MARKOV CHAIN MONTE CARLO METHOD AND APPLICATION

马氏链蒙特卡洛方法及其应用

姓名 张元鑫

学号 2014011045

班级 无 42

指导教师 欧智坚

2016 年 11 月 12 日

目 录

摘　要

这篇文章主要研究了两个问题，一个是用 Metropolis-Hasting 算法对二维高斯的相关系数进行估计，在重复实验的基础下，得到了相对误差 1%以内的采样结果；另一个是分别用 AIS,TAP,RTS 和 SAMS 模型进行采样，实现了 RBM 模型的归一化系数的估计。本文研究了不同模型得出的归一化系数估计值的稳定性和计算效率，在已有文献的基础上，重新优化设计了采样方法，最终得到满意的估计值。

关键词：蒙特卡洛方法 MH 算法 归一化系数估计

1 二维高斯分布相关系数估计

1.1　介绍

马氏链蒙特卡洛（MarkovChainMonteCarlo,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，逐渐成长为一个颇具分量的理论分支，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。蒙特卡洛方法，即为了要模拟一个服从于给定分布 π 的变量，生成一个比较容易实现的不可约遍历链作为随机样本，使其平稳分布于 π 的一个方法（林元烈书）。现在我们想用 Metropolis-Hasting 算法对二维高斯分布进行随机采样，并使用生成的随机样本来估计二维高斯的相关系数。

1.2　模型

试验中，我们对这样一个二维高斯分布进行随机采样

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

即一个均值分别为 5, 10, 方差为 1 和 4, 互协方差为 1, 1 的二元正态分布。

在 MH 算法中，还用到了马氏链模型。即

$$P(X_{n+1}=x|X_1=x_1,X_2=x_2,X_3=x_3\ldots X_n=x_n)=P(X_{n+1}=x|X_n=x_n)$$

1.3 基本理论和方法

Metropolis 等人在 1953 年最早给出了通过生成一串相关的马氏链实现从分布 π 采样的思路，后来 Hasting 将其进一步推广。简单来说，就是我们要利用给定的稳态分布，通过概率转移矩阵（参照矩阵），生成一组随机采样。

1.4 具体实现方法

（1） 首先给定一个 $x^{(t)}$,通过概率转移矩阵 $T(x^{(t)},y)$生成一个 y

（2） 在 0~1 均匀分布中抽取一个随机数 U 为比较参照值，得到 $x^{(t+1)}$

$$x^{t+1}=\begin{cases}y & if\ \ r(x^t,y)>U\\ x^t & else\end{cases} \qquad (1.4.1)$$

其中

$$r(x,y)=min\left\{1,\frac{\pi(y)T(y,x)}{\pi(x)T(x,y)}\right\} \qquad (1.4.2)$$

（3） 重复上述步骤即可得到 $X=\{X_n,n\geqslant 0\}$，是不可约链，平稳分布于 π

1.5 方法分析

上述方法中，式 1.4.2 是 Metropolis(1953)和 Hasting(1970)当年建议使用的接收方程，实现起来较为方便。后来 Baker（1965）和 Chrles Stein 分别提出了下接受方程

$$r(x,y)=\frac{\pi(y)T(y,x)}{\pi(y)T(y,x)+\pi(x)T(x,y)} \qquad (1.5.1)$$

$$r(x,y)=\frac{\delta(x,y)}{\pi(x)T(x,y)} \qquad (1.5.2)$$

其中$\delta(x,y)$是使得$r(x,y)\leq 1$对所有 x,y 成立的对称函数。
这里我使用了当年 Metropolis 用的接收方程，并且让$T(y,x)=T(x,y)$,最后即得到

$$r(x,y)=min\left\{1,\frac{\pi(y)}{\pi(x)}\right\} \qquad (1.5.3)$$

1.6 算法

Algorithm 1 Metropolis-Hasting 算法（以二元正态分布为例）

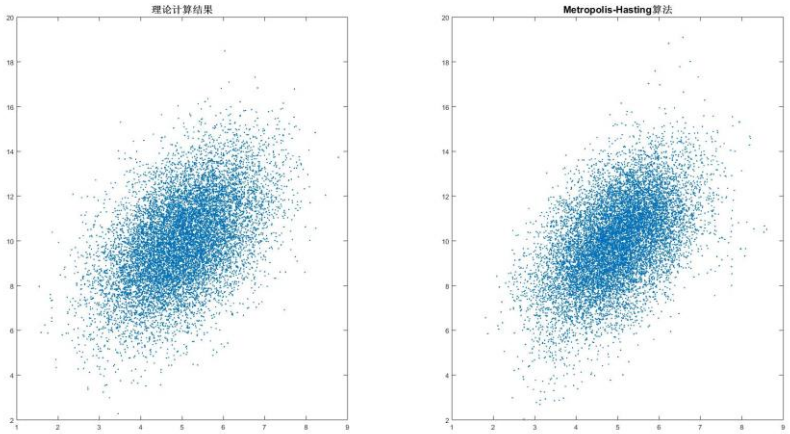
input：均值矩阵 μ，协方差矩阵 Σ
Initialize N
Initialize S_k, k=1,2,3…N
Initialize U_k, k=1,2,3…N
for i=1 **to** N **do**
    sample from $\frac{\pi(y)}{\pi(x)}$ and U_k
    results store in x
end for
Error= mean[(corrcoef(x)-0.5)/0.5]

1.7 算法分析

在方法清楚之后，算法实现起来还是比较简单的。但刚开始时，我设定的转移概率是服从一个正态分布，但是算法实现效率极低，后来经过 matlab 时间工具分析后，发现主要的时间开销主要就是循环中产生随机数的过程。后来将产生随机数的过程提出循环，转移概率采用均匀分布，时间开销大幅度减小，接收大概 10000 个点时的单次运行时间大约为 7~9s。这样得到的分布虽然比之前算出相关系数相对误差误差大，但是时间开销能节省 1/3 左右。

1.8 数值结果

在方法清楚之后，算法实现起来还是比较简单的。但刚开始时，算法实现效率极低，后来经过 matlab 时间工具分析后，发现主要的时间开销主要就是循环中产生随机数的过程。后来将产生随机数的过程提出循环，时间开销大幅度减小，接收大概 10000 个点时单次运行时间大约 10s。多次运行后，发现采样后相关系数和真实的相关系数的相对误差大概在 1%以内。画出仿真得到的二元正态分布图和实际对照如下：



这里要注意的是，在利用马氏链进行抽样时，在收敛之前的一段时间，各个状态的边际分布还不能认为是稳定分布，所以在进行估计的时候，应该把前面的这些个迭代值去掉，这个过程一般被称为“burn-in”。

1.9 讨论

实验中我们使用的是式 1.4.1 的接受方程，并假定$\boldsymbol{T}(\boldsymbol{y},\boldsymbol{x})=\boldsymbol{T}(\boldsymbol{x},\boldsymbol{y})$，这个和 Metropolis 最早提出的算法基本相同。这种方法可以在已知信息量最小的情况下，用非常高的效率采样出一个分布样本，实用性强且易于实现这个不变分布。

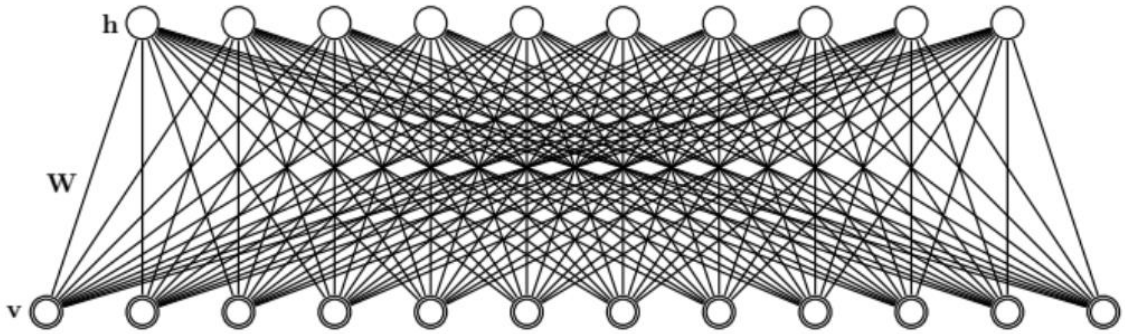
2 RBM 模型的归一化常数的估计

2.1 介绍

深度置信网络（Deep Belief Networks）是一种无监督学习下的机器学习模型，而受限波尔兹曼机（RNM）是深度学习中最重要最基础的模型之一。但是当数据量比较大时，RBM 的归一化系数计算量就会呈指数级增长以至于无法计算。所以这部分我们主要讨论用 AIS,SAMS,TAP 和 RTS 算法来实现对 RBM 的归一化系数的估计。

2.2 RBM 模型

受限波尔兹曼机是一种常用的无向图模型，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，模型的的参数为 $\theta=\{\mathrm{W},\mathrm{b},\mathrm{a}\}$。模型的能量为：

$$\mathrm{E}(\mathrm{v},\mathrm{h};\theta)=-\mathrm{v}^{\mathrm{T}}\mathrm{W}\mathrm{h}-\mathrm{b}^{\mathrm{T}}v-a^{T}h=-\sum_{i}\sum_{j}W_{ij}v_{i}h_{j}-\sum_{i}b_{i}\,v_{i}-\sum_{j}a_{j}h_{j}$$

概率分布为：

$$\mathrm{p}(\mathrm{v},\mathrm{h};\theta)=\frac{1}{\mathrm{Z}}e^{-E(v,h;\theta)} \qquad (2.2.1)$$

Z 为归一化系数，其定义为：

$$\mathrm{Z}=\sum_{v}\sum_{h}e^{-E(v,h;\theta)} \qquad (2.2.2)$$

输入层的边缘概率为：

$$\mathrm{p}(\mathrm{v})=\frac{1}{\mathrm{Z}}\sum_{h}e^{-E(v,h)} \qquad (2.2.3)$$

2.3 基本理论和方法

之前我们已经提到，在输入数据值极大时，直接用公式对归一化系数的真值进行计算变得及其复杂，所以，想到通过别的一些方法来对这个数进行估计。估计值的准确性可以通过重复实验求结果的方差来判断。这里我阅读文献资料后，主要实现了以下四种不同的 RBM 模型下的归一化系数估计。

2.3.1 Annealed Importance Sampling

AIS 算法的核心思想就是既然要求Z^B的 RBM 模型比较复杂，那么我们不妨自己设计一个比较简单的 RBM 模型，寻找Z^A和Z^B的关系，从而实现对Z^B值的估计。要求Z^B/Z^A, 我们在这里使用了简单重要性（SIS）采样的方法，并假设我们可以从这个简单的 RBM 中独立采样，通过一个简单的 Monte Carlo 估计，我们得到下面这个式子

$$\frac{Z_B}{Z_A}=\frac{1}{M}\sum_{i=1}^{M}=\frac{1}{M}\sum_{i=1}^{M}\omega^{(i)}=\widehat{r_{SIS}} \tag{2.3.1}$$

参考 Ruslan Salakhutdinov 的论文可知，在 AIS 模型中，

$$\frac{Z_K}{Z_0}=\frac{Z_1}{Z_0}\frac{Z_2}{Z_1}\frac{Z_3}{Z_2}\ldots\frac{Z_K}{Z_{K-1}} \tag{2.3.2}$$

假设P_K和P_{K-1}的边际分布足够接近，即可得到

$$\frac{Z_K}{Z_{K-1}}\approx\frac{1}{M}\sum_{i=1}^{M}\frac{P_k^*(x_k)}{P_{k-1}^*(x_k)} \qquad x^{(i)}\sim P_k$$

通过迭代上式可变形成为：

$$\frac{Z_B}{Z_A}\approx\frac{1}{M}\sum_{i=1}^{M}{\omega_{AIS}}^{(i)}=\widehat{r_{AIS}} \tag{2.3.3}$$

$$\omega_{AIS}=\prod_{k=1}^{K}\frac{P_k^*(x_k)}{P_{k-1}^*(x_k)} \tag{2.3.3}$$

之后只要计算出Z_A的值与$\widehat{r_{AIS}}$相乘即可。这样，我们通过一个简单的受限波尔兹曼机反复采样迭代，可以估算出另一个玻尔兹曼机的归一化常数。

2.3.2 Rao-Blackwellized Tempered Sampling

RTS 的实现原理与 AIS 较为相似，但也有部分的不同。RTS 的核心想法就是通过给定 v 和β_k的分布，采样出样本$\{x^{(i)},\beta_{k^{(i)}}\}$, 然后通过计算 Rao-Blackwellized form（Robert & Casella.2013）

$$\hat{c}_k=\frac{1}{N}\sum_{i=1}^{N}q(\beta_k|x^{(i)}) \tag{2.3.4}$$

其中，由 David E. Carlson (2016)的推导结果可得

$$q(\beta_k|x)=\frac{f_k(x)r_k/\hat{Z}_k}{\sum_{k'=1}^{K}f_{k'}(x)r_{k'}/\hat{Z}_{k'}} \tag{2.3.5}$$

归一化系数Z_k的准确值可以用一个估计值$\hat{Z}_k$表示为：

$$Z_k = \frac{\hat{Z}_k r_1}{r_k}\frac{q(\beta_k)}{q(\beta_1)} \quad k = 2,3,\dots,K \qquad (2.3.6)$$

带入$\hat{c}_k$即可得

$$\hat{Z}_k^{RTS} = \frac{\hat{Z}_k r_1}{r_k}\frac{\hat{c}_k}{\hat{c}_1} \quad k = 2,3,\dots,K \qquad (2.3.7)$$

2.3.3 Thouless-Anderson-Palmer Method

TAP 模型和 AIS 和 RTS 相比有很大的不同，它是一种从物理模型 he1 差分方程出发最后能够迅速准确估计归一化系数的一种方法。由式 2.2.1 取对数，我们可得

$$\ln \mathrm{P(v)} = -\mathrm{F^c}(v) + F \qquad (2.3.8)$$

这里的 F 物理角度表示这个 RBM 机的自由能。基于 Georges 和 Yedidia 曾提出的自旋粒子的自由能高温扩散的模型，我们可以得到一个自由能勒让德变换方程

$$-\beta\,\Gamma\,[\mathrm{m}] = -\beta\left(\mathrm{F}\left[\mathrm{q}^*[\mathrm{m}]\right] + \sum_i q_i^*[m]m_i\right) \qquad (2.3.9)$$

根据 Martlou Gabrie 的推导，在 q=0，$\beta = 1$时，我们的 RBM 机的自由能 F 满足下面这个式子

$$-\beta\mathrm{F} = -\beta\mathrm{F}[\mathrm{q}=0] = -\beta\,\Gamma\,[\mathrm{m}^*] \qquad (2.3.10)$$

$$\Gamma\,[\mathrm{m}^{\mathrm{v}}, m^h] \approx -S(m^v, m^h) - \sum_i a_i m_i^v - \sum_j b_j m_j^h$$

$$-\sum_{i,j} W_{ij} m_i^v m_j^h - \frac{\mathrm{W_{ij}^2}}{2}(m_i^v - (m_i^v)^2)\left(m_j^h - \left(m_j^h\right)^2\right) \qquad (2.3.10)$$

带入$\mathrm{m^v}$和$\mathrm{m^h}$求得自由能 F 的估计值，根据$\mathrm{F} = -\ln\mathrm{Z}$,即可求得归一化系数 Z 的估计值。

2.3.4 Self-adjusted Mixture Sampling

不同于之前 3 个算法，SAMS 算法可以通过几个不同的 RBM 模型来进行归一化系数估算。SAMS 方法的核心思路就是两步，包括 Local Jump 和 Monte Carlo。SAMS 算法的提出，主要是 Zhiqiang Tan（2014）针对之前 Labeled Mixtrue Sampling 中的 Globel Jump 复杂度太高，对整个过程做了优化和提高。在 LMS 采样中，我们把所有 RBM 机的概率分布合成一个联合分布：

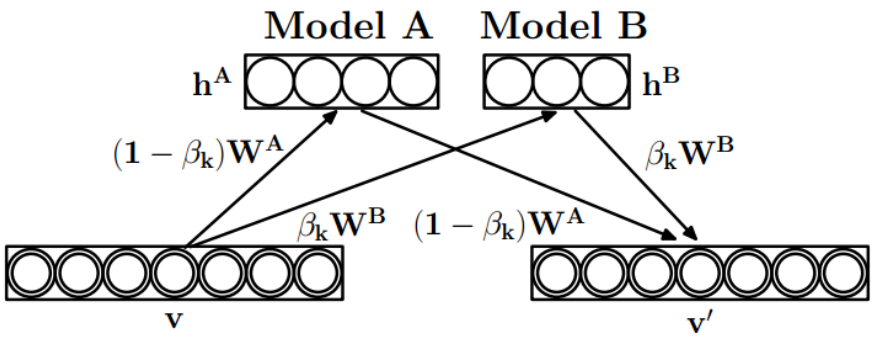
$$(\mathrm{L,X}) \sim \mathrm{p(j,x;\zeta)} \propto \frac{\pi_{\mathrm{j}}}{e^{\zeta}} q_j(x) \qquad (2.3.11)$$

其中 L 表示 RBM 机的 label, π是几种 RBM 机的权重占比分布，ζ是归一化系数取对数的估计值。（我们可以设定$\zeta_1 = 0$）

2.4 实现方法

2.4.1 AIS 的实现

因为我们是要实现从一个 RBM 机 A 到另一个 RBM 机 B 的跳转，所以我们需要一个参数表示二者的权重。这里我们取 β (β ∈[0,1] ,即 0≤ β 0≤ β 1≤ β 2≤ β 3 ……≤ β k≤1)。跳转示意图如下图所示：



用归一化后的概率密度函数表示为：

$$\mathrm{p}\left(h_j^A=1|\mathrm{v}\right)=\mathrm{g}\left((1-\beta_k)\left(\sum_i W_{ij}^A v_i+a_j^A\right)\right) \qquad (2.4.1)$$

$$\mathrm{p}\left(h_j^B=1|\mathrm{v}\right)=\mathrm{g}\left(\beta_k\left(\sum_i W_{ij}^B v_i+a_j^B\right)\right) \qquad (2.4.2)$$

$$\mathrm{p}(v_i'=1|\mathrm{h})=\mathrm{g}\left((1-\beta_k)\left(\sum_j W_{ij}^A h_j^A+b_i^A\right)+\beta_k\left(\sum_j W_{ij}^B h_j^B+b_i^B\right)\right) \qquad (2.4.3)$$

其中，归一化函数$\mathrm{g(x)}=\frac{1}{1-\mathrm{e}^{-\mathrm{x}}}$。由上面 3 个条件概率，给定$\beta_k$和初始的 v, 我们可以通过吉布斯采样获得新的 v 的采样值。获得 v 新的的采样之后，代入下式求得$P_k^*(v)$,

$$P_k^*(v)=\sum_{h^A,h^B} e^{(1-\beta^k)E(v,h^A;\theta^A)+\beta^k E(v,h^B;\theta^B)}$$

$$=\mathrm{e}^{(1-\beta^k)\Sigma_i b_i^A v_i}[\prod_{j=1}^{F_A}\left(1+e^{(1-\beta^k)(\Sigma_i W_{ij}^A v_i+a_j^A)}\right)]*\mathrm{e}^{\beta^k\Sigma_i b_i^B v_i}\prod_{j=1}^{F_B}\left(1+e^{\beta_k(\Sigma_i W_{ij}^B v_i+a_j^B)}\right)] \quad (2.4.4)$$

这里，我们选择一个最简单的 RBM 机$\theta_A=\{0,0,0\}$,带入式 2.2.3,即可得到

$$Z_A=\prod_j(1+e^{a_j})\prod_j(1+e^{b_j})$$

$$P_A(v)=\prod_i\frac{1}{1+e^{-b_i}} \qquad (2.4.5)$$

得到Z_A和$\widehat{r_{AIS}}$的值后，即可求得Z_B

2.4.2 RTS 的实现

实验中，参考 David E. Carlson (2016) 的论文，我们可以设定 $\mathrm{r_k}$ 为一常数，这样由式 2.3.7 我们可得

$$\hat{Z}_k^{RTS} = \hat{Z}_k \frac{\hat{c}_k}{\hat{c}_1} \quad k = 2,3,\dots,K \qquad (2.4.6)$$

我们初始可以将$\hat{Z}_k$全部设为 1，然后通过反复迭代$\hat{Z}_k$，其可得到较为精确的归一化系数估计值。需要注意的是，在计算$q(\beta_k|x)$时，要根据Z_A和β^k做相应的归一化处理（陈誉博 2016）。David E. Carlson (2016)提到的实现方法中β_k的每次采样都是根据$q(\beta_k|x)$的分布来进行吉布斯采样，我在实验中发现若用这种方法，β_k在几次采样后条件概率会集中到$\beta_k = 1$的分布上，实验结果不收敛，因此不能使用。分析原因，有可能是他当年处理的数据模型和我们现在使用的模型相比更为精确导致。由于没有办法获得更贴合待处理数据的 RBM 机，所以这次实验中我选择了使β_k从$\{\beta_1,\dots,\beta_k\}$中随机采样的方法。最后实验结果也证明这种方法是有效的。

2.4.3 TAP 的实现

TAP 方法最核心的部分就是要求出m^v和m^h的精确值，通过迭代法我们可以使得m^v和m^h不断逼近它们的精确值。迭代公式如下：

$$m_j^h[t+1] \leftarrow g(b_j + \sum_i W_{ij} m_i^v[t] - W_{ij}^2\big(m_j^h[t] - 0.5\big)(m_i^v[t] - (m_i^v[t])^2)) \qquad (2.4.7)$$

$$m_i^v[t+1] \leftarrow g(a_j + \sum_j W_{ij} m_j^h[t+1] - W_{ij}^2(m_i^v[t] - 0.5)\big(m_j^h[t+1] - (m_j^h[t+1]\big)^2)) \qquad (2.4.8)$$

2.4.4 SAMS 的实现

假设我们有 4 个 RBM 机 A,B,C,D，设他们所占的权重相等，即$\pi_j = \frac{1}{4}$，那么当前 RBM 机转移到与它 Label 临近的 RBM 机的概率如下表所示：

	A	B	C	D
A	0	1	0	0
B	0.5	0	0.5	0
C	0	0.5	0	0.5
D	0	0	1	0

我们将上图转移关系写成矩阵形式

$$\Gamma = \begin{pmatrix} 0.0 & 1.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.5 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{pmatrix}$$

然后我们从$\Gamma(L_{t-1},:)$求出下一个跳转至的 RBM 机的 Label—j,这一步用 MH 算法，接受概率如下：

$$\min\left\{1, \frac{\Gamma(j, L_{t-1})}{\Gamma(L_{t-1}, j)} * \frac{p(j|X_{t-1};\zeta)}{p(L_{t-1}|X_{t-1};\zeta)}\right\} \qquad (2.4.9)$$

然后，我们得到下一个 RBM 机的 Label 之后，就可以用之前 AIS 的方法做 Markov mov，产生这个 RBM 机下的X_t。

之后，我们需要更新 ζ 的值来不断精确这个估计值。根据 Zhiqiang Tan（2014）的理论，我们设定$\delta^{(t)} =$

$\{1\{L_t=1\},\ldots,1\{L_t=m\}\}^T$,然后有：

$$\zeta^{(t-0.5)}=\zeta^{(t-1)}+\gamma_t(\delta^{(t)}-\pi) \tag{2.4.10}$$

$$\zeta^{(t)}=\zeta^{(t-0.5)}-\zeta_1^{(t-0.5)} \tag{2.4.11}$$

其中$\gamma_t=t_0/\max(t_o,t)$. Liang et al.(2007) 建议$t_0\in(2m,100m)$.重复迭代多次后我们得到一个较为准确的归一化系数估计值。

2.5 方法分析

以上 4 种方法中，AIS 和 RTS 主要是利用一个简单的 RBM 模型 A 对另一个复杂的模型 B 归一化系数进行估计，精确度和算法效率都和你给的 v 初始分布和θ_A有关,所以可能需要大量的实验才能得到一个最优的算法。SAMS 和 AIS 以及 RTS 都稍有区别 ，这种算法同时观测了多个模型，并利用这些模型之间的内在联系通过 Metropolis-Hasting 算法获得稳态分布下的归一化系数。TAP 用了一个物理模型解决问题，思路比较清晰。

2.6 算法

2.6.1 AIS 的算法

Algorithm 2 AIS 算法

Select β^k with $0=\beta_0\le\beta_1\le\beta_2\le\beta_3\ldots\ldots\le\beta_k\le1$
Sample x1 from $P_A=P_0$
for k=1 **to** K-1 **do**
  Sample x_{k+1} given x_k using $T_k(x_{k+1}\leftarrow x_k)$
end for
Set $\omega_{AIS}=\prod_{k=1}^{K}P_k^*(x_k)/P_{k-1}^*(x_k)$

2.6.2 RTS 的算法

Algorithm 3 RTS 算法

Input: $\{\beta_k\}_{k=1,..K}$, N,M
Initialize $\log\hat{Z}_k,k=2,\ldots,K$
for m=1 **to** M **do**
Initialize $\beta\epsilon\{\beta_1,\ldots,\beta_k\}$
Initialize $\hat{c}_k=0,k=1,\ldots,K$
  for i=1 **to** N **do**
    Transition in x leaving $q(x|\beta)$ invariant
    Sample $\beta|x$ randomly from $\{\beta_1,\ldots,\beta_k\}$
    Update $\hat{c}_k\leftarrow\hat{c}_k+\frac{1}{N}q(\beta_k|x)$
  end for

Update $\hat{Z}_k^{RTS} \leftarrow \hat{Z}_k \frac{\hat{c}_k}{\hat{c}_1}$ $k = 2, \ldots, K$

end for

2.6.3 TAP 的算法

Algorithm 4 TAP 算法

Initialize $\mathrm{m_v}$,$\mathrm{m_h}$ $\leftarrow 1$
Initialize N
for k=1 **to** N **do**
Update $\mathrm{m_j^h} \leftarrow g(b_j + \sum_i W_{ij} m_i^v - W_{ij}^2 \left(m_j^h - 0.5\right)(m_i^v - (m_i^v)^2))$
Update $\mathrm{m_i^v} \leftarrow g(a_j + \sum_j W_{ij} m_j^h - W_{ij}^2 (m_i^v - 0.5)\left(m_j^h - (m_j^h)^2\right))$
end for
Set $logZ \leftarrow -\Gamma\,[\mathrm{m^v}, m^h]$

2.6.4 SAMS 的算法

Algorithm 5 SAMS 算法

input Gamma矩阵，N
input $\theta_A\{W_A, b_A, a_A\}, \theta_B\{W_B, b_B, a_B\}, \theta_C\{W_C, b_C, a_C\}, \theta_D\{W_D, b_D, a_D\}$
Initialize $\mathrm{X_t}, \zeta_k$
for k=1 **to** N **do**
Generate j from prob $\min\left\{1, \frac{\Gamma(\mathrm{j}, \mathrm{L_{t-1}})}{\Gamma(\mathrm{L_{t-1}}, \mathrm{j})} * \frac{p(j|X_{t-1};\zeta)}{p(L_{t-1}|X_{t-1};\zeta)}\right\}$
Generate $\mathrm{X_t} \sim \boldsymbol{\psi}_{\mathrm{L_t}}(X_{t-1},)$
Update $\zeta^{(\mathrm{t}-0.5)} \leftarrow \zeta^{(\mathrm{t}-1)} + \gamma_{\mathrm{t}}\left(\delta^{(\mathrm{t})} - \pi\right)$
Update $\zeta^{(\mathrm{t})} \leftarrow \zeta^{(\mathrm{t}-0.5)} - \zeta_1^{(\mathrm{t}-0.5)}$
end for

2.7 算法分析及数值结果

实验中，我先开始实现的是 AIS 算法，参考了 Ruslan Salakhutdinov 在论文后留下的源代码，发现其算法已经对循环算法做了极大的优化，时间复杂度为 O(K)（K 为取β^k） 的个数。这种算法最后得出的估计值的准确度与取β^k的个数正相关，但是时间开销也会对应增大。折中起见，我的β^k在[0,1]均匀分布间取了 10000 个点，最后 AIS 实验结果如下表：

使用模型	logZ	方差	单次运行时间	20 次运行时间
$\boldsymbol{h_{10}}$	226.1	0.2929	40s	587s
$\boldsymbol{h_{20}}$	220.9	0.0098	38s	981s
$\boldsymbol{h_{100}}$	348.33	0.0103	54s	1522s
$\boldsymbol{h_{500}}$	459.3	0.3282	138s	2783s

RTS 算法的实验结果如下。David E. Carlson 文献上来说 RTS 的算法比 AIS 相比更优秀更稳定。但是遗憾是，我的代码并没有很好的实现作者的构想，大致是因为 matlab 对循环算法的优化度还不够，导致时间复杂度和理论有差距。

使用模型	logZ	方差	单次运行时间	20 次运行时间
h_{10}	225.0	0.7735	27s	565s
h_{20}	218.2	1.0129	32s	643s
h_{100}	349.1	1.0251	58s	1084s
h_{500}	453.0	5.8591	184.5	3504s

TAP 算法的实验结果如下。算出来的结果和其他算法比都偏小，但是收敛速度和代码效率极快，时间复杂度只有 O(n)，n 为迭代次数，因为迭代速度极快，所以速度最快。

使用模型	logZ	方差	单次运行时间	20 次运行时间
h_{10}	212	10e-26	1.8s	28s
h_{20}	203	10e-27	2.1s	41s
h_{100}	341	10e-27	5.4s	110s
h_{500}	449	10e-6	67s	1307s

SAMS 算法的实验结果如下,算出 4 个模型相对于Z_{h10}的值。SAMS 算法收敛速度极慢，迭代 40W 左右次后结果才能收敛。

使用模型	logZ（相对Z_{h10}）	方差	单次运行时间	20 次运行时间
h_{10}	0	10e-5	407s	6971s
h_{20}	-6.9	1.089	407s	6971s
h_{100}	121	0.003	407s	6971s
h_{500}	233	0.345	407s	6971s

实验最后，我使用 AIS 算法求出的归一化系数进行似然值计算，得出的结果如下：

	h_{10}	h_{20}	h_{100}	h_{500}
似然值（取对数）	-173.9	-145.0	-113.0	-99.9

从似然值可以看出，随着 RBM 机隐变量数量的增加，即 RBM 机越复杂，似然值越大，我们的估计值就越不准确。这说明随着 RBM 机模型复杂度的升高，我们对它的归一化系数的估计难度就越高。

2.8 讨论

4 种算法综合而言，AIS 和 RTS 的准确度较高，收敛速度中等，在这个实验中最为合适。我最终选择了 AIS 算法算出来的归一化系数进行似然值计算。另外俩种，TAP 在这种数据模型下的准确度太低，但和其他算法相比，其速度是一大优势，在我们只想得到归一化系数的粗略估计时，可以使用这种方法。SAMS 算法由于是同时对多个模型的归一化系数进行计算，所以迭代次数多，时间长。综合精确度、收敛速度和算法效率考虑，AIS 算法在这次实验中应该是最优的。

3 总结

这次的大作业主要探讨了和深度学习相关的 MH 算法以及 RBM 模型。深度学习是近些年比较热点的一个研究方向。RBM 模型是深度学习和多层神经网络一个最为基础的模型，通过这次的学习我基本掌握了 RBM 模型的基础概念和运作模式，学会了不同的算法对 RBM 的归一化系数进行估计，收获良多。

致 谢

衷心感谢欧智坚老师和助教平时课上和课后对我的悉心指导教育，让我对随机过程产生了极大的兴趣，学到了扎实的基础知识。同时也要感谢无 42 班陈誉博、胡昌然同学对本人实验过程中的大力帮助。

参考文献

[1] Ruslan Salakhutdinov. Learning Deep Generative Models. 2009.

[2] Marylou Gabrie ,Eric W. Tramel, Florent Krzakala. Training Restricted Boltzmann Machines via the Thouless-Anderson-Palmer Free EnergyLearning Deep Generative Models.

[3] Jun S. Liu. Monte Carlo Strategies In Scientific Computing. 2001.

[4] David E. Carlson. Partition Functions from Rao-Blackwellized Tempered Sampling.

[5] Zhiqiang Tan. Optimally adjusted mixture sampling and locally weighted histogram analysis. 2014.

[6] 林书烈. 《应用随机过程》. *P115-P116*

[7] 卯诗松，程依明. 《概率论数理统计教程》. *P148-P153*

[8] 陆大金. 《随机过程及其应用》.

[9] Christopher M. Bishop F.R.Eng. Pattern Recognition and Machine Learning.