

# Markov Chain Monte Carlo

## Introduction, Comparison & Analysis

Tzu-Heng Lin, 2014011054, W42

Department of Electronic Engineering, Tsinghua University, Beijing, China  
lzhbrian@gmail.com

### ABSTRACT

Markov Chain Monte Carlo (MCMC) is a technique to make an estimation of a statistic by simulation in a complex model. Restricted Boltzmann Machine(RBM) is a crucial model in the field of Machine Learning. However, training a large RBM model will include intractable computation of the partition functions, i.e.  $Z(\theta)$ . This problem has aroused interest in the work of estimation using a MCMC methods. In this paper, we first conduct Metropolis-Hastings Algorithm, one of the most prevalent sampling methods, and analyze its correctness & performance. We then implement three algorithms: TAP, AIS, RTS, to estimate partition functions of an RBM model. Our work not only give an introduction about the available algorithms, but systematically compare the performance & difference between them. We seek to provide an overall view in the field of MCMC.

### 1. INTRODUCTION

**Markov Chain** A Markov Chain is a special stochastic process in which the current state only depends on its previous state, we call this property the Markov property or memorylessness. i.e.

$$P(X_{t+1} = x | X_t, X_{t-1}, \dots) = P(X_{t+1} = x | X_t) \quad (1)$$

Given a Markov Chain, if a vector  $\pi$ , has the property:

$$\pi = \pi P \quad (2)$$

then we call  $\pi$  the stationary distribution, it denotes the final state distribution of the stochastic process in a Markov Chain.

Markov Chain has a wide application in many fields in the real world, such as social science[1], economics & finance[4] and of course, computer science[10], etc.

**Markov Chain Monte Carlo** Given a probability distribution  $p(x)$ , it would be a great thing if we could generate some samples of it by a simple method, so here comes the Markov Chain Monte Carlo(MCMC). If we could construct a Markov Chain which its stationary distribution  $\pi$  just equals to  $p(x)$ , then we could use this Markov Chain to sample from this distribution  $p(x)$ . And this is the main idea of MCMC

In this paper, we implement the Metropolis-Hastings Algorithm, which is one of the most widely used sampling method. We also deal with the accepting rate, which I will introduce later, for previous work[11] have shown that the accepting rate may influence the result of the experiment and there is theoretical support in choosing an optimal accepting rate.

**Restricted Boltzmann Machine** A Restricted Boltzmann Machine (RBM)[7] is a significant work bringing hypothesis in statistical physics to computer science. By stacking several layers of RBM, we will get a fundamental model, Deep Belief Network[6], in the field of Deep Learning, which is nowadays the hottest class of algorithms used in Machine Learning.

**Estimating Partition functions** In the process of training an RBM, however, will include intractable computation of the partition function. When the model grows large, the complexity of this work will be incompletion. The good news is, several researches have shown that there are ways to avoid this by using an MCMC approach instead, to estimate it.

In this paper, we implement three prevalent MCMC methods of estimating a partition function, Thouless-Anderson-Palmer Sampling(TAP)[3], Annealed Importance Sampling(AIS)[9, 12], Rao-Blackwellized Tempered Sampling(RTS)[2], respectively, and give an overall comparison on the theory & performance between them.

### 2. RELATED WORK

**Monte Carlo Sampling Method**  
**Partition Function Estimation**

...

### 3. METROPOLIS-HASTINGS

In the Introduction, we have shown the main idea of an MCMC sampling method. In this section, we will introduce the Metropolis Hastings Algorithm and conduct an experiment.

#### 3.1 Algorithm<sup>1</sup>

##### 3.1.1 Detailed Balance Condition

Before stepping further into the MH Algorithm, We would first introduce a theorem called the Detailed Balance Condition.

In the introduction, we said that we want to construct a Markov Chain which its stationary distribution  $\pi(x)$  just equals to the required probability distribution  $p(x)$ .

At first, a theorem is needed.

**THEOREM 3.1 (DETAIL BALANCE CONDITION).** *Given a non periodic Markov Chain, if*

$$\pi(i)P_{ij} = \pi(j)P_{ji} \text{ for all } i, j \quad (3)$$

*then  $\pi(x)$  is the stationary distribution of this Markov Chain.*

##### 3.1.2 MCMC sampling method

Suppose we already have a transition matrix  $Q$  for a Markov Chain,  $q(i, j)$  denote the probability of transition from state  $i$  to state  $j$ . For the general case,

$$p(i)q(i, j) \neq p(j)q(j, i)$$

That is to say, we do not have the detailed balance condition (Theorem 3.1) So we introduce an  $\alpha(i, j)$  s.t.

$$p(i)q(i, j)\alpha(i, j) = p(j)q(j, i)\alpha(j, i) \quad (4)$$

By symmetrical characteristic, we choose:

$$\alpha(i, j) = \min\left\{\frac{p(j)q(j, i)}{p(i)q(i, j)}, 1\right\} \quad (5)$$

So the new Markov Chain  $Q'$  would have the property of which its stationary distribution is  $p(x)$

$$p(i) \underbrace{q(i, j)\alpha(i, j)}_{Q'(i, j)} = p(j) \underbrace{q(j, i)\alpha(j, i)}_{Q'(j, i)} \quad (6)$$

We call the  $\alpha(i, j)$  we introduced, accepting ratio. It means that, in the original Markov Chain  $Q$ , when state  $i$  transits to state  $j$  with a probability of  $q(i, j)$ , we accept this transition with a probability of  $\alpha(i, j)$

Now, we have derived the MCMC sampling method.

##### 3.1.3 Metropolis-Hastings Algorithm

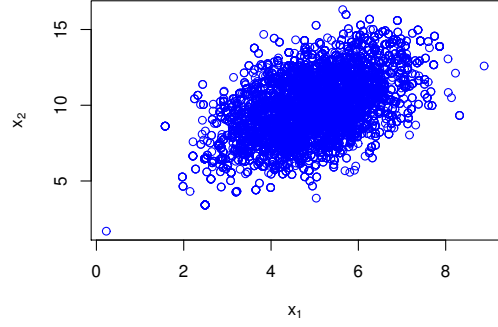
The MCMC sampling method is a marvellous work. However, it has a critical drawback that if  $\alpha(i, j)$  &  $\alpha(j, i)$  are too small, we would seldom accept the transition.

A solution is that we multiply both  $\alpha(i, j)$  &  $\alpha(j, i)$  with a constant to make sure that the larger one between them equals 1. By doing so, we change the accepting ratio to

$$\alpha(i, j) = \min\left\{\frac{p(j)q(j, i)}{p(i)q(i, j)}, 1\right\} \quad (7)$$

and we get Metropolis-Hastings Algorithm[5].

<sup>1</sup>Available at [https://github.com/lzhbrian/MCMC/blob/master/metropolis\\_hastings/metropolis\\_hasting.R](https://github.com/lzhbrian/MCMC/blob/master/metropolis_hastings/metropolis_hasting.R) in Matlab



**Figure 1: Sampling result of 10,000 points correlation = 0.500873 , set sd.T = 3.0**

##### 3.1.4 Symmetric Case

In a Markov Chain whose transition matrix is symmetric, we have

$$q(i, j) = q(j, i)$$

so the accepting ratio could be simplified to

$$\alpha(i, j) = \min\left\{\frac{p(j)}{p(i)}, 1\right\} \quad (8)$$

which is also known as the Metropolis Algorithm[8].

##### 3.1.5 Continual Case

In a continual Markov Chain, such as the experiment we are going to do in the next subsection, we have a vague definition of transition matrix  $Q$ . So we introduce a concept called the proposal jump size,  $sd.T$ .

The method we get  $x_{k+1}$  from  $x_k$  is to add a sampled point of a normal distribution with a variance of the jump size and  $\mu = 0$ . For a two dimensional example, we have:

$$x_{k+1} = x_k + sd.T \begin{pmatrix} norm_1 \\ norm_2 \end{pmatrix} \quad (9)$$

### 3.2 Sampling Results

In our work, we use an example of a bivariate Normal distribution, with

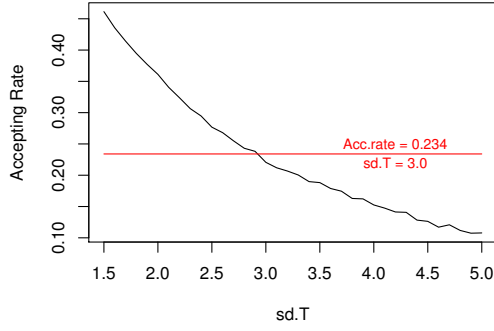
$$\mu = \begin{pmatrix} 5 \\ 10 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}$$

By theoretical computation, we can easily compute the the pearson correlation between the two dimensional value is 0.5.

$$\rho = 0.5$$

We then generate 10,000 samples using the MH algorithm, setting the standard deviation of proposal to 3.0. We can see from the result (Figure 1) that we have derived 10,000 sampled points with pearson correlation = 0.5, which matches the theoretical value.

### 3.3 Performance Analysis



**Figure 2: Accepting rate on different proposal jump size.**

### 3.3.1 Choice of proposal jump size

MH algorithm is an effective MCMC method for many diverse problems. However, its efficiency depends crucially on the selection of the proposal density. With the proposal jump size being small, the accepting rate would be very low and eventually stick to only one point (eg. the initial point); When the proposal jump size is too big, the accepting rate would be too high.

Roberts et al. have shown in previous work[11] that the optimal accepting rate of the MH algorithm should approximately be at 0.234 for the case of an N-dimensional Gaussian target distribution. We test the accepting rates in different proposal jump size (Figure 2) and find that the optimal value should be at approximately 3.0 to acquire a model with accepting rate being close to 0.234. That is the reason why we choose 3.0 as our proposal jump size.

### 3.3.2 Efficiency

Due to the limit of the accepting rate, for a high dimensional condition, using the MH sampling methods may spend more time in traverse all of the possible states, which could sometimes be less satisfying. Thus, many would switch to Gibbs Sampling Algorithm.

## 3.4 Gibbs Sampling

Gibbs Sampling is a special case of Metropolis Hastings Algorithm, by letting the accepting rate = 1, we will get a Gibbs Sampler. As the length & time limit, we will not specify more here.

## 4. PARTITION FUNCTION ESTIMATION

### 4.1 Restricted Boltzmann Machine

Restricted Boltzmann Machine ... However, calculating partition functions has always been a ...

### 4.2 Algorithms

#### 4.2.1 Thouless-Anderson-Palmer Sampling<sup>2</sup>

#### 4.2.2 Annealed Importance Sampling<sup>3</sup>

#### 4.2.3 Rao-Blackwellized Tempered Sampling<sup>4</sup>

#### 4.2.4 Other method

There are many other methods which can also estimate the partition functions. Such as Self-adjusted mixture sampling (SAMS)[13] proposed a method to estimate multiple partition functions together to improve the efficiency. As the length & time limit, we only implement 3 methods here in this paper.

### 4.3 Estimating Results

### 4.4 Performance Analysis

#### 4.4.1

<sup>2</sup>Available at <https://github.com/lzhhbrian/MCMC/blob/master/rbm/TAP.m> in Matlab

<sup>3</sup>Available at <https://github.com/lzhhbrian/MCMC/blob/master/rbm/AIS.m> in Matlab

<sup>4</sup>Available at <https://github.com/lzhhbrian/MCMC/blob/master/rbm/RTS.m> in Matlab

## 5. CONCLUSION

In this paper, we discuss about the Markov Chain Monte Carlo sampling method which are now undoubtedly one of the most important sampling methods.

We systematically compare three methods of partition function estimation which are crucial works in training a Restricted Boltzmann Machine or a Deep Belief Network.

As future work, we would like to join more methods to the comparison and if could, propose some improvement to the algorithms available.

## 6. ACKNOWLEDGEMENT

I would like to thank Yubo Chen, Liren Yu, Yuanxin Zhang, XueChao Wang, Changran Hu, for the discussion with me on the algorithms. Without them, I wouldn't have the possibility to accomplish this work in such a short time. This paper is a project of Stochastic Process Course in Tsinghua University, taught by Prof. Zhijian Ou.

## 7. REFERENCES

- [1] D. Acemoglu, G. Egorov, and K. Sonin. Political model of social evolution. *Proceedings of the National Academy of Sciences*, 108(Supplement 4):21292–21296, 2011.
- [2] D. Carlson, P. Stinson, A. Pakman, and L. Paninski. Partition functions from rao-blackwellized tempered sampling. *arXiv preprint arXiv:1603.01912*, 2016.
- [3] M. Gabri  , E. W. Tramel, and F. Krzakala. Training restricted boltzmann machine via the thouless-anderson-palmer free energy. In *Advances in Neural Information Processing Systems*, pages 640–648, 2015.
- [4] J. D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, pages 357–384, 1989.
- [5] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [6] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [7] J. L. McClelland, D. E. Rumelhart, P. R. Group, et al. Parallel distributed processing, 1987.
- [8] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- [9] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.
- [10] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.
- [11] G. O. Roberts, A. Gelman, W. R. Gilks, et al. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.
- [12] R. Salakhutdinov. *Learning deep generative models*. PhD thesis, University of Toronto, 2009.
- [13] Z. Tan. Optimally adjusted mixture sampling and locally weighted histogram analysis. *Journal of Computational and Graphical Statistics*, (just-accepted), 2015.