots, m\} \times \mathcal{X}$ 上混合，构成联合分布：

$$(L, X) \sim p(j, v; \hat{\zeta}) \propto \frac{\pi_j}{e^{\hat{\zeta}_j}} q_j(v) \quad (22)$$

其中 $L = (L_1, L_2, \dots, L_m)$ 为标记， $\pi = (\pi_1, \pi_2, \dots, \pi_m)^T$ 是固定的混合权重， $\sum_{i=1}^m \pi_i = 1$ 。 $\hat{\zeta} = (\hat{\zeta}_1, \hat{\zeta}_2, \dots, \hat{\zeta}_m)$ 是 m 个目标分布的归一化系数(对数) ζ 的假设(对数)值， $\hat{\zeta}_1 = 0$ 。那么在(22)条件下 L 的边缘分布：

$$p(L = j; \hat{\zeta}) = \frac{\pi_j e^{-\hat{\zeta}_j + \zeta_j}}{\sum_{l=1}^m \pi_l e^{-\hat{\zeta}_l + \zeta_l}}, \quad j = 1, \dots, m \quad (23)$$

在(22)条件下的条件分布：

$$p(v|L = j) \propto q_j(v),$$

$$p(L = j|v; \hat{\zeta}) = \frac{\pi_j e^{-\hat{\zeta}_j} q_j(v)}{\sum_{l=1}^m \pi_l e^{-\hat{\zeta}_l} q_l(v)} \quad (24)$$

其中 $p(L = \cdot | v; \hat{\zeta})$ 是一个 $\{1, \dots, m\}$ 上的离散分布。想要获得 $p(L = \cdot | v^{(t-1)}; \hat{\zeta})$ 的采样，可以使用MH算法中状态转移的方法。对于 $k = 1, \dots, m$ ，令 $\mathcal{N}(k)$ 是所有标记中与 k 相邻的标记， $\Gamma(k, \cdot)$ 为从 k 跳到另一个标记的提议分布。一个典型的例子是令 $\Gamma(k, \cdot) = 1/s(k)$ if $j \in \mathcal{N}(k)$ ，否则取0； $s(k)$ 是 $\mathcal{N}(k)$ 中元素的个数[2]。由此就可以定义标记跳转的马氏链，如图5（以 $m = 5$ 的情况为例）：

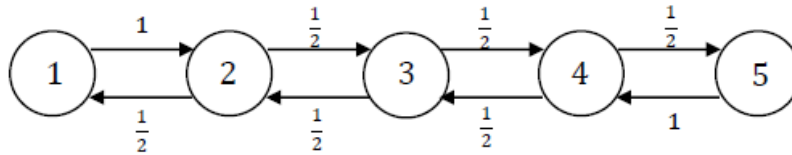


图 5 SAMC 算法的标记跳转 Markov 链

下面采用随机近似的方式更新归一化系数的取值。构造方程：

$$p(L = j; \hat{\zeta}) = \pi_j, \quad j = 1, \dots, m \quad (25)$$

方程的解为：

$$\zeta^{(t-\frac{1}{2})} = \zeta^{(t-1)} + \gamma_t (\delta^{(t)} - \pi), \quad \zeta^{(t)} = \zeta^{(t-\frac{1}{2})} - \zeta_1^{(t-\frac{1}{2})} \quad (26)$$

其中 $\delta^{(t)} = (1\{L_t = 1\}, \dots, 1\{L_t = m\})^T$ ， $\zeta_1^{(t-\frac{1}{2})}$ 是 $\zeta^{(t-\frac{1}{2})}$ 的第一个元素， $\gamma_t = t_0/$

$\max(t_0, t)$, $1\{\cdot\}$ 为示性变量。 t_0 的值一般取在 $2m$ 到 $100m$ 之间[8][13]。

3.2.4.3 算法实现

Algorithm 5 Stochastic Approximation Monte Carlo Algorithm

Initialize ζ , t_0 , L_0 , $v^{(0)}$

Set the value of π and $\Gamma(\cdot, \cdot)$

for $t = 1:N$ **do**

 Generate $j \sim \Gamma(L_{t-1}, \cdot)$ and evaluate acceptance probability

$$\alpha = \min\left\{1, \frac{\Gamma(j, L_{t-1})}{\Gamma(L_{t-1}, j)} \frac{p(j|v^{(t-1)}; \zeta)}{p(L_{t-1}|v^{(t-1)}; \zeta)}\right\}$$

 Generate a u from Uniform[0,1] distribution

 if $u < \alpha$ then $L_t = j$ else $L_t = L_{t-1}$

 Transition in v using (6)

 Update ζ using (26)

end for

3.3 算法性能分析

3.3.1 AIS 算法分析

AIS算法是目前估算RBM模型归一化常数的最主要的算法。每一次运行AIS的过程中，在Gibbs Sample进行状态跳转的过程中，需要计算 F_B 个隐变量的概率，而每计算一个概率都需要计算 $\sum_i W_{ij}^A v_i + a_j^A$ ，共运算 F_A （显变量个数）次。Gibbs Sample一共循环 K 次。AIS算法一共运行 N 次。因此容易知道AIS算法的时间复杂度为 $O(F_A F_B K N)$ 。

由于AIS算法中迭代次数为确定值，而且在保证 $P_k^*(v)$ 与 $P_{k-1}^*(v)$ 足够接近（SIS的基本条件）的条件下， ω_{AIS} 的计算准确性能够得到保证。因此 N 的值不需要非常大。所以，只要适当的选取模型A的参数和 β_k 的分布保证两个中间分布具有足够的相似度，就可以在较短的时间内得到稳定的结果。

3.3.2 RTS 算法分析

我在使用MATLAB对RTS算法进行仿真的时候，对原算法进行了一些改进：

- 1) 将逆温度系数的采样分布从后验分布改成了先验分布。因为如果采用后验分布对逆温度系数进行采样的话，一旦出现后验分布非常畸形的情况，就会使逆温度系数陷入 $\beta \rightarrow 0$ 或者 $\beta \rightarrow 1$ 的区域，有时甚至会出现两个区域中振荡的情况，难以取到中间区域的值。但是这种方法会对最后结果的计算精确度产生影响，因为在迭代刚开始的时候，逆温度系数的后验分布与先验分布差别较大，因此取平均的时候可能会导致计算结果产生偏差。
- 2) 将判断 $\max|r_k - \hat{c}_k| < 0.1/K$ 作为结束条件改为固定迭代次数。因为如果采用 $\max|r_k - \hat{c}_k| < 0.1/K$ 作为结束条件的话，当两者误差很小的时候，其取值会有小的波动，很有可能出现程序长时间无法退出循环的情况，严重影响计算效率。
- 3) 将长MCMC链估计的步骤省略，提高时间性能。在原算法中，需要先用Rao – Blackwellized形式，构造短MCMC链，采用Gibbs Sample的方法对 \hat{Z}_k 进行迭代更新；待 \hat{Z}_k 与 Z_k 足够接近后，再用一个长MCMC链继续更新。但是在仿真过程中发现如果把初始化迭代的步数适当加长（小于长MCMC链的长度），得到的

结果基本趋近于稳定，而且足够接近于真实值。所以我直接使用 \hat{Z}_k 的初始化迭代过程来计算 Z_k 。

RTS算法中包含 N 次迭代。每次迭代过程中包含 N 次计算 \hat{c}_k 。每次计算 \hat{c}_k 时需要 numswEEP 次吉布斯采样进行状态跳转，复杂度为 $O(F_A F_B)$ ；还需要计算 $f_k^*(v)$ ，复杂度为 $O(F_A K)$ 。所以可以得到RTS的时间复杂度为 $O((4 \times \text{numswEEP} \times F_A F_B + F_A K)N)$ 。从复杂度结果可以看出，RTS算法的时间性能稍劣于AIS算法，尤其当隐变量数目较大时差距更明显。

与AIS算法不同的是，RTS算法的准确性取决于迭代次数。如果迭代次数太少，那么结果的精确程度就会大打折扣。所以必须牺牲一定的时间性能来换取结果的精确性。

3.3.3 TAP 算法分析

TAP算法每次循环都需要 $(F_A + F_B)$ 次计算，总循环次数为 N ，所以TAP算法的时间复杂度为 $O(N(F_A + F_B))$ 。

TAP算法作为四种方法中唯一一个确定性算法，其时间效率和算法稳定性都远高于其他三种算法。实际上，TAP算法并没有算法稳定性的概念，因为每次计算得到的结果都是相同的。而且，由于TAP算法中的计算表达式具有一定的物理含义，而且迭代算子的产生是对自由能求最小值的过程，所以由物理规律可以预知算法的收敛速度非常快，因此迭代步数也可以适当减小以获得更好的时间性能。

3.3.4 SAMC 算法分析

SAMC算法中每次迭代过程里最耗时的部分即为 v 的状态跳转。迭代次数为 N 。所以从以上结果可以得出SAMC算法的时间复杂度为 $O(F_A F_B N)$ 。

但是，仿真过程中发现，SAMC算法的收敛速度非常慢。因为每次循环中不仅需要 v 的跳转，还需要 L_t 的跳转。而且SAMC算法一次性计算多个模型的相对值，但是实际上每次循环中只能更新其中一个模型的自由能。这也就导致了时间上的浪费。在四个算法中，SAMC的收敛速度最慢。

3.4 数值结果与分析

以下所有仿真环境均为MATLAB R2014a。

3.4.1 AIS 算法数值分析

AIS算法的运行结果如下：

Model	True $\log \hat{Z}$	Estimates			$\log - \text{prob}$ likelihood	Time
		$\log \hat{Z}$	$\log(\hat{Z} - \hat{\sigma})$	$\log(\hat{Z} + \hat{\sigma})$		
h10	226.1132	226.1036	226.0505	226.1539	-173.8665	59.7s
h20	—	221.0804	221.0133	221.1433	-145.1329	61.4s
h100	—	348.4799	348.4081	348.5469	-113.1199	76.7s
h500	—	459.5887	459.3030	459.8107	-99.2285	148.9s

以上结果的运行条件为： β 在0到1之间均匀取10000个点，共运行100次AIS runs；模型A的参数为 $\theta^A = \{0, b^A, 0\}$ ，其中 b^A 由base rate方法产生[12]；标准差由所有AIS runs的结果计算得到。从数值结果中可以看出，得到的计算结果与真值非常接近，而且运行多个AIS runs得到的所有结果的方差也非常小。可见，AIS

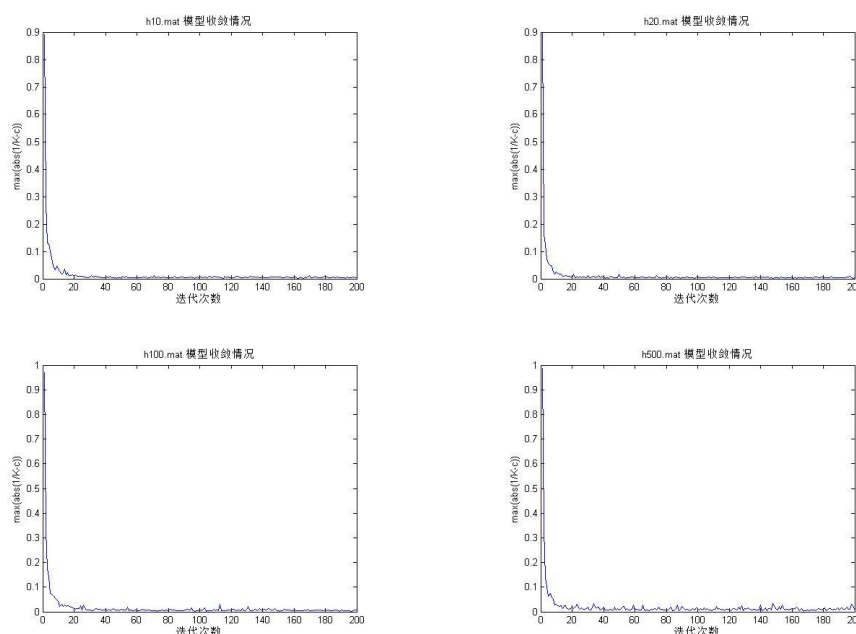
算法的数值性能非常好,而且运行时间也在可接受的范围内。正如 3.3.1 节中提到,只要选取适当的(足够密集的) β 以及适当的A模型(接近B),就能在较短的时间内得到较为精确的估计值。

3.4.2 RTS 算法数值分析

RTS算法的运行结果如下:

Model	True $\log \hat{Z}$	Estimates			log - prob likelihood	Time
		$\log \hat{Z}$	$\log \hat{Z} - \hat{\sigma}$	$\log \hat{Z} + \hat{\sigma}$		
h10	226.1132	225.2142	224.7519	225.6764	-172.9771	40.9s
h20	—	220.4643	220.0890	220.8395	-144.5168	45.6s
h100	—	349.3772	347.3904	351.3640	-114.0172	87.2s
h500	—	463.1152	461.9344	464.2961	-102.7550	281.2s

四个模型的收敛情况见下图:



以上结果的运行条件为: 迭代次数为200次, 每次迭代更新 \hat{c}_k 的次数为100次, 每次更新 \hat{c}_k 之前Gibbs sweep的次数为50次; 模型A的参数为 $\theta^A = \{\mathbf{0}, \mathbf{b}^A, \mathbf{0}\}$, 其中 \mathbf{b}^A 由base rate方法产生[12]; 标准差通过计算后100次迭代的所有数据并使用David E. Carlson给出的公式[4]

$$\text{Var}[\log \hat{Z}_k^{RTS}] \approx \frac{\sigma_1^2}{\hat{c}_1^2} + \frac{\sigma_k^2}{\hat{c}_k^2} - \frac{2\sigma_{1k}}{\hat{c}_k \hat{c}_1}, \sigma_1^2 = \text{Var}[\hat{c}_1], \sigma_k^2 = \text{Var}[\hat{c}_k], \sigma_{1k} = \text{Cov}[\hat{c}_1, \hat{c}_k]$$

计算得到。当RBM模型的隐变量个数较大时, RTS算法的时间性能与数值性能急剧恶化。另外, 从上图可以看出算法的收敛性能也有一些波动。关于时间性能恶化的解释见 3.3.2 小节。至于计算结果精确度的差距, 我认为有以下几个原因:

- 1) 迭代次数不够。上述仿真过程中迭代次数均为200次, 对于较小的RBM模型可能够用, 但是对于较大的RBM模型可能不够, 导致结果不精确。
- 2) 舍去长MCMC链的过程。在 3.3.2 小节中提到了对RTS算法的这一改进。虽然改进之后时间效率有很大提高, 但是计算精度也随之降低。不过计算精度降低的

程度与时间性能的提高程度相比，牺牲一些精度还是值得的。

- 3) 逆温度系数采取先验分布采样而不是后验分布采样。在 3.3.2 小节中提到了对 RTS算法的这一改进。不过，同上一条相同，虽然计算结果的精度有一些损失，但是与迭代速度的提高程度相比也是值得的。

3.4.3 TAP 算法数值分析

TAP算法的运行结果如下：

Model	True $\log \hat{Z}$	Estimates			log – prob likelihood	Time
		$\log \hat{Z}$	$\log \hat{Z} - \hat{\sigma}$	$\log \hat{Z} + \hat{\sigma}$		
h10	226.1132	212.2529	211.5501	212.9557	-160.0159	0.8s
h20	— —	200.2899	196.9614	203.6185	-124.3424	1.0s
h100	— —	336.0618	335.8667	336.2569	-100.7017	2.7s
h500	— —	446.3519	446.3519	446.3519	-85.9917	26.8s

以上结果的运行条件为：迭代次数为10000， m^v, m^h 初值均设置为0.1；标准差采取最后1000次的迭代数据为样本进行计算。从以上结果中可以看出TAP算法两个显著的特点：时间性能极佳，但数值性能极差。关于时间性能的分析见 3.3.3 小节。关于数值性能较差的原因，我认为TAP算法中采用的都是近似表达式，如(19)(20)(21)三个表达式，都是真实值的二阶近似。由于取近似的时候后面无穷项全部舍弃，所以会导致数值上的计算产生误差。如果能够近似到三阶甚至更高，数值性能可能会有所改善。

3.4.4 SAMC 算法数值分析

SAMC算法的运行结果如下：

Model	True $\log \hat{Z}$	Estimates			log – prob likelihood	Time
		$\log \hat{Z}$	$\log \hat{Z} - \hat{\sigma}$	$\log \hat{Z} + \hat{\sigma}$		
h10	226.1132	226.1132	226.1132	226.1132	-173.8761	721.8s
h20	— —	219.4437	219.3553	219.5322	-143.4962	721.8s
h100	— —	347.7488	347.6299	347.8676	-112.3887	721.8s
h500	— —	459.3679	459.2490	459.4868	-99.0077	721.8s

SAMC算法得到的四个相对值的变化趋势见图 6。

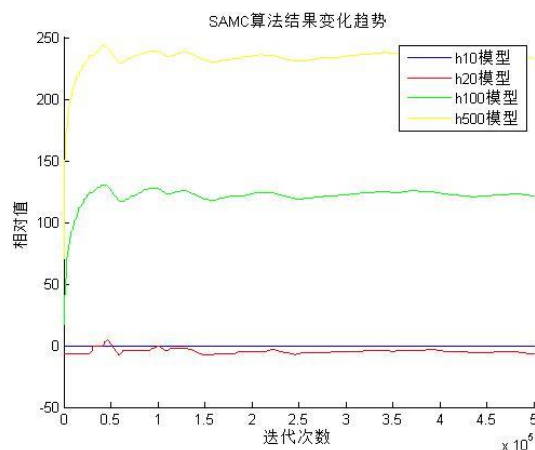


图 6 SAMC 算法的相对值变化趋势

以上结果的运行条件为： $\pi = (\pi_1, \pi_2, \dots, \pi_m)^T = 1/m$, t_0 为8到400之间的随机整数， Γ 取图 5 所示马氏链的转移概率矩阵；迭代次数取500000次；标准差采取最后5000次的迭代数据为样本进行计算。从以上结果可以看出，SAMC算法的收敛情况比较稳定，计算值也比较精确，算法稳定性也非常高。但是SAMC算法一个最突出的问题就是时间复杂度过高。关于时间性能的讨论见 3.3.4 小节。关于数值性能，从图 6 可以看出迭代已经基本趋于稳定。

4 结论(Conclusion)

MH算法是MCMC方法中最为基础的方法。对于低维变量的情况，使用MH算法可以对任意目标分布进行高效的采样。MH算法中提议分布的选取对于采样结果有很大影响。提议分布与目标分布越接近，采样效果越好。对于高维变量，MH算法中“构造转移核”的思想依然是状态跳转的精髓所在，但是在实际使用中一般使用MH算法的改良方法——吉布斯采样。在估算RBM模型归一化系数的四个算法中，基本都采用了吉布斯采样的方法进行马氏链的状态跳转。

本文使用的估算RBM模型归一化系数的算法有AIS算法、改进的RTS算法、TAP算法以及SAMC算法。经过理论分析与数值结果分析可以得到如下结论：

- 1) AIS算法的数值精确度非常高。虽然时间效率不是特别高，但是也在可接受的范围内。另外AIS算法对于不同模型的计算效果非常稳定，几乎不受隐变量个数的影响。
- 2) 改进的RTS算法的计算效率受模型中隐变量的个数影响。隐变量个数较少时效率较高，但当隐变量个数较多时其计算效率急剧恶化。由于我对RTS算法进行了一些提高计算效率同时降低计算精确度的改进，所以RTS算法的数值精确度略逊于AIS算法，尤其是对大模型的计算精确度不高，但是总体的精确度还算可以接受。
- 3) TAP算法的计算效率非常高，因为TAP算法的基本思想来自于物理规律。但是，由于计算时采用了二阶近似，导致最终的计算结果会有舍入误差，数值精确程度大大降低。
- 4) SAMC算法的计算精确度也比较高，但是由于迭代过程每次只更新一个模型的值，所以需要很长的迭代步数才能达到稳定，导致计算效率非常低。

在四种算法中，AIS算法的综合性能最为优异；RTS算法的综合性能也比较高，但是稍逊色于AIS算法；TAP的计算效率最高，但是数值精确度最低，综合性能不高；SAMC算法的计算精确度比较高，但是计算效率最低，综合性能也不高。

最终我们使用综合性能最好的AIS算法对四个RBM模型的归一化常数进行了估计，并计算得到了相应的似然值：

模型	h10	h20	h100	h500
归一化常数	226.103579	221.080377	348.479896	459.588714
似然比	-173.866512	-145.132875	-113.119862	-99.228497

5 致谢(Acknowledgement)

感谢无42班虞立人同学在AIS算法上对我的帮助, 帮助我更好的理解AIS算法, 并为后面几种算法打下理论基础。

感谢无48班王禹同学在RTS算法上对我的指导, 帮助我解决仿真结果不正确的问题。

感谢无42班曾斯涵、王雪超同学在TAP算法上对我的指导和帮助, 让我对TAP算法的物理思想有比较深刻的认识, 并且在经过讨论之后给出了对RTS算法进行改进的基本思路。

感谢无42班裘雨薇同学在SAMC算法上对我的指导和帮助, 通过和她讨论我才理解标记混合采样的具体操作方法。

感谢戴音培助教对我的指导, 在RTS算法上对我的仿真提出了指导性的意见, 并鼓励我继续debug。

6 参考文献(Reference)

- [1] A. Georges and J. S. Yedidia. How to expand around mean-field theory using high-temperature expansions. *Journal of Physics A: Mathematical and General*, 24(9):2173–2192, 1999.
- [2] Charles J. Geyer and Elizabeth A. Thompson. Annealing Markov Chain Monte Carlo with Applications to Ancestral Interface. *Journal of the American Statistical Association*, Vol. 90, No. 431: 909-920, 1995.
- [3] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [4] David Carlson, Patrick Stinson, Ari Pakman, and Liam Paninski. Partition functions from rao-blackwellized tempered sampling. *arXiv preprint arXiv:1603.01912*, 2016.
- [5] D. J. Thouless, P. W. Anderson, and R. G. Palmer. Solution of ‘Solvable model of a spin glass’. *Philosophical Magazine*, 35(3):593–601, 1977.
- [6] E. Bolthausen. An iterative construction of solutions of the TAP equations for the Sherrington–Kirkpatrick model. *Communications in Mathematical Physics*, 325(1):333–366, 2014.
- [7] Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.
- [8] Liang, F., C. Liu and R.J. Carroll (2007). Stochastic approximation in Monte Carlo computation. *Journal of American Statistical Association*, 102, 305-320.
- [9] Marylou Gabrié, Eric W Tramel, and Florent Krzakala. Training restricted boltzmann machine via the thouless-anderson-palmer free energy. In *Advances in Neural Information Processing Systems*, pages 640–648, 2015.
- [10] Neal, R. Annealed importance sampling. *Statistics and Computing*, 2001.
- [11] Robert, C. and Casella, G. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.
- [12] Ruslan Salakhutdinov. *Learning deep generative models*. PhD thesis, University of Toronto, 2009.
- [13] Wang, F. and Landau, D. P. Efficient multiple-range random walk algorithm to calculate the density of states. *Phys. Rev. Let. E*, 2001.
- [14] Zhiqiang Tan. Optimally adjusted mixture sampling and locally weighted histogram analysis. *Journal of Computational and Graphical Statistics*, (just-accepted), 2015.
- [15] 林元烈. 应用随机过程. 清华大学出版社, 2002

7 附录(Appendix)

A. Metropolis – Hastings算法的证明

已知马氏链的转移概率矩阵为 P ，可能存在平稳分布 $\pi = \pi P$ 。马氏链存在平稳分布的条件为该马氏链为不可约非周期链。 π 是平稳分布的一个充分条件如下：

$$P(j, k)\pi_j = \sum_{i,j} P(k, j)\pi_k \quad (27)$$

(27)式被称为细致平衡条件。要证明通过MH方法构造出来的链满足Markov性并以 π 为平稳分布，只需要证明Metropolis – Hastings算法的转移核满足细致平衡条件。显然，MH方法构造出来的链具有Markov性，因为 $X^{(t)}$ 的产生仅仅依赖于 $X^{(t-1)}$ 。下面以二元函数为例证明MH方法构造出来的链满足细致平衡条件。以下证明分为三种情况：

1) $\pi(Y)T(Y, X^{(t-1)}) = \pi(X^{(t-1)})T(X^{(t-1)}, Y)$ ，则从接受概率的计算表达式可得：

$$\alpha(Y, X^{(t-1)}) = \alpha(X^{(t-1)}, Y) = 1$$

而

$$P(X^{(t-1)}, Y) = \alpha(X^{(t-1)}, Y)T(X^{(t-1)}, Y) = T(X^{(t-1)}, Y)$$

$$P(Y, X^{(t-1)}) = \alpha(Y, X^{(t-1)})T(Y, X^{(t-1)}) = T(Y, X^{(t-1)})$$

将上述结果代入条件可以得到：

$$\pi(Y)P(Y, X^{(t-1)}) = \pi(X^{(t-1)})P(X^{(t-1)}, Y)$$

细致平衡条件满足；

2) $\pi(Y)T(Y, X^{(t-1)}) < \pi(X^{(t-1)})T(X^{(t-1)}, Y)$ ，则从接受概率的计算表达式可得：

$$\alpha(Y, X^{(t-1)}) = 1, \alpha(X^{(t-1)}, Y) = \frac{\pi(Y)T(Y, X^{(t-1)})}{\pi(X^{(t-1)})T(X^{(t-1)}, Y)}$$

所以

$$\pi(X^{(t-1)})P(X^{(t-1)}, Y) = \pi(X^{(t-1)})\alpha(X^{(t-1)}, Y)T(X^{(t-1)}, Y) = \pi(Y)T(Y, X^{(t-1)})$$

$$\pi(Y)P(Y, X^{(t-1)}) = \pi(Y)\alpha(Y, X^{(t-1)})T(Y, X^{(t-1)}) = \pi(Y)T(Y, X^{(t-1)})$$

显然细致平衡条件满足；

3) $\pi(Y)T(Y, X^{(t-1)}) > \pi(X^{(t-1)})T(X^{(t-1)}, Y)$ ，则从接受概率的计算表达式可得：

$$\alpha(X^{(t-1)}, Y) = 1, \alpha(Y, X^{(t-1)}) = \frac{\pi(X^{(t-1)})T(X^{(t-1)}, Y)}{\pi(Y)T(Y, X^{(t-1)})}$$

所以

$$\pi(X^{(t-1)})P(X^{(t-1)}, Y) = \pi(X^{(t-1)})\alpha(X^{(t-1)}, Y)T(X^{(t-1)}, Y) = \pi(X^{(t-1)})T(X^{(t-1)}, Y)$$

$$\pi(Y)P(Y, X^{(t-1)}) = \pi(Y)\alpha(Y, X^{(t-1)})T(Y, X^{(t-1)}) = \pi(X^{(t-1)})T(X^{(t-1)}, Y)$$

显然细致平衡条件满足；证毕。

B. 公式(5)的推导

根据RBM模型的二部图结构容易得出以下表达式：

$$P_k^*(v) = \sum_{h^A, h^B} e^{(1-\beta_k)E_A + \beta_k E_B}$$

其中

$$(1 - \beta_k)E_A + \beta_k E_B = (1 - \beta_k)E(v, h^A; \theta^A) + \beta_k E(v, h^B; \theta^B)$$

$$= -(1 - \beta_k) \left(\sum_i \sum_j W_{ij}^A v_i h_j + \sum_i b_i^A v_i + \sum_j a_j^A h_j \right) \\ - \beta_k \left(\sum_i \sum_j W_{ij}^B v_i h_j + \sum_i b_i^B v_i + \sum_j a_j^B h_j \right)$$

所以容易得到:

$$P_k^*(v) = e^{(1-\beta_k) \sum_i b_i^A v_i} \times \sum_h e^{(1-\beta_k) \sum_j (\sum_i W_{ij}^A v_i h_j + a_j^A h_j)} \times e^{\beta_k \sum_i b_i^B v_i} \times \sum_h e^{\beta_k \sum_j (\sum_i W_{ij}^B v_i h_j + a_j^B h_j)} \\ = e^{(1-\beta_k) \sum_i b_i^A v_i} \times \sum_h \prod_{j=1}^{F_A} e^{(1-\beta_k) (\sum_i W_{ij}^A v_i h_j + a_j^A h_j)} \times e^{\beta_k \sum_i b_i^B v_i} \times \sum_h \prod_{j=1}^{F_B} e^{\beta_k (\sum_i W_{ij}^B v_i h_j + a_j^B h_j)} \\ = e^{(1-\beta_k) \sum_i b_i^A v_i} \times \prod_{j=1}^{F_A} \sum_{h_j} e^{(1-\beta_k) (\sum_i W_{ij}^A v_i h_j + a_j^A h_j)} \times e^{\beta_k \sum_i b_i^B v_i} \times \prod_{j=1}^{F_B} \sum_{h_j} e^{\beta_k (\sum_i W_{ij}^B v_i h_j + a_j^B h_j)} \\ = e^{(1-\beta_k) \sum_i b_i^A v_i} \times \left[\prod_{j=1}^{F_A} (1 + e^{(1-\beta_k) (\sum_i W_{ij}^A v_i + a_j^A)}) \right] \times e^{\beta_k \sum_i b_i^B v_i} \times \left[\prod_{j=1}^{F_B} (1 + e^{\beta_k (\sum_i W_{ij}^B v_i + a_j^B)}) \right]$$

证毕。

C. RTS 算法的似然值解释

RTS算法的基本思想可以从似然值的角度来解释。从(12)可以推知, 采样值 $\{\beta_{k(i)}\}$ 的对数似然值为:

$$\log q(\{\beta_{k(i)}\}_{i=1}^N) = \sum_{i=1}^N \log(Z_{k(i)}) - N \log(\sum_{k=1}^K r_k Z_k / \hat{Z}_k) + const \quad (28)$$

而因为 $\beta_{k(i)}$ 是从 $q(\beta|v^{(i)})$ 中采样得到的, 所以我们可以采用Rao – Blackwellized的形式来缩减(28)中的第一个求和式的方差, 得到如下表达式:

$$L_{RB}[Z] = \sum_{i=1}^N \sum_{k=2}^K \log(Z_k) q(\beta_k | v^{(i)}) - N \log \left(\sum_{k=1}^K r_k Z_k / \hat{Z}_k \right) + const \\ = N \sum_{k=2}^K \log(Z_k) \hat{c}_k - N \log \left(\sum_{k=1}^K r_k Z_k / \hat{Z}_k \right) + const$$

对似然值求最大值可以得到归一化系数的估计值。令上式对 Z_k 求导并令导数等于0, 可以得到一系列线性方程组:

$$\sum_{k'=2}^K \frac{r_{k'}}{\hat{Z}_{k'}} \left(\frac{\delta_{k',k}}{\hat{c}_k} - 1 \right) Z_{k'} = r_1, \quad k = 2, \dots, K$$

这组线性方程组的解就是(14)。所以, RTS算法可以看做是在Rao – Blackwellized形式下的似然值最大化算法。