Bayesian Learning via Stochastic Gradient Langevin Dynamics

Students: Zakaria Boulkhir & Omar Iken

January 26, 2022

1 Introduction

Bayesian learning has been a great perspective in the understanding and resolving of many problems presented by machine learning algorithms; such as their uncertainty. However, this framework remains very vague and open for research and are particularly not well suited for modern problems that are based on large-scale datasets. In the meanwhile, the discovery of neural networks has arisen in the field. It has been, since it was discovered, the most ever used type of models especially in industrial applications. This sudden big change of interest comes also from the fact that these models are highly adjustable and are opening different new types of learning and data generation. Most importantly, the framework presented in the Bayesian setting does not fit into that of the neural networks. Besides the need for large datasets, The most used optimizers in Neural Networks are a blocking barrier towards an implementation of a Bayesian approach.

In this paper, the authors represented using other literature a frame work that proved empirically fitting to use Bayesian inference within the Stochastic Gradient Decent Optimization setting in a neural network. However, this might need more theoretical improvements and more generalizations towards other sampling interests and other optimizers (Which is provided by more recent literature).

2 Approach Overview

The approach proposed by the authors aims at learning from large-scale datasets based on iterative learning from small mini-batches. This approach relies on the combination of two main existing approaches: 1) stochastic gradient algorithms and 2) Langevin dynamics. The combination of these two concepts makes it possible to deal with large-scale datasets using Bayesian inference, since, as is well known, Bayesian MCMC methods require access to the entire dataset, which is time consuming. Therefore, the main idea of the proposed approach is to add the right amount of noise to a standard stochastic gradient optimization algorithm which is the main idea of Langevin dynamics. The authors have shown that the iterations converge to samples of the true posterior distribution. The main objective is to perform Bayesian inference from large-scale datasets by combining:

Stochastic Gradient The main idea is that instead of computing the gradient on the whole data set, we compute the gradient on a subset and try to approximate the true gradient over the iterations. The parameters start at a random point and adjusted iteratively till convergence. The noise in the gradient caused by the use of these subsets is resolved over this iterations. This process saves both time and memory calculations. The parameters updates at a given iteration t are given by:

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

Where ϵ_t is a sequence of step sizes. $p(x|\theta)$ is the probability of a data sample x given the parameter θ (likelihood), and $p(\theta)$ is the prior distribution.

Langevin Dynamics Langevin dynamics is motivated and initially derived as a discretization of a stochastic differential equation whose equilibrium distribution is the posterior distribution. Langevin dynamics injects noise into the parameter updates in such a way that the trajectory of the parameters will converge to the full posterior distribution rather than just the maximum a posteriori mode. The parameters updates given by Langevin dynamics are:

$$\Delta\theta_t = \frac{\epsilon}{2}\left(\nabla \log p(\theta_t) + \sum_{i=1}^{N} \nabla \log p(x_i|\theta_t)\right) + \eta_t, \qquad \eta_t \sim \mathcal{N}(0,\epsilon) \tag{2}$$

Stochastic Gradient (+) Langevin Dynamics Combining the two equations given above (1 and 2) allows efficient use of large-scale datasets while capturing the parameters in a Bayesian way. The main steps are: 1) change the gradient of the Langevin dynamics to mini-batch estimation. 2) inject noise into the parameter updates such that the parameter trajectory converges to the full posterior distribution. 3) the step size must go to zero. Notice that the noise variance is balanced with gradient step sizes. Thus the parameters update is given by:

$$\Delta\theta_t = \frac{\epsilon_t}{2}\left(\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla \log p(x_{ti}|\theta_t)\right) + \eta_t, \qquad \eta_t \sim \mathcal{N}(0,\epsilon) \tag{3}$$

The proposed method uses updating rules similarly to stochastic gradient descent, but in addition to that, injects noise so that the resulting sample set approaches the posterior distribution rather than a point estimate. The resulting algorithm starts off being similar to stochastic optimization, then automatically transitions to an algorithm that simulates samples of the posterior using Langevin dynamics.

In order for this method to work, i.e. to ensure that the method will converge to a local maximum, two main conditions are required, and they both concern the step size:

$$\sum_{t=1}^{\infty} \epsilon_t = \infty \qquad\qquad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty \tag{4}$$

The first condition ensures that the parameters will reach the high probability regions whatever the starting point is. While the second one ensures that the parameters will converge to the mode and not just bouncing around it. A typical choice would be of the form $\epsilon_t = a(b+t)^{-\gamma}$ where $a, b \in \mathbb{R}$ and $\gamma \in (0.5, 1]$.

Posterior Sampling The idea is that the samples collection should start after the algorithm has entered its posterior sampling phase, which will not happen until after it becomes Langevin dynamics. The main condition to ensure that the algorithm is in its posterior sampling phase is given by:

$$\alpha = \frac{\epsilon_t N^2}{4n}\lambda_{max}(M^{1/2}V_s M^{1/2}) \ll 1 \tag{5}$$

Where λ_{max} is the largest eigenvalue, M is the preconditioning matrix and V is empirical variance of the parameters.

3 Implementation

The implementation relies on three different important aspects:

- Defining a prior for our parameters (θ within the paper).
- Finding a relevant likelihood distribution depending on the priors chosen before ($p(x_i|\theta)$)
- Finally, computing the update ($\nabla\theta$) of the SGD and adding the noise (η_t).

The intuitive explanation to why this works is that it lets the SGD optimization term goes towards the mode, but it will never be achieved due to the second noisy term. Moreover, the conditions learning rate prevents the parameters from leaving that mode, resulting in a random walk behavior.

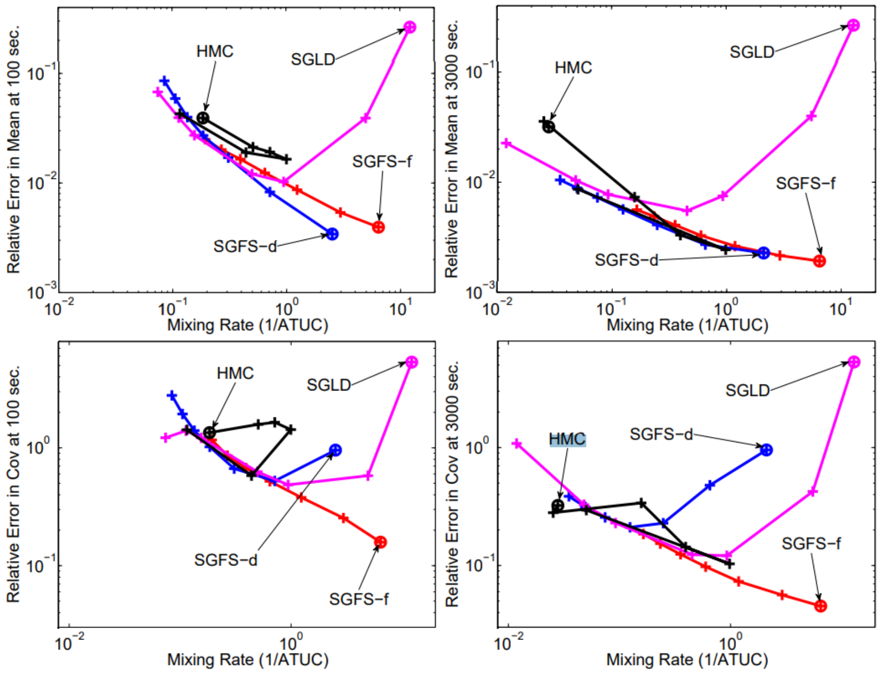

Figure 1: A demonstration of the performances of SGLD in comparison with other MC methods. A graph from [2]

4 Experiments

To demostrate the effectiveness of this method, we rely on results published in a different paper. As we can clearly see from experiments within a different paper in 1, the algorithm SGLD which was first suggested by this paper is outperforming most of the other MCMC methods. The convergence was quick and correct with a very high probability. We can further notice that HMC has failed in all of the experiments to make any logical converging paths.

Mixture of Gaussians To show that our method works well, we start by applying it on a very basic and simple example with two parameters. This first example is the mixture of Gaussians:

- $\theta_1 \sim N(0, \sigma_1^2)$ and $\theta_2 \sim N(0, \sigma_2^2)$ where $\sigma_1^2 = 10$, $\sigma_2^2 = 1$
- $x_i \sim N(\theta_1, \sigma_x^2) + \frac{1}{2}N(\theta_1 + \theta_2, \sigma_x^2)$ where $\sigma_x^2 = 2$

We draw 100 data points from the model using two modes, the primary where $\theta_1 = 0$ and $\theta_2 = 1$ and the secondary mode where $\theta_1 = 1$ $\theta_2 = -1$. The stochastic gradient noise and the injected noise are shown in figure 2. The stochastic gradient Langevin algorithm has two main phases, the first phase where the stochastic gradient noise dominates the injected noise, and a second phase where the converse occurs.

Logistic Regression For this second example, we apply stochastic gradient Langevin algorithm to a Bayesian logistic regression model. We will be using the data from the *UCI* dataset, more specifically, the *a9a* dataset which consists of 32561 observations and 123 features.

$$p(y_i|x_i) = \sigma(y_i \beta^T x_i)$$

Where β are the parameters, and σ is the sigmoid function. We use a Laplace prior for β and with a scale of 1. The gradient of the log likelihood is: $\frac{\partial}{\partial \beta} \log p\left(y_i \mid x_i\right) = \sigma\left(-y_i \beta^T x_i\right) y_i x_i$. And the prior is just the $-\operatorname{sign}(\beta)$.

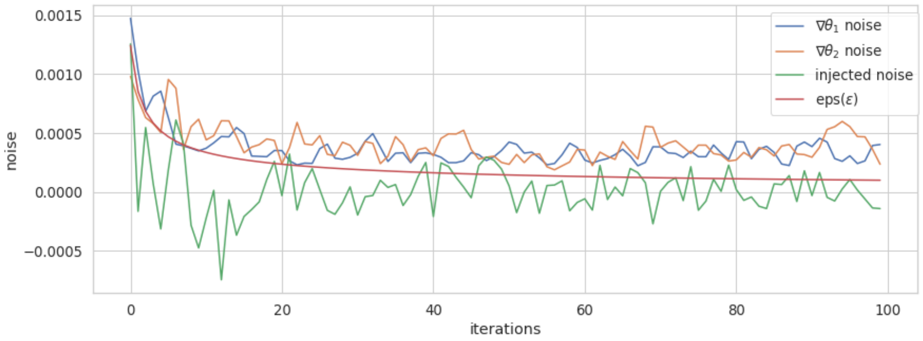

Figure 2: Stochastic gradient noise and the injected noise

Independent Components Analysis Finally, in the paper they also represent one final experiment concerning the Independent Components Analysis. This experiment although trying to implement it is more complicated and we were particularly intrigued to understand and thereby implement it. The experiment started from a probabilistic model specification that assumes independent, heavy tailed marginal distributions,

$$p(\mathbf{x}, W) = |\det(W)| \left[\prod_i p_i\left(\mathbf{w}_i^T\mathbf{x}\right)\right] \prod_{ij} \mathcal{N}\left(W_{ij}; 0, \lambda\right)$$

5 Thoughts & Takeaways

- The paper is will written and well explained.
- We managed to understand the majority of the proofs and theoretical aspects.
- The paper does represent an experimental implementation of the methods, but roughly explains the bayesian concepts behind it.
- We felt a big lack of theoretical groundings, as most proofs are just explained intuitively.
- The proposed method is very interesting since it brings delicately together two different aspects of physics and machine learning, namely stochastic gradient and Langevin dynamics.
- The experiments are done using a variety of basic and well known models (e.g. mixture of Gaussians, logistic regression, etc.).
- The part about Independent Components Analysis is much complicated to process and therefore unimplemented.

References

[1] M. Welling and Y. 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.

[2] Ahn, S., Korattikara, A., & Welling, M. (2012). Bayesian posterior sampling via stochastic gradient Fisher scoring. arXiv preprint arXiv:1206.6380.