Try to use variataional inference to accerate MCMC

Ke (Becky) Bai

December 16, 2018

Abstract

Many reasons restrict MCMC from quick convergence. This paper discusses two important factors. The first problem is the burn-in time to find a good initial point for the MCMC chain for high dimensional data. The second one is the multimodality which spends the chain a long time to move among modes. To solve the first problem, we could use the variational method to do the initialization. For the second problem, we could use parallel MCMC.

1 Introduction

Suppose we have an unnormalized distribution with explicit form $p(x)$ and we want to sample from this distribution, we are faced with several difficulties; How to guarantee the sampled points are independent? How to guarantee the samples can represent the whole distribution? To solve these problems, variational Inference(VI) and Markov Chain Monte Carlo (MCMC) are two famous sampling methods.

These two methods both have strengths and weaknesses. From the perspective of the algorithms. The variational method seeks to use a simple distribution, which is easy to sample from, to approximate the target distribution. With this approximate distribution, we can easily calculate the KL-distance between the approximate distribution and the target distribution and use this distance to evaluate the convergence of the approximation. However, in order to calculate the KL divergence, we need to get the ratio of the approximate distribution and target distribution, which will bring huge variances to estimation. MCMC methods has the convergence guarantee only if we can wait long enough. It can find the correlation between each dimension of the data while the variational method usually assumes each dimension is independent. However, when dealing with complex distributions, It is especially hard to move from one mode to another. It is hard to calculate whether the algorithm has converged.

From the perspective of parallelization, the variational method is more efficient than the current MCMC methods. Because the former one can directly generate samples without the time-sequence requirement.

In this paper,Wewill first describe a new variational inference method proposed recently ([1]) and use it as an initialization method for MCMC sampling. In the third paragraph,We will introduce the parallel MCMC method, which is a very efficient way to accelerate the convergence. In the fourth part,We will discuss the possibility of the combination of these two algorithms.

For the experimental part, We firstly do some toy examples to explore strengths and weaknesses of each task. During the experiment, We found that things are not fully going on the direction I anticipate. MCMC works better and more efficient than we expected. Especially on the experiments with low dimension.

2 Semi-Variational Inference (SIVI)

As I mentioned above, we need to get an approximation of the true target distribution $q(\boldsymbol{x})$, which can be any distribution with explicit forms. The approximate distribution is $q(\boldsymbol{x}|\boldsymbol{\psi})$, where $\boldsymbol{\psi}$ is the variational parameter to be inferred in traditional case. Usually, it is the parameter of mean-field Gaussian distribution. In a recent paper [1], the author proposes a hierarchy structure called semi-variational inference (SIVI). The $\boldsymbol{\psi}$ are treated as a random variable $\boldsymbol{\psi} \sim q_{\phi}(\boldsymbol{\psi})$. When the distribution of the random variable degenerates to the point density function, SIVI reduces to ordinary variational inference.

SIVI has two advantages, the first advantage is that it is a more general distribution. Since the marginal distribution of $\boldsymbol{z} \sim \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\psi})}\, q(\boldsymbol{x}|\boldsymbol{\psi})$ is usually implicit. Another advantage is that we can still calculate KL divergence between the target distribution and the approximate distribution using $\mathbb{E}_{\boldsymbol{\psi}\sim q_{\boldsymbol{\phi}(\boldsymbol{\psi})}}\, \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{\psi})} \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x}|\boldsymbol{\psi})}$.

In the implementation, $\boldsymbol{\psi} = \boldsymbol{T}_{\phi}(\boldsymbol{\epsilon})$, $\boldsymbol{\epsilon} \sim q(\boldsymbol{\epsilon})$. $\boldsymbol{T}$ is a deterministic transfer like a neural network, ϕ is the parameters of this mapping to be learned like the weight parameters of the neural network. $q(\boldsymbol{\epsilon})$ is random variables (Multivariate Normal).

$q_{\boldsymbol{\phi}}(\boldsymbol{\psi})$ are easily to degenerate to the point density function. The details can be found in Proposition 1 of the original paper. To prevent from this, the authors added an extra term to KL-distance loss:

$$B_k = \mathbb{E}_{\boldsymbol{\psi},\boldsymbol{\psi}^{(1)},\cdots\boldsymbol{\psi}^{(n)}\sim q_{\phi}(\psi)} \frac{q(\boldsymbol{z}|\boldsymbol{\psi}) + \sum_{k=1}^{K} q(\boldsymbol{z}|\boldsymbol{\psi}^{(k)})}{K+1} \tag{1}$$

The main idea behind this is to sample multiple hyper-parameters ψ to calculate the loss at the same time in order to prevent ψ from collapse into a single value.

3 Parrallel MCMC

3.1 Proposal distribution

In parallel MCMC([2]), the authors tried to use different chains to capture different modes, which can also be understood as a partition of the variables that we need to estimate. The authors choices Langevin diffusion $d\theta_t = \sigma^2/2\nabla \log \pi(\theta_t)dt + \sigma dW_t$ as the proposal distribution, where W_t is standard Brownian motion and σ is much smaller than one. Since there is a gradient term in the function, the Langevin diffusion could find the nearby mode quickly but hard to transfer from one mode to another.

3.2 Weight Estimation

Adaptive importance sample and ratio estimation

The authors used adaptive importance sampling to estimate the weight. Firstly, they need to calculate the normalizer of each distribution according to

$$\hat{c}_j = T^{-1} \sum_{t=1}^{T} g(\theta_t) \mathbb{1}_{\Theta_j}(\theta_t) / q_j(\theta_t) \tag{2}$$

Use the weight can be calculated by $\hat{w}_{j,n} = \sum_{i=1}^{n} \hat{c}_j^{(i)} / \big[\sum_{i=1}^{n} \sum_{k=1}^{J} \hat{c}_k^{(i)} \big]$.

KL-Divergence and gradient descent

Here, I propose a similar method to estimate the weight. Since the generated samples are mostly centralized to one mode. We could use a Gaussian distribution or a student-t distribution to approximate the local distribution. In this way, we can approximate the whole distribution as a weighted form

$$\hat{\pi}(\theta) = \sum_j w_j \hat{\pi}_j(\theta_j) \tag{3}$$

Where $\hat{\pi}_j(\theta_j)$ is the distribution with explicit form.

Then we can easily calculate the KL-divergence between the approximate distribution and the target unnormalized distribution. And use gradient descent on w to minimize this distance to match the two distributions. This can also be understood as a one-layer neural network whose inputs are the estimated distribution from different chains and the output is the estimation of the target distribution. In this way, we can directly get w_i without the local normalizer. If the sum of the weight is larger than one, we can directly normalize it. Another advantage is that we can update these weights on-line. The KL-divergence can help us to quantify the convergence of the distribution as well.

The reason we choices the KL divergence metric is its form; $KL(P_\theta||Q)$. When q is an unnormalized distribution $Q = CQ$, the gradient of θ will not change because $KL(P_\theta||CQ) = KL(P_\theta||Q) - E_{P_\theta} \log C = KL(P_\theta||Q) - \log C$

4 Combination of VI and MCMC

There has been already some paper tries to bridge variational inference and MCMC[3][4].

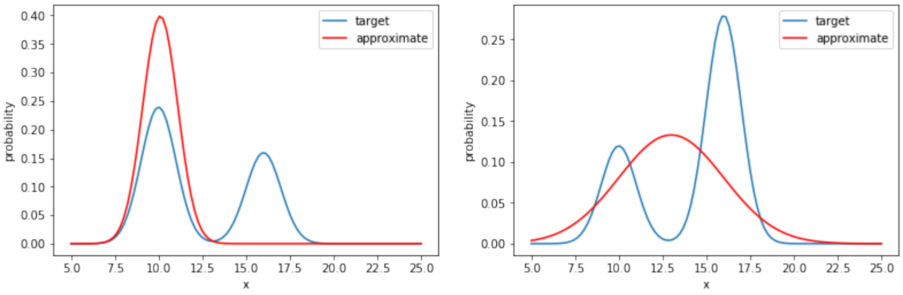
Optimization methods are often used to find a good initial point of MCMC. In this paper, the variational inference can be treated as an optimization tool. However, rather than other optimal method which aims to find the maximum point, variational method also take care of the whole distribution like the multiple modes, which will be a better choice.

In the degenerated case where we use a single Guassian distribution to approximate a complex distribution, we could adjust the variance of the approximate distribution to meet different needs. Consider the target distribution is a two-mixture Gaussian distribution. If these two distributions are far from each other. The approximate Gaussian distribution will capture the one mode when the variance is small (Fig 1(a)). It will capture the whole distribution when the variance is large (Fig. 1(b)). In SIVI, we treat the parameter as a random variable, which is a more general case of VI. So we can have a better estimation of the distribution.

The inspiration is that MCMC works very well on low dimensional tasks. MCMC can capture the local details of a distribution very well. When the dimension of the distribution is low, the convergence rate of MCMC is much quicker than variational inference method. In the high dimensional case, variational methods are more efficient, but the approximate distribution tend to neglect many details. Parallel MCMC can help this.

Assume we have already used the variational method a capture different modes in a complex distribution. When chooses the initial point, we could use the clustering algorithm like ”K-means” to choice K initial points, we could also choose the points who owns the largest distance between each other.

With good initial points, parallel MCMC could converge quickly.



(a) σ=1

(b) σ=3

Figure 1:

5 Experiment

5.1 plan

As far, I have done experiments on 1d, 2d and logistic regression problem using SIVI and MCMC. The next step is to use parallel MCMC to do these experiments.

5.2 1d distribution

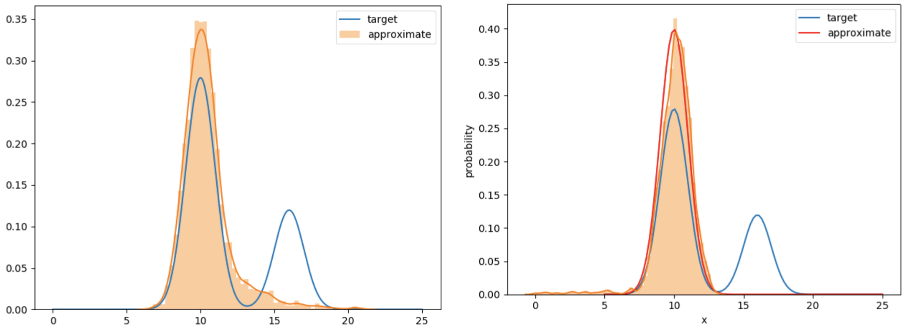
For 1d problem, we set two problem settings. Both of them are two-mixture Gaussian distributions. The difference is that one is located at coordinate original point. The other is far from the coordinate original point.

$$P_1 = 0.3 * N(-2, 1) + 0.7 * N(2, 1) \tag{4}$$

and

$$P_2 = 0.7 * N(10, 1) + 0.3 * N(16, 1) \tag{5}$$

For distribution P_1, both of these two methods perform very well and converge quickly (Not Shown here). For distribution P_2, we can see the small difference between this two distributions from (Fig. 2). MCMC covers one mode after 50k iterations. SIVI's tail has already covered another mode after 5k iterations. If MCMC algorithms can start with the points from the tail of SIVI. We can easily get a good estimation.



(a) SIVI

(b) MCMC

Figure 2: The left one is SIVI model, the right one is MCMC model.

5.3 2d distribution

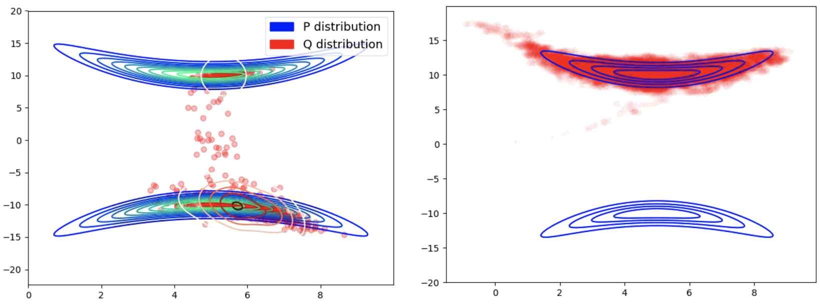
In this 2d case in (Fig. 4). SIVI could capture two modes, but it cannot fit the origin distribution well. MCMC model has 50% to reach one of two modes. We can anticipate that parallel MCMC will work very well.

5.4 Regression

We choices two datasets, one is waveform, this is a small dataset. The feature number is 22. Training size is 400, testing size is 4600. The other is a9a, which is a larger one. The feature number is 122. Training size is 32561. Testing size is 16281. The feature of a9a dataset is binary.

For the smaller dataset, the performance of these two algorithms is nearly the same. We get this conclusion from observing the same distribution of the output of the test samples (Fig. 4(a)). The accuracy on the test set are also the same for these two methods.

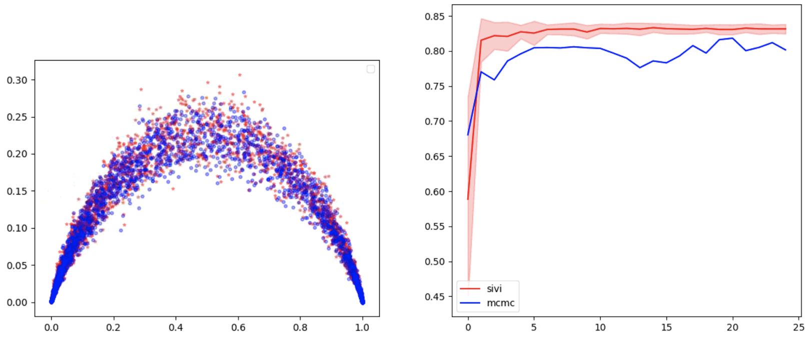
For the larger dataset, we observe that the MCMC method have difficulties. As we can see from (Fig. 4(b)), SIVI quickly converge to a point that MCMC cannot reach.



(a) SIVI

(b) MCMC

Figure 3: The left one is SIVI model, the right one is MCMC model.



(a) SIVI

(b) MCMC

Figure 4: (a)Distribution of the output of The red dots represent SIVI. The blue dots represent MCMC. The x axis means the probability to be a positive sample. y axis represents the standard deviation. (b)Accuracy on the test set of a9a.

6 Conclusion

In the low dimensional case, MCMC is more efficient and accurate than variational methods. When the dimension enlarges, variational is more likely to capture the whole distribution roughly, which could be good initial points for parallel MCMC.

References

[1] Mingzhang Yin and Mingyuan Zhou. Semi-Implicit Variational Inference. *arXiv:1805.11183 [cs, stat]*, May 2018. arXiv: 1805.11183.

[2] Douglas N. VanDerwerken and Scott C. Schmidler. Parallel Markov Chain Monte Carlo. *arXiv:1312.7479 [stat]*, December 2013. arXiv: 1312.7479.

[3] Daniel Levy, Matthew D. Hoffman, and Jascha Sohl-Dickstein. Generalizing Hamiltonian Monte Carlo with Neural Networks. *arXiv:1711.09268 [cs, stat]*, November 2017. arXiv: 1711.09268.

[4] Tim Salimans, Diederik P. Kingma, and Max Welling. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. *arXiv:1410.6460 [stat]*, October 2014. arXiv: 1410.6460.