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 Bolztmann 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(θ). 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

Markov Chain Monte Carlo

Restricted Boltzmann Machine A Restricted Bolztmann Machine (RBM)[4] 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[3], 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 incontractable computation of the partition function. In this paper, we implement three prevalent methods of estimating a partition function, Thouless-Anderson-Palmer Sampling(TAP)[2], Annealed Importance Sampling(AIS)[5, 7], Rao-Blackwellized Tempered Sampling(RTS)[1], respectively, and give an overall comparison on the theory & performance between them.

2. RELATED WORK

Monte Carlo Sampling Method

Partition Function Estimation

...

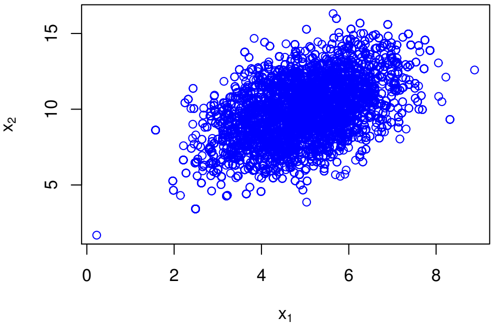

Figure 1: Sampling result of 10,000 points correlation = 0.500873 , set sd.T = 3.0

3. METROPOLIS-HASTINGS

Metropolis Hastings(MH) Algorithm

3.1 Algorithm[1]

3.1.1 Detailed Balance Condition

3.1.2 General Case

3.1.3 Symmetric Case

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.

[1] Available at https://github.com/lzhbrian/MCMC/blob/master/metropolis_hastings/metropolis_hasting.R in Matlab

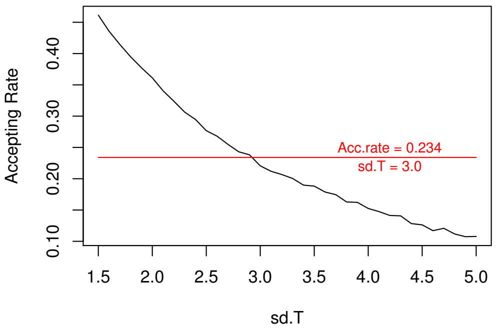

Figure 2: Accepting rate on different proposal jump size.

3.3 Performance Analysis

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[6] 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 appoximately 3.0 to acquire a model with accepting rate being close to 0.234.

3.4 Gibbs Sampling

For a high dimensional condition, because of the limit of accepting rate, the efficiency of MH algorithm is not satisfying, so many would switch to Gibbs Sampling Algorithm.

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 Bolztmann Machine

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

4.2 Algorithms

4.2.1 Thouless-Anderson-Palmer Sampling[2]

4.2.2 Annealed Importance Sampling[3]

4.2.3 Rao-Blackwellized Tempered Sampling[4]

4.2.4 Other method

There are many other methods which can also estimate the partition functions. Such as Self-adjusted mixture sampling(SAMS)[8] 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

[2] Available at https://github.com/lzhbrian/MCMC/blob/master/rbm/TAP.m in Matlab

[3] Available at https://github.com/lzhbrian/MCMC/blob/master/rbm/AIS.m in Matlab

[4] Available at https://github.com/lzhbrian/MCMC/blob/master/rbm/RTS.m in Matlab

5. CONCLUSION

In this paper, we discuss about the Monte Carlo sampling method which are now undoubtedly one of the most important sampling methods. We systematically compare three methods of partition function estimation. 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. Carlson, P. Stinson, A. Pakman, and L. Paninski. Partition functions from rao-blackwellized tempered sampling. *arXiv preprint arXiv:1603.01912*, 2016.

[2] M. Gabrié, 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.

[3] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[4] J. L. McClelland, D. E. Rumelhart, P. R. Group, et al. Parallel distributed processing, 1987.

[5] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[6] 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.

[7] R. Salakhutdinov. *Learning deep generative models*. PhD thesis, University of Toronto, 2009.

[8] Z. Tan. Optimally adjusted mixture sampling and locally weighted histogram analysis. *Journal of Computational and Graphical Statistics*, (just-accepted), 2015.