Bayesian Learning via Stochastic Gradient Langevin Dynamics

Narjisse Lasri
IMT Nord Europe
narjisse.lasri@etu.imt-lille-douai.fr

Amal Chaoui
Centrale Lille
amal.chaoui@centrale.centralelille.fr

Abstract

Stochastic gradient optimization is widely used in most learning tasks since it provides faster convergence. Stochastic Gradient Langevin Dynamics (SGLD) was introduced to leverage the power of SGD in the Bayesian framework. The main idea is to add a noise term that scales as the step size in the gradient update expression. Although the algorithm achieves good performance for different learning tasks, we show that it results in a poor mixing behavior of posterior samples. To address this issue, we suggest a variant algorithm based on Hamiltonian Monte Carlo that decorrelates the moves and improves the chain mixing.

1 Introduction

In the last few years the number of large scale machine learning datasets has been increasing. Methods with optimization-based approaches, especially stochastic optimization or Robbins-Monro algorithms have been very successful to process them. One of their main advantage is the possibility to only pass once through the data, which considerably reduces the memory requirements. Bayesian methods, on the other hand, are one of the categories of methods dropped by the recent advances in large-scale machine learning. This can be explained by the fact that each iteration of the algorithms requires computations on the whole data set. Nevertheless, Bayesian methods are attractive in their ability to capture the uncertainty of the learned parameters and to avoid overfitting. The method presented in the article Bayesian Learning via Stochastic Gradient Langevin Dynamics [Welling and Teh(2011)] combines Robbins-Monro type algorithms with Langevin dynamics. This association limits the memory requirements while avoiding overfitting.

We first explained the concepts of the Stochastic Gradient Langevin Dynamics (SGLD) method in section 2. We then presents the application of the SGLD with logistic regression in section 3. In section 4, we show the strenghs and the weaknesses of the method. We finally suggest a contribution to the paper with a variant algorithm based on Hamiltonian dynamics in section 5.

2 Stochastic Gradient Langevin Dynamics

2.1 Theoretical setting

Notations Let θ denote a parameter vector, with $p(\theta)$ a prior distribution, and $p(x \mid \theta)$ the probability of data item x given the model parameterized by θ. The posterior distribution of a set of N data items $X = \{x_i\}_{i=1}^{N}$ is: $p(\theta \mid X) \propto p(\theta) \prod_{i=1}^{N} p(x_i \mid \theta)$.

Stochastic Optimization The Stochastic Optimization is a class of methods used to obtain the maximization of the log-posterior. The general idea is that an approximative gradient is computed on a subset of n data items ($X_t = x_{t1}, ..., x_{tn}$) instead of the whole dataset. At each iteration, it is updated as follow :

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

Where (ϵ_t) is a sequence of step sizes and satisfy the property:

$$\sum_{t\geq 1} \epsilon_t = \infty \quad \sum_{t\geq 1} \epsilon_t^2 < \infty$$

The issue with this method is that they do not capture parameter uncertainty and can potentially overfit data. Bayesian approaches can overcome this problem.

Markov chain Monte Carlo through Langevin dynamics Langevin dynamics is a class of Markov chain Monte Carlo (MCMC) techniques. It takes gradient steps and injects Gaussian noise into the parameter updates.

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

However, the main issue is that each iteration requires computations over the whole dataset. The consequence is a very high computational costs for large datasets.

Stochastic Gradient Langevin Dynamics The article proposes to combine the stochastic gradient algorithms and Langevin dynamics to benefit the advantages of both method. The proposed update is :

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

In the initial phase the algorithm imitates an efficient stochastic gradient ascent algorithm. In the second phase the injected noise dominates and the algorithm approximates a Langevin dynamics algorithm. This allows efficient use of large datasets while allowing for parameter uncertainty to be captured in a Bayesian manner.

2.2 Posterior sampling

Transition into Langevin dynamics phase The transition from the burn-in phase of optimization into the posterior sampling phase happens when the variance of the injected noise $\eta_t \sim \mathcal{N}(0, \epsilon_t M)$ dominates the variance of the stochastic gradient step. M is a preconditioning matrix introduced to better adapt the step sizes to the local structure of the posterior. This defines a condition on the sample threshold α for the transition to happen, which can be expressed $\frac{\epsilon_t N^2}{4n}\lambda_{max}(M^{1/2}V_s M^{1/2}) = \alpha \ll 1$, with $\lambda_{max}(A)$ being the largest eigenvalue of A.

Posterior expectation estimation Once the posterior samples $\{\theta_t\}_{1\leq t\leq T}$ collected, one can estimate the posterior expectation $\mathbb{E}[f(\theta)] \approx \frac{1}{T}\sum_{t=1}^{T} f(\theta_t)$. However, since the step size ϵ_t decreases, the mixing rate of the chain decreases too. The authors, therefore, proposed to weight the samples using the step sizes: $\mathbb{E}[f(\theta)] \approx \frac{\sum_{t=1}^{T}\epsilon_t f(\theta_t)}{\sum_{t=1}^{T}\epsilon_t}$, to prevent over-emphasizing the tail end of the distribution.

3 Application and results: logistic regression

We applied SGLD to classification tasks using 2 toy datasets of linearly and non-linearly separable classes, and a real bank marketing dataset. For this, we consider the following Bayesian logistic regression model with a Gaussian prior over the regression parameter θ:

$$\theta \sim \mathcal{N}(0, I)$$

$$y_i \underset{iid}{\sim} \mathcal{B}(\sigma(y_i x_i^T \theta)), \;\; \forall i \in \{1, \dots, N\}$$

We illustrate the obtained results in the following sections.

3.1 Toy datasets

Our first application of the SGLD was performed on linearly separable dataset. We observed two phases on the log-posterior: the first one where the log-posterior grows rapidly (SGD) and the second one where the algorithm stabilizes in a high probability area of the posterior distribution and samples (Langevin Dynamics). The accuracy was good and the two classes were well-separated by SGLD.

In the second application, we used a circular dataset. Optimising SGLD over small batches allowed more flexibility in exploring the posterior domain, hence faster convergence for this application. SGLD also classified well on this other dataset with an accuracy of 100%.

3.2 Bank marketing dataset

The bank marketing dataset consists of data (17 features) collected from 11162 customers by a Portuguese banking institution. The classification goal is to predict whether the client will subscribe to a term deposit after a phone call marketing advertisement.

We compute the mean and standard deviation of the test and training accuracy over the posterior samples obtained using SGLD logistic regression. We compare the results obtained by using SGLD logistic regression to that of the SGDClassifer from `sklearn`. The classification results are illustrated in table 2 for $a = 0.01, b = 500, \gamma = 0.6, M = I$ and batch size of 10 and 5000 iterations. We obtain similar performance as a simple SGD algorithm while taking advantage of the Bayesian framework that enables integrating uncertainty in the interpretation of the results.

Table 1: Training and test accuracies obtained by SGLD and SGDClassifier

	SGLD	NUTS
Train acc.	0.801 ± 0.001	0.801 ± 0.001
Test acc.	0.788 ± 0.003	0.790 ± 0.002

Figure **??** illustrates the trace plot of the first 4 parameters of θ from the posterior samples taken as the last 1000 samples of the chain. We clearly how correlated the chain moves are.

4 Strengths and weaknesses

Strenghts The combination of SGD and BML allows SGLD to benefit from two main advantages :

- Using BML while leveraging the power of the mini-batch optimization
  - SGD: computing over mini-batch and allowing fast optimization
  - BML: quantifying uncertainty while regularising the model thank to the prior that acts as a regularizer
- Perform optimization and the posterior sampling in the same algorithm

Weakenesses Although SGLD seems to reconcile the Bayesian framework and the frequentist framework, there are some limitations to the use of this MCMC algorithm:

- Need to tune the step size parameters a, b, γ to obtain a faster convergence and decrease the chaotic behavior.
- Poor mixing during the sampling phase (c.f. figure). This is expected since the step sizes become smaller throughout iterations. This means that the chain can easily get trapped in one single mode of the posterior, making the algorithm quite unsuitable to sample from complex posterior distributions with multiple modes.

5 Contribution: Stochastic Gradient Hamiltonian Dynamics

In this section, we attempt to address the poor mixing issue of the MCMC SGLD during the posterior sampling phase. For this purpose, we take advantage of the Hamiltonian dynamics to untangle the correlation between the posterior samples by adding an additional variable p (momentum) while computing the updates on a batch of training samples only. We call this MCMC variant Stochastic Gradient Hamiltonian Dynamics (SGHD). We recall the expression of the Hamiltonian

$$H(q, p) = -\log \pi(q) + p^T M_H p$$

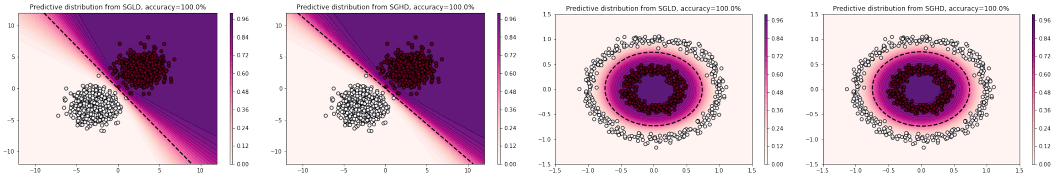
with π being the posterior distribution we want to sample from, and M_H a mass matrix that encodes relevant posterior information.

We use the leapfrog integrand to approximate the Hamiltonian flow and update p and q using the following equations:

$$p_{1/2} = p + \frac{h}{2}\nabla \log \pi(q), \quad q' = q + hMp_{1/2}, \quad p' = p_{1/2} + \frac{h}{2}\nabla \log \pi(q').$$

We assess the performance of SGHD against SGLD on 2 classification tasks: linearly and non-linearly separable classes, with 1500 and 700 samples resp. We set the gradient step size parameters to $a = 0.1, b = 40, \gamma = 0.7$. We take a batch size of $N/2 = 750$ and 50 samples resp. for the linearly and non-linearly separable classes. We also provide the trace plots of the parameters of θ for the linearly separable classes case computed by SGLD and SGHD in figures **??**, **??**.

Figure 1 illustrates the decision boundaries obtained for each classification task. We also provide the trace



(a) SGLD (b) SGHD (c) SGLD (d) SGHD

Figure 1: Decision boundaries obtained by SGLD and SGHD for different classification tasks.

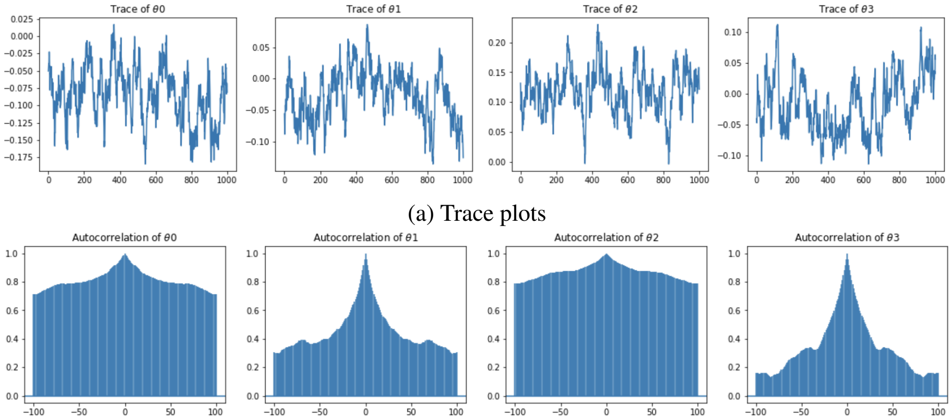
Table 2: Training and test accuracies obtained using SGLD, SGHD, and NUTS.

	SGLD	SGHD	NUTS
Train acc.	0.801 ± 0.001	0.713 ± 0.042	0.801 ± 0.001
Test acc.	0.788 ± 0.003	0.708 ± 0.040	0.790 ± 0.002

Clearly, SGHD improves the mixing behavior and decorrelates the chain moves. The same global tendency was also observed for the non-linearly separable class case. This advantage in favor of the Hamiltonian process can therefore be exploited in the hope of not getting trapped in a single posterior mode. This hypothesis however needs to be confirmed on multimodal posterior distributions.

6 Conclusion

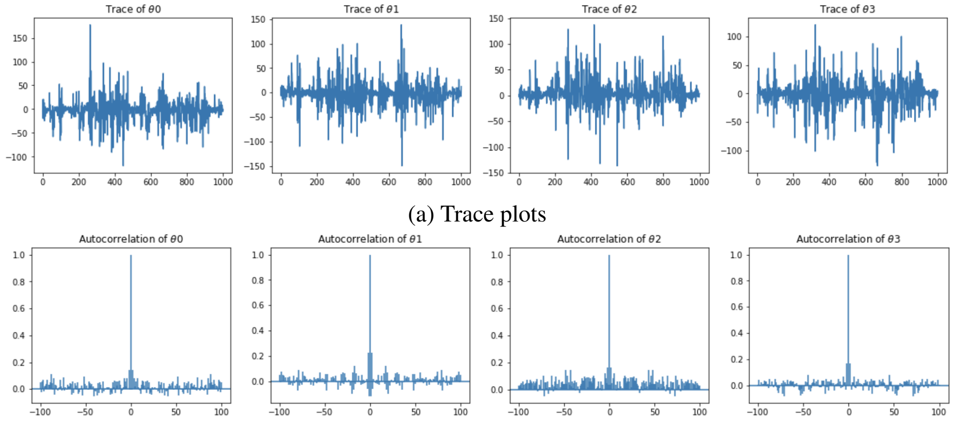
In this report, we illustrated the performance of SGLD in sampling from the posterior for the logistic regression classification task. We showed that the poor mixing of the obtained chain can result in



(a) Trace plots

(b) Autocorrelation

Figure 2: Trace plots of the first 2 parameters of θ obtained using SGLD.



(a) Trace plots

(b) Autocorrelation

Figure 3: Trace plots of the first 2 parameters of θ obtained using SGHD.

approximating a single mode only in the case of multimodal posterior distributions. We suggest SGHD, a variant based on the Hamiltonian dynamic, and showed that it improves the chain mixing and performs as well as SGLD on 2 classification tasks. However, it is important to mention that the performance of SGHD relies on the information encoded in the mass matrix M_H. For the experiments, we computed it as the covariance of the posterior samples obtained from SGLD. On the other hand, we have noticed that SGHD is sensitive to the size of the batch and the initial θ_0, which require manual tuning to guarantee convergence. Other areas of improvement can also include the tuning of the gradient step size parameters a, b, γ that affect the performance of both SGLD and SGHD.

References

[Welling and Teh(2011)] 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, page 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.