Online MCMC based Bayesian inference

Ankit Bhardwaj 150101
Bhavy Khatri 150186
Rohit Gupta 150593

1 Abstract

MCMC is a sampling based approximate inference method (a popular example is Gibbs sampling) for estimating intractable posterior distributions. One important problem with MCMC algorithms has always been scalability. There has been a lot of recent interest in making these MCMC methods faster by designing online versions of MCMC algorithm. In this project, we have surveyed the method of Stochastically optimizing the gradients to obtain online versions of MCMC algorithms. The problem of online MCMC can be summarised as, Can we approximately sample from a Bayesian posterior distribution if we are only allowed to touch a small mini-batch of data-items for every sample we generate?[3] We look at many other research papers on improving the basic SGLD an online MCMC algorithm and other online methods.

2 Introduction

Markov chain Monte Carlo methods create samples from a possibly multi-dimensional continuous random variable, with probability density proportional to a known function. These samples can be used to evaluate an integral over that variable, as its expected value or variance.[1]. There are many MCMC algorithms like Metropolis-Hastings (MH) Sampling which assume a proposal distribution to draw samples and accept with some probablity. Another is Gibbs Sampling which is a special case of MH Sampling in which proposal is the conditional distribution samples from these conditionals in a cyclic order with acceptance probability 1.

2.1 Langevin Dynamics

In Langevin Dynamics Constructs proposal distribution using gradient of the log-posterior as follows

$$\theta^* = \theta^{(t-1)} + \frac{\eta}{2}\nabla_\theta[\log p(D|\theta) + \log p(\theta)]|_{\theta^{(t-1)}}$$
$$\theta^{(t)} \sim N(\theta^*, \eta)$$

Then accept/reject using an MH step to accept $\theta^{(t)}$ or resample it. This algorithm is efficient computationally and almost as fast as standard gradient ascent/descent based MAP estimation.

2.2 Stochastic Gradient (Online) Langevin Dynamics

It is a extension of Langevin Dynamics where we process data in small mini-batches. SGLD does stochastic gradient descent + MH. Given minibatch $D_t = \{x_{t1}, ..., x_{tN_1}\}$ we have stochastic updates as [10]

$$\theta^* = \theta^{(t-1)} + \eta_t\nabla_\theta[\frac{N}{|D_t|}\sum_{n=1}^{N_t}\log p(x_{tn}|\theta) + \log p(\theta)]|_{\theta^{(t-1)}}$$
$$\theta^{(t)} \sim N(\theta^*, \sigma^2)$$

Basic SGLD e.g. Exhibits slow convergence and mixing. Uses same learning rate η_t in all dimensions of θ. Doesnt apply to models where θ is constrained. Also it assumes that the model is differentiable. A lot of recent work on improving the basic SGLD to handle such limitations.

3 Approach

3.1 Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex [9]

One of the problems that we find with SGLD approach is that it doesnt apply to models where the parameter is constrained (e.g., non-neg or prob. vector). The authors of the above paper - Sam Patterson and Yee Whye Teh have tackled this problem. They have investigated Langevin MCMC methods on probability simplex.
The probability simplex is the span of probability vectors that represent discrete probability distributions over K items. One important example would br LDA.

$$\Delta_K = \{\pi_1, ..., \pi_K : \pi_k \geq 0, \sum_{k=0}^{K} \pi_k = 1\}$$

On such models, the methods include VI and MCMC methods but not online MCMC methods. The three problems that SGLD faces when used in this context are:

- Constraint in simplex (constrained system).
- Distribution lying mostly along the boundaries of the simplex leading to unstable gradients in many parametrizations.
- High dimensionality of simplices.

Langevin Dynamics has isotropic proposal distribution due to which there is slow mixing. It has been seen that preconditioning Θ is a way to solve this problem. This has been done in a different research paper - 'Riemann manifold Metropolis adjusted Langevin algorithm' [7], which is called as Reimannian Langevin Dynamics(RLD). Here in this report, we are not giving details of the method. Briefly, RLD consists of a Gaussian proposal along with a MH correction step. We am not giving the exact form of equations here.
It should be easy to understand that in the equation of updates of the parameter, we can replace the gradient of posterior (which appears exactly as in LD) by the SGLD approximation hence giving us the stochastic version of RLD. We call that SGRLD. In the paper, the authors have worked out the details of different parametrizations of the simplex and how the use of SGRLD affects the parameters of these models. We are summarizing the results of presented below:
Reduced-Mean: In the simplex, we replace the final variable π_K by $1 - \sum_{k=0}^{K-1} \pi_k$. This gives us a K-dimensional vector lying on K-1 dimensional plane forming a simplex. But the bound of $0 \leq \pi_k \leq 1$ may be violated.
Expanded Mean: In this method, we set prior on the parameters - Dirichlet prior making it so that $\pi_i = \frac{\theta_i}{\sum_{i=1}^{K} \theta_i}$. The boundary condition of $\theta_i > 0$ is satisfied by taking absolute values. This case bypasses the boundary conditions altogether. The prior on θ is gamma prior.
Reduced-Natural: $\pi_k = \frac{exp(\theta_k)}{1+\sum_{k=0}^{K-1} exp(\theta_k)}$ where we take no boundary constraints on θ.
Expanded-Natural: Take the same form as above except for the sum going to all dimensions and we remove 1 from denominator.
Through the different experiments the authors have concluded that expanded-mean sampler is more efficient and accurate. Thus, they used that sampler to sample from the simplex proposal distribution. The authors have gone on to provide the application of SGRLD on LDA and show the results of several different experiments. They have been successful in applying Langevin Monte Carlo techniques to a constrained parameter space such as the probability simplex.

3.2 Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks [5]

The two major problems in training of deep neural networks are - the arbitrary curvature of objective function which is addressed by using adaptive preconditioning for Stochastic Gradient Descent and the overfitting problem which is addressed by early stopping and Bayesian model averaging. For Bayesian model averaging, we need to estimate posteriors which is where SGLD method comes in. The problem in application of this method is that the rapidly changing curvature renders default SGLD methods inefficient. For this purpose, the authors have given the method of adaptive preconditioning on SGLD.

The contemporary methods that are used for estimating posteriors of neural networks are Bayes by Backprop - SVI method, probabilistic backpropagation - an online expectation propogation method. These methods assume the posterior is comprised of separable Gaussian distributions. This can lead to unreasonable approximation errors and underestimation of model uncertainty.
The authors have proposed the novel method - preconditioned SGLD or pSGLD which can, with adding trivial computational overhead, improve efficiency. Several theoretical and experimental justifications have been provided in the paper.
In this method, we employ a user-chosen preconditioning matrix in the learning rate. This allows different amounts of updates in different dimensions. Preconditioning aims to constitute a local transform such that the rate of curvature is equal in all directions. The authors have used the same preconditioner as in RMSprop. The update equation for preconditioner is dependent on the current gradient value. We are providing the entire algorithm below:

1. Inputs $\{\epsilon_t\}_{t=1:T}, \lambda, \alpha$
2. Outputs $\{\theta_t\}_{t=1:T}$
3. Initialize $V_0 = 0$, random θ_1
4. for t = 1 to T:
   (a) Sample minibatch gradient $D_n^t = \{d_{t_i}\}_{i=1:n}$
   (b) Estimate gradient $\bar{g}(\theta_t, X^t) = \frac{1}{n}\sum_{i=1}^{n} \nabla log(p(d_{t_i}|\theta_t))$
   (c) $V(\theta_t) = \alpha V(\theta_{t-1}) + (1-\alpha)\bar{g}(\theta_t, D^t) \odot \bar{g}(\theta_t, D^t)$
   (d) $G(\theta_t) = \text{diag}(1 \oslash (\lambda 1 + \sqrt{V(\theta_t)}))$
   (e) $\theta_{t+1} = \theta_t + \frac{\epsilon_t}{2}[G(\theta_t)(\nabla_\theta logp(\theta_t) + N\bar{g}(\theta_t, D^t)) + \Gamma(\theta_t)] + N(0, \epsilon_t G(\theta_t))$

The steps c and d are the update equations for the preconditioner. The circular symbols mean element wise multiplication and division respectievely. The authors have utilized various methods to prove the performance of the above reported algorithm which we shall avoid going into details about as it is beyond the scope of this report. The author has shown empirical results for Logistic Regression, Feedforward Neural Nets, and Convolutional Neural Nets, demonstrating that preconditioned SGLD method gives state-of-the-art performance on these models.

3.3 Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring[2]

We will first look at some of the properties of SGLD algorithm. Detailed discussion about SGLD was done in previous section but some of the facts need to be mentioned to motivate ourself towards using SGFS. SGLD algorithm as an online MCMC method suffers from the problem of low mixing rate. Note that the mixing time refers to the time for a Markov Chain to reach its steady state distribution. In MH, after computing the updates for the parameter we need to compute the acceptance probability to decide whether to retain the sample or not. This step is expensive in general as we need to compute the probabilities over whole dataset [10]. As the step size goes to zero, the discretization error in the Langevin equation disappears and we do not need to conduct expensive Metropolis- Hasting(MH) accept/reject tests that use the whole dataset [2]. Although, decreasing step size lead to low computation but it also lead to low mixing rate. This is due to because it will not be able to search in the support set/space of parameter.
SGFS overcome the low mixing rate problem of SGLD by dividing the update into two cases based on step size. If the step size is large then sample from approximate normal distribution, if step size is small then sample from accurate approximation of posterior(just like langevian dynamics case). The normal distribution here refers to the approximation of posterior distribution using bayesian central limit theorem. BCLT states that when the data size is very large and likelihood of each data point is conditionally independent(note that likelihood is conditioned on parameter). Then posterior distribution can be approximated by normal distribution.
SGFS algorithm can also be seen as a trade off between sampling accuracy and mixing rate. Sampling accuracy determines similarity between distribution of generated samples and true distribution. In case of high step size mixing rate is high but the sampling accuracy is compromised as we are just generating from the approximate normal distribution. In case of small step size we are generating from the actual non gaussian distribution which increases the sampling accuracy but this come at the expense of lower mixing rate. This can be summarised in the following table:

Step Size	Sampling Accuracy	Mixing Rate
High	Low	High
Low	High	Low

Table 1: Trade off in SGFS Algorithm

In short, SGFS provide the freedom to manage the right trade off between sampling accuracy and mixing rate depending on the problem at hand [2]. Note, this feature was certainly missing in SGLD.

3.4 Stochastic Gradient Hamiltonian Monte Carlo

HMC: In hamiltonian monte carlo we define posterior distribution in terms of potential energy. Here parameter is seen as the position and potential energy is function of position. We also define a hamiltonian function as a sum of potential energy and kinetic energy, where kinetic energy is a function of momentum - an auxiliary variable. Now in HMC we generate both position and momentum samples from joint distribution which is proportional to negative exponential of hamiltonian function. Hamiltonian dynamics defines how both position and momentum change w.r.t time. We discretize hamiltonian dynamics to write the update equation for both the variables. After that MH accept/reject test is done on the updated variables.
However, we need to compute the gradient of the potential energy function in order to simulate from hamiltonian dynamical system. This turn out to be one of the major limitation of HMC in case when there is a large dataset [4]. In the paper they tried two different approaches:

- **Naive Stochastic Gradient HMC:** Instead of computing the gradient on full dataset, we compute it using only minibatch of it. However, the natural implementation of the stochastic approximation can be arbitrarily bad. To see why it is the case, let's first take a look at the noisy gradient update equation:

  $$\nabla \hat{U}(\theta) = \nabla U(\theta) + N(0, V(\theta))$$

  Here, V is the covariance of the stochastic gradient noise, which can depend on the current model parameters and sample size. When we replace gradient of U with stochastic gradient of U, this introduces noise in momentum update. In this naive approach entropy of joint distribution of (θ, r) increases with time. This hints at the fact that the joint distribution tends toward a uniform distribution, which can be very far from the target distribution .

- **Stochastic gradient HMC with friction:** To minimize the effect of the injected noise on the dynamics, we add a friction term to the momentum update:

  $$d\theta = M^{-1} r dt$$
  $$dr = -\nabla U(\theta) - BM^{-1} r dt + N(0, 2Bdt)$$

  Here, B is the diffusion matrix contributed by gaussian noise. The friction term helps decrease the total energy of the system, thus reducing the influence of noise. This type of dynamical system is commonly referred to as second-order Langevin dynamics in physics. Importantly, we note that the Langevin dynamics used in SGLD are first-order, which can be viewed as a lim- iting case of our second-order dynamics when the friction term is large. In particular, the dynamics of SGLD can be viewed as second-order Langevian dynamics with a large friction term.

3.5 Bayesian Sampling Using Stochastic Gradient Thermostats[8]

One important problem in all stochastic gradient MCMC methods is that the introduction of noise prevents proper sampling after discretization incase step size is large or noise is unknown. The main idea of the method is to add additional variables to stabilize momentum fluctuations due to noise. The paper compares all SG-MCMC methods to approximation of canonical ensembles through dynamics. This is a concept from statistical physics. Existing methods neglect the condition that system temperature should be near target temperature. This causes unconstrained noise to lead to incorrect sampling. Using theories from stochastic physics, we add additional variables that play the

conventional role of thermostat.
The canonical form of the posterior would be as:

$$p(\theta|X) = \frac{1}{Z}\text{exp}(-U(\theta))$$
$$U(\theta) = -\text{log}p(X|\theta) - \text{log}p(\theta)$$

Now, we are going to provide the statistical physics justification and the algorithm for the thermostat. The mathematical justification is very detailed and beyond our scope.
The probability of the states in a canonical ensemble follows the canonical distribution:

$$\rho(\theta, p) \propto \text{exp}(-H(\theta, p)/k_B T)$$

The following equation is valid:

$$k_B T = \frac{1}{n} E[p^T p]$$

This has been derived from the thermal equilibrium condition from statistical physics. According to the authors, all Bayesian dynamics-based sampling methods are canonical ensembles with $k_B T = 1$. Then:

$$\rho(\theta, p) \propto \text{exp}(-H(\theta, p))$$
$$\rho_\theta(\theta) \propto \text{exp}(-U(\theta))$$
$$\rho_p(p) \propto \text{exp}(-K(p))$$

In ordinary LD, this is held valid but in case of SGLD, with the addition of stochastic force, the condition above is thrown off. Therefore, to generate correct samples, one needs to introduce a proper thermostat, which adaptively controls the mean kinetic energy. The additional variable ξ is introduced for this:

$$d\theta = pdt$$
$$dp = \tilde{f}(\theta)dt - \xi pdt + \sqrt{2A}N(0, dt)$$
$$d\xi = (\frac{1}{n}p^T p - 1)dt$$

All these together form the following algorithm termed in the paper as Stochastic Gradient Nose-Hoover Thermostat (SGNHT).

1. Input: Parameters h, A.
2. Initialize: $\theta_{(0)} \in R^n, p_{(0)} \sim N(0, I), \xi_{(0)} = A$
3. for t = 1 to T, do:
   (a) Evaluate $\nabla U(\theta_{(t-1)})$ through SGLD
   (b) $p_{(t)} = p_{(t-1)} - \xi_{(t-1)} p_{(t-1)} h - \nabla U(\theta_{(t-1)})h + \sqrt{2A}N(0, h)$
   (c) $\theta_{(t)} = \theta_{(t-1)} + p_{(t)} h$
   (d) $\xi_{(t)} = \xi_{(t-1)} + (\frac{1}{n} p_{(t)}^T p_{(t)} - 1)h$

As mentioned before, we don't go into the mathematical justification of the algorithm given above. The authors have also provided many experimental results for the algorithm. This algorithm allows for the use of larger discretization step, smaller diffusion factor, or smaller minibatch to improve the sampling efficiency without sacrificing accuracy.

3.6 Variance Reduction in Stochastic Gradient Langevin Dynamics[6]

Authros propose a new Langevin algorithm designed to reduce variance in the stochastic gradient, with minimal additional computational overhead and also provide a memory efficient variant of the algorithm. They find that conversion to the true posterior under reasonable assumptions provide better rate of convergence than basic SGLD. They test it for variety of machine learning tasks like regression, classification, independent component analysis and mixture modeling. They give two alorithms where we use approximate gradient for data points other than minibatch set

- **SAGA-LD**
  - Store the gradient information about each point. If the data point is not in minibatch approximate it using stored gradient.
  - explicitly store N approximate gradients $\{g_{\alpha i}\}_{i=1}^{N}$ for all N points
  - As we iterate through the data, if a data point is not selected in the current minibatch, we approximate its gradient with $g_{\alpha i}$ else find gradient find $\nabla \log p(x_i|\theta_t)$, update $g_{\alpha i}$ thus gradient becomes

$$\sum_{i=1}^{N} \nabla \log p(x_i|\theta_t) \approx \frac{N}{N_t} \sum_{x_i \in B_t} (\nabla \log p(x_i|\theta_t) - g_{\alpha i}) + \sum_{i=1}^{N} g_{\alpha i}$$

  - Low computational cost, but high memory overhead $O(Nd)$ compared to SGLD
- **SVRG-LD**
  - Instead of storing gradient for every data point, store the total approximate gradient and update it after every m iterations.
  - After every m iterations (say t) update $\hat{g} = \sum_{i=1}^{N} \nabla \log(x_i|\theta_t)$ and use it to sample for next m points.
  - Significant improvement in memory overhead compared to SAGA-LD but with increased computational cost as full gradient is calculated for the full dataset after miterations.

4 Conclusion

In this project we tried to survey the recent work on improving the basic SGLD to handle limitations. Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex [9] address the problem that basic SGLD doesnt apply to models where θ is constrained. It can handle the case of constrained variables. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks [5] uses a preconditioner matrix in the learning rate to improve convergence addressing the problem of slow convergence and mixing. Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring [2] also address the issue of slow mixing. Stochastic Gradient Hamiltonian Monte Carlo [4] improves upon the natural implementation of the stochastic approximation and nosiey gradients by using second-order Langevin dynamics with a friction term. Bayesian Sampling Using Stochastic Gradient Thermostats[8] algorithm allows for the use of larger discretization step, smaller diffusion factor, or smaller minibatch to improve the sampling efficiency without sacrificing accuracy addressing the problem of noisy gradients. Variance Reduction in Stochastic Gradient Langevin Dynamics[6] present techniques for reducing variance in stochastic gradient Langevin dynamics where the high variance is inherent in these noisy gradients degrades performance and leads to slower mixing. Still there is a lot of work to be done for improvements in removing the limitations of basic SGLD but it sure has come a long way. We learned various key insights on improving the stochastic gradient methods and how SGLD is not limited to one field but has influence in statistics, physics etc. We learned many a times solution for a problem has already been addressed in some other field and mere extension to given problem may give remarkable results.

Acknowledgments

This work was done as a part of coursework for course CS698X: Topics in Probabilistic Modeling and Inference (Spring 2019) under Prof. Piyush Rai. We thank him for guiding us and encouraging us in studying the recent advancements in this growing field.

References

[1] Markov Chain Monte Carlo. `https://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo`.

[2] Sungjin Ahn, Anoop Korattikara Balan, and Max Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. In *ICML*, 2012.

[3] Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. *arXiv preprint arXiv:1206.6380*, 2012.

[4] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic Gradient Hamiltonian Monte Carlo. *arXiv e-prints*, page arXiv:1402.4102, Feb 2014.

[5] David Carlson Chunyuan Li, Changyou Chen and Lawrence Carin. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks.

[6] Kumar Avinava Dubey, Sashank J Reddi, Sinead A Williamson, Barnabas Poczos, Alexander J Smola, and Eric P Xing. Variance reduction in stochastic gradient langevin dynamics. In *Advances in neural information processing systems*, pages 1154–1162, 2016.

[7] M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society B 73 (2011), 137.*

[8] Ryan Babbush Changyou Chen Robert D. Skeel Nan Ding, Youhan Fang and Hartmut Neven. Bayesian Sampling Using Stochastic Gradient Thermostats.

[9] Sam Patterson and Yee Whye Teh. Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex.

[10] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 681–688, USA, 2011. Omnipress.