Stochastic techniques for MCMC methods

Chirag Gupta, Aditya Modi, Ayush Mittal
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur
chiragvg,admodi,ayushmi@iitk.ac.in

Advisor
Professor Piyush Rai

Project proposal [1]

Approximate Bayesian inference seeks to perform inference over the complete posterior when it is not available in an analytic form (for example, if the prior is non-conjugate). Markov Chain Monte Carlo (MCMC) methods and variational methods are the two primary techniques used for approximate Bayesian inference. MCMC methods (refer [2]) yield a numerical approximation of the inference integral over the posterior by drawing samples that are asymptotically distributed as the posterior of interest. Recent advances in large-scale learning have made stochastic methods prevalent in learning models. However, the progress of stochastic methods in Bayesian learning is still in the nascent phase. Traditionally, MCMC methods or variational inference methods require computation over the complete dataset. Therefore, we mainly intend to look at the recent advances in using stochastic methods for Bayesian inference.

MCMC methods - Improving the mixing time

Let θ be a parameter vector, $p(\theta)$ the prior distribution over θ, and $p(x|\theta)$ the likelihood term. Given N data-points, $x_1, x_2, \dots x_N \in \mathbf{R}^d$, the posterior looks like $p(\theta|X) \propto p(\theta) \prod_{i=1}^{N} p(x_i|\theta)$. In fully Bayesian learning, the aim is not only to give a point estimate as in MAP, but also to learn the distribution over the parameter space θ, i.e. $p(\theta|X)$. MCMC methods provide a way to estimate the expectation over the posterior or sample from the posterior to assess the uncertainty in prediction.
Popular MCMC algorithms like Gibbs sampling, Metropolis-Hastings algorithm suffer from slow convergence rate to the posterior distribution. This is primarily due to the random walk behavior that these methods exhibit. To make the posterior samples explore all modes of the distribution, referred to as the chain mixing well, hybrid MCMC (HMC) [10], adapts physical system dynamics to propose future states in the sampling process. For understanding HMC, consider a constant energy physical system with $U(x)$ and $K(p)$ as the two modes of decomposition. Considering the constant energy model, we can simulate the state of the object using Hamiltonian dynamics. Using a method called leap-frog method, we approximate future states (x, p) of a system, by discretizing time. The method updates the position and momentum variables sequentially $(p_i(t + \delta/2) \rightarrow x_i(t + \delta) \rightarrow p_i(t + \delta))$. By adding auxiliary variables $u \sim \mathcal{N}(0, I_d)$, we form a constant energy system with $E(\theta') = H(\theta, u) = K(u) + U(\theta)$. The canonical distribution obtained from statistical mechanics literature $p(\theta, u) \propto exp(-K(u))exp(-U(\theta))$, can be shown to factorize over θ and u. Mostly the energy function taken for u leads to a Gaussian distribution for the auxiliary variables. Simulating the Hamiltonian dynamics on this system, would help us in obtaining MCMC samples efficiently. This requires calculating the value $-\frac{\partial \log p(\theta)}{\partial \theta_i}$. We use the Hamiltonian dynamics as our proposal distribution in MH algorithm and to explore every θ, we sample a random u from the distribution $K(u) = \mathcal{N}(0, I_d)$ at each step. In the Hamiltonian dynamics step, we take L leapfrog steps in u

[1] This is the mid-term report for the course CS772: Probabilistic Machine Learning

and θ. We discard the auxiliary variables u and take samples of θ. If $L = 1$, we get what we call, Langevin dynamics. The algorithm is given below in 1. The parameter ρ requires careful tuning to obtain samples from the posterior.

Data: N data points, ρ
Initialize θ_0;
while *more samples needed* **do**
  Sample $v \sim \mathcal{U}(0,1)$ and $u \sim \mathcal{N}(0, I_d)$;
  $\Delta(\theta) = -\frac{\partial \log p(\theta|X)}{\partial \theta}$;
  $\theta_n = \theta_{i-1} + \rho(u + \rho\Delta(\theta_{i-1})/2)$;
  if $v < \mathcal{A} = \min\{1, \frac{p(\theta_n)}{p(\theta_{i-1})} exp(-\frac{1}{2}(u'u))\}$ **then**
    $(\theta_i, u_i) = (\theta_n, u)$;
  end
  else
    $(\theta_i, u_i) = (\theta_{i-1}, u_{i-1})$;
  end
end

Algorithm 1: MCMC with Langevin Dynamics

Making MCMC stochastic

As we have seen in the algorithm for Langevin dynamics, we need to compute the gradient and update the state θ which leads to samples from posterior. On the other hand, for MAP estimation, we require to give a point estimate which maximizes the negative log posterior. MAP inference over the posterior is essentially an optimization problem of maximizing the log posterior. Consider a natural gradient ascent step with step-size ϵ_t -

$$\theta_{t+1} = \theta_t + \frac{\epsilon_t}{2}\left(\nabla \log p(\theta_t) + \sum_{i=1}^{N} \log p(x_i|\theta_t)\right) \tag{1}$$

The stochastic gradient ascent (refer [3]) step works over mini-batches of size $n << N$. The idea is to see only a part of the data at every step, and approximate the true gradient by treating it as a stochastic quantity.

$$\theta_{t+1} = \theta_t + \frac{\epsilon_t}{2}\left(\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \log p(x_{ti}|\theta_t)\right) \tag{2}$$

The data-points may truly be coming in an online fashion and we may be unable to use previous data due to limited storage space. However, it is possible that all the data (N samples) is available in hard disk, but only a part of it (n samples) can be loaded into RAM.

Notice that the update in Langevin dynamics looks strikingly similar to the gradient update.

$$\theta_{t+1} = \theta_t + \frac{\epsilon}{2}\left(\nabla \log p(\theta_t) + \sum_{i=1}^{N} \log p(x_i|\theta_t)\right) + \eta_t \tag{3}$$

Here, $\eta_t \sim N(0, \epsilon)$ is noise that is injected, so that the sequence does not simply converge to the MAP. Essentially, the noise encourages the chain to explore new (possibly low probability) areas of the probability distribution while the gradient step encourages the chain to move towards high probability areas, and draw more samples from there.

The similarity to gradient descent suggests a simple stochastic Langevin dynamics update -

$$\theta_{t+1} = \theta_t + \frac{\epsilon}{2}\left(\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \log p(x_{ti}|\theta_t)\right) + \eta_t \tag{4}$$

This technique was proposed in [11], along with some analysis and experimental validation. They show that, if the parameters ϵ_t are chosen such that they decrease towards zero such that they satisfy

conditions for convergence in gradient descent, it would explore the complete posterior support. Also, as ϵ_t decrease, the acceptance probability of the MH-step increases and after few steps, we can simply ignore the step in the algorithm. For showing that the samples indeed follow the posterior distribution, they argue that the variance due to stochasticity in gradients will be dominated by the Gaussian noise which we have added at each step. Therefore, the total stochastic gradient step is approximately the exact gradient step if Lipschitz assumption is made over the gradients. Therefore, the obtained sequence would approach a sequence generated by Langevin dynamics and we would obtain samples from the posterior distribution.

Experiments

We did some experiments to reproduce the results as shown in the work by Welling et al. We see that the initial phase in the algorithm looks similar to a gradient descent algorithm but with the decreasing step size, we get the samples very close to Langevin dynamics. A very important aspect of the algorithm is tuning the parameter ϵ_t at each t. The Langevin dynamics method in itself is highly dependent on proper tuning of the step sizes. Along with this, we now have to maintain a decreasing trend in the values. If the noise term is very large as compared to the stochastic gradient, we would just follow a random walk behavior rather than posterior sampling. This is because we are neglecting the MH accept-reject step. Also, if the parameter doesn't decrease properly, we would not be able to explore all modes and only one local optimum would be found.

We firstly reproduced the results for the experiment using a synthetic dataset. Considering a multi-modal posterior distribution the mixture of Gaussians was used:

$$\theta_1 \sim N(0, \sigma_1^2); \qquad \theta_2 \sim N(0, \sigma_2^2)$$
$$x_i \sim \frac{1}{2} N(\theta_1, \sigma_x^2) + \frac{1}{2} N(\theta_1 + \theta_2, \sigma_x^2)$$

where $\sigma_1^2 = 10$, $\sigma_2^2 = 1$ and $\sigma_x^2 = 2$. Drawing 300 data points from the distribution with $\theta_1 = 0$ and $\theta_2 = 1$ gave a posterior distribution with two modes. Running the stochastic gradient Langevin algorithm with 10000 sweeps over the dataset with given step sizes resulted in the following outputs - figure 1,2 and 3.

As in the original paper, we also apply the algorithm to a Bayesian logistic regression problem. Using the a9a adult dataset consisting of 32561 observations with 123 features, we created a train-test split of 80-20%. Using batch sizes of 10 and over multiple runs, we see that the accuracy increases quite rapidly (see figure 4). The pattern of increase in joint probability also shows how the SGLD algorithm progresses.

We see that proper tuning of the hyperparameters is very important for the algorithm to work. We aim to get a better understanding of this dependence and explore further improvements for the same.

Relationship with optimization and future work

As we have stated, the SGLD algorithm establishes a strong parallel between stochastic optimization and MCMC posterior sampling. Recent work [4], [5] also suggests that if we use the idea of Langevin dynamics for non-convex optimization, we can get better local minimas which closely approximate the global minima. The intuition behind the claim is the ability of the Bayesian analogue to explore the complete parameter space. We can therefore, also consider the applicability of theses methods for achieving better results of a wide category of optimization problems. Further, we also aim to study the extension of optimization with momentum variables in the same way which closely resembles stochastic gradient Hamiltonian MCMC methods (refer [6]). The relationship of Bayesian and optimization techniques is not new as the idea was presented way back when Gibbs sampling and simulated annealing were studied.

Another idea that we may explore is using variance reduction methods in estimating the gradient at each step. The Langevin dynamics method requires the computation of exact gradient and SGLD requires the variance in stochastic gradient to reduce at a sufficiently fast rate. In the same spirit, if we use a method for computing gradients for the log posterior which inherently reduces the variance, then the high sensitivity towards the hyperparameter can be reduced. We may also look to make modifications to the noise term, in spirit with [7] and [1]. We also aim to look deeper into the unified framework presented in [9].

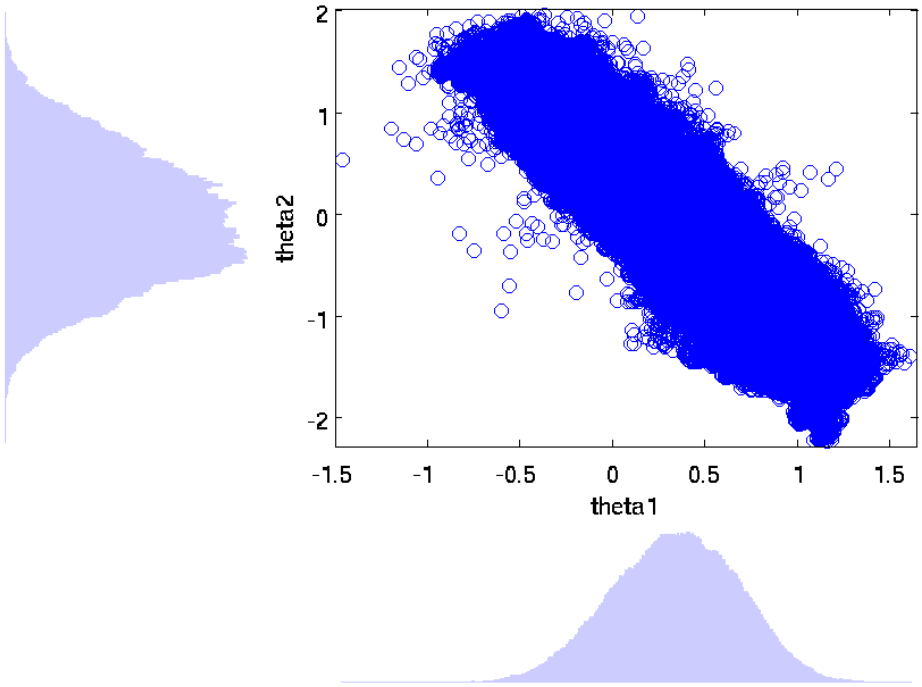

Figure 1: Random walk behavior for improper step size

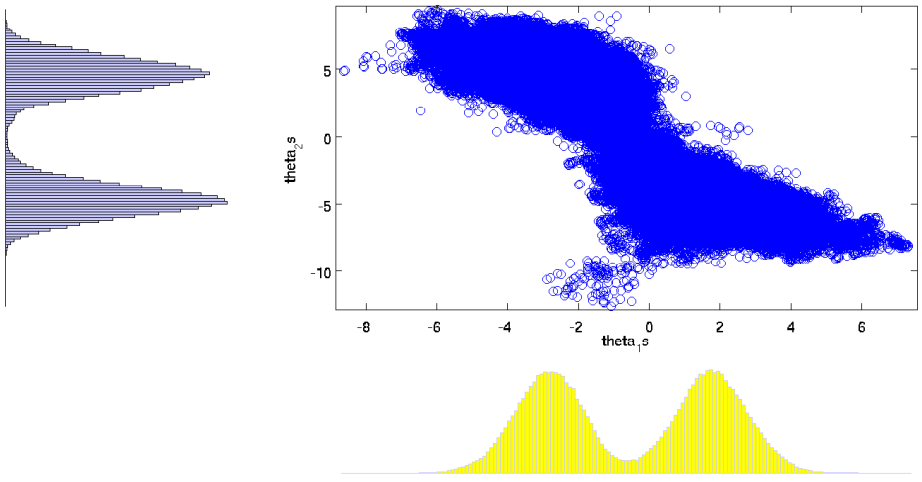

Figure 2: Proper step size is able to get samples from the multimodal posterior

[8])

References

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

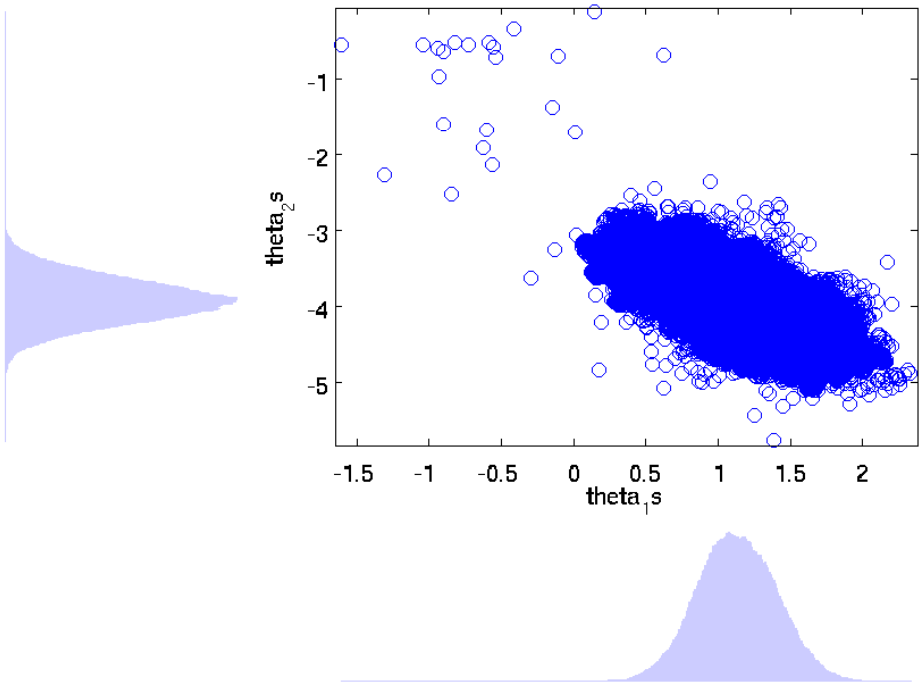

Figure 3: SGLD is unable to escape one mode for improper step size

[2] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[4] Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li, and Lawrence Carin. Bridging the gap between stochastic gradient mcmc and stochastic optimization. *arXiv preprint arXiv:1512.07962*, 2015.

[5] Changyou Chen, Nan Ding, and Lawrence Carin. On the convergence of stochastic gradient mcmc algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pages 2269–2277, 2015.

[6] Tianqi Chen, Emily B Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. *arXiv preprint arXiv:1402.4102*, 2014.

[7] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems*, pages 3203–3211, 2014.

[8] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[9] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems*, pages 2899–2907, 2015.

[10] Radford M Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011.

[11] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

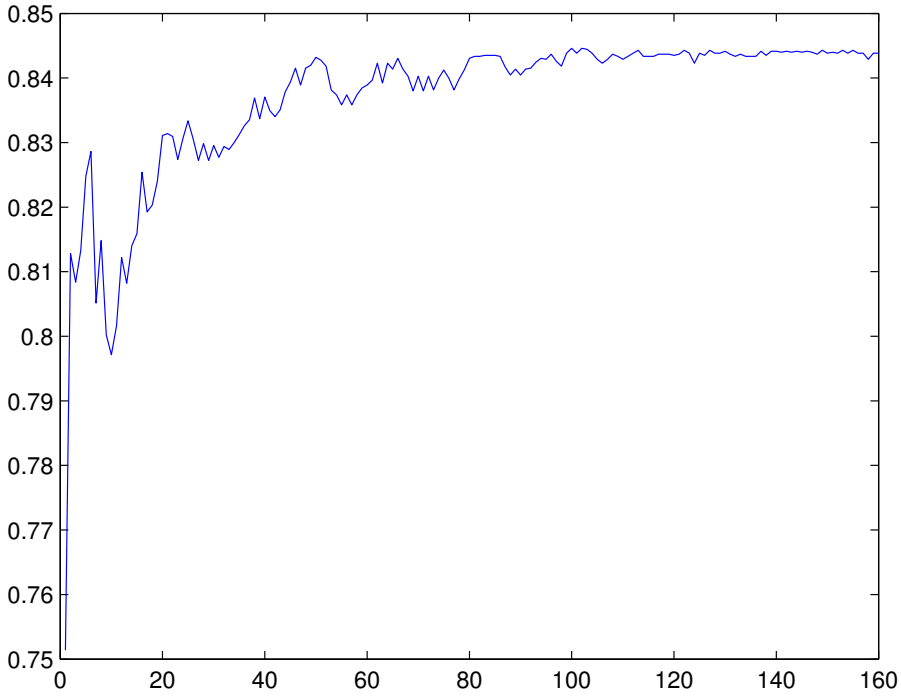

Figure 4: Accuracy over a9a dataset for logistic regression using SGLD