Monte Carlo Simulation
10-Page Essay on SGLD and HSGLD

Pinyan Xu

Department of Statistics, University of Chicago

1 Introduction

In this essay, we will look into the stochastic gradient Langevin Dynamics (SGLD) and Hamiltonian stochastic gradient Langevin Dynamics (HSGLD) methods. The SGLD method is proposed for Bayesian learning from large-scale data sets. It utilizes the Robbins-Monro algorithms that stochastically optimize the cost function by dealing with a small subset of data at each iteration and updating model parameters by small gradient steps[7]. Combining with the stochastic optimization algorithm, SGLD uses Langevin dynamics to inject noise into the parameter to make sure the parameters will converge to the full posterior distribution[7]. The HSGLD algorithm then marries the Hamilton Monte Carlo (HMC) methods with SDLD in order to remedy for the limitation of the former and maintain the desired target distribution at the same time[2].

The essay will be divided into two parts. In first part, I will introduce and analyze the SGLD algorithm; in the second part, I will then move to HSGLD method and evaluate its performances.

The detailed outline of the essay is as follows. Section 2 and 3 will be focusing on SGLD and HSGLD respectively. Subsection 2.1 and 2.2 introduce the two components of SGLD, namely, stochastic optimization and Langevin dynamics. In the next subsection, the SGLD algorithm is presented and evaluated in terms of its convergence to the posterior distribution. In section 2.4, I would like to examine the algorithm with a simple example. Section 3.1 includes an introduction of Hamilton Monte Carlo method and its limitation. Then I will discuss the naive stochastic gradient HMC method as a resolution. Following that, section 3.2 offers a method to improve the naive algorithm by adding a "friction" to the momentum updates. Section 3 concludes with simulations of the HSGLD algorithm.

2 Stochastic Gradient Langevin Dynamics

2.1 Stochastic Optimization

Given data $X = \{x_1, x_2, ..., x_N\}$ which we assume to be sampled from a data-generating model that depends on parameter vector θ, the parameter vector data has a prior distribution $\pi(\theta)$, and a posterior distribution

$$p(\theta|X) \propto \pi(\theta) \prod_{i=1}^{N} p(x_i|\theta),$$

where $\prod_{i=1}^{N} p(x_i|\theta) = p(X|\theta)$ denotes the likelihood function.
The task of optimization is to find the maximum a posteriori(MAP) parameters θ^*. What I will introduce here is a class of methods called stochastic optimization or Robbins-Monro algorithms.

In the setting of Bayesian inference, the problem we are trying to solve is $g(\theta^*) = \min_\theta \pi(\theta) \prod_{i=1}^N p(x_i|\theta)$. Or equivalently, we can rewrite it as the optimization of log-likelihood function,

$$g(\theta^*) = \min_\theta \log(\pi(\theta)) + \sum_{i=1}^N \log(p(x_i|\theta)).$$

Since the log-likelihood function is convex, the problem reduces to finding the solution of the equation $\nabla g(\theta^*) = 0$. Robins and Monro proposes an approximation method by recursion

$$\theta_{k+1} = \theta_k - \epsilon \nabla g(\theta_k),$$

which is similar to the Newton's method[4]. When dealing with large scale data, the algorithm will process a small subset of n data points $X_t = \{x_{t1}, x_{t2}, ..., x_{tn}\}$ at each iteration t. The parameter θ is updated by

$$\Delta\theta_t = \frac{\epsilon_t}{2}\left(\nabla \log(\pi(\theta_t)) + \frac{N}{n}\sum_{i=1}^n \nabla \log(p(x_i|\theta_t))\right). \tag{2.1}$$

According to Bouleau and Lepingle[3], for θ_t to converge almost surely, we still need several conditions:
(1) $\epsilon_t \geq 0, \forall t \geq 0$;
(2) $\sum_{t=0}^\infty \epsilon_t = \infty$;
(3) $\sum_{t=0}^\infty \epsilon_t^2 < \infty$;
(4) $|X_t| \leq B,$ for a fixed bound B;
(5) $g(\theta)$ is strictly convex.
We can immediately see that condition (4) and (5) are satisfied in our setting. In practice, ϵ_t is often chosen to be $a(b+t)^{-\gamma}$ where $\gamma \in (0.5, 1]$ to ensure convergence.

Algorithm 2.1 Stochastic Gradient Algorithm

```
Input: Data set X, prior distribution $\pi(\theta)$, likelihood function $p(x|\theta)$ and sequence of step size $\epsilon_t$
Initial Draw: Draw $\theta_0 \sim \pi(\theta)$.
At iteration t+1:
Step 1: Select a small subset of the data set X $\{x_{t1}, ..., x_{tn}\}$.
Step 2: Update $\theta_{t+1} = \theta_t - \frac{\epsilon_t}{2}\left(\nabla \log(\pi(\theta_t)) + \frac{N}{n}\sum_{i=1}^n \nabla \log(p(x_i|\theta_t))\right)$.
Output: $\theta^* \sim p(\theta|X)$.
```

As Welling and Teh point out, the problem with the above algorithm is that it does not capture parameter uncertainty and may cause the issue of overfitting[7]. In order to account for parameter uncertainty, they propose a class of Markov chain Monte Carlo(MCMC) techniques called Langevin dynamics, which I will briefly outline in the next subsection.

2.2 Langevin Dynamics

In physics, Langevin equation is a stochastic differential equation that models the dynamics of molecular systems. The equation has the following form:

$$m\frac{dv}{dt} = -\lambda v + \eta(t),$$

where m stands for the mass, v the velocity and $\eta(t)$ the noise term. If we examine it carefully, we can find out that $m\frac{dv}{dt} = F = \nabla E$ where F represents the force and E is the energy function. In addition, let θ stand for the position of the particle. Then we get $v\Delta t = \Delta\theta$. Now the Langevin equation can be transformed into a form that is similar to equation (2.1)

$$\Delta\theta_t = -\frac{\Delta t}{\lambda}\nabla E + \frac{\Delta t}{\lambda}\eta(t).$$

Equating $\frac{\epsilon}{2}$ with $\frac{\Delta t}{\lambda}$, and E with the log-likelihood function, the Langevin dynamics gives us a way to update the parameter with Gaussian noise:

$$\Delta\theta_t = \frac{\epsilon}{2}\left(\nabla\log(\pi(\theta_t)) + \sum_{i=1}^{N}\nabla\log(p(x_i|\theta_t))\right) + \eta(t), \tag{2.2}$$

where $\eta(t) \sim N(0,\epsilon)$.

2.3 Stochastic Gradient Langevin Dynamics

Combining Stochastic Gradient method and Langevin dynamics, we will get the following algorithm.

Algorithm 2.2 Stochastic Gradient Langevin Dynamics Algorithm

```
Input: Data set X, prior distribution $\pi(\theta)$, likelihood function $p(x|\theta)$ and sequence of step size $\epsilon_t$
Initial Draw: Draw $\theta_0 \sim \pi(\theta)$.
At iteration t+1:
Step 1: Draw $\eta_t \sim N(0, \epsilon_t)$.
Step 2: Select a small subset of the data set X $\{x_{t1}, ..., x_{tn}\}$.
Step 3: Update $\theta_{t+1} = \theta_t - \frac{\epsilon_t}{2}\Big(\nabla\log(\pi(\theta_t)) + \frac{N}{n}\sum_{i=1}^{n}\nabla\log(p(x_i|\theta_t))\Big) + \eta_t$.
Output: $\theta^* \sim p(\theta|X)$.
```

Next, I would like to show that θ_t in the algorithm will approach samples from the posterior distribution as $t \to \infty$[7]. Let

$$g(\theta) = \nabla\log(\pi(\theta)) + \sum_{i=1}^{N}\nabla\log(p(x_i|\theta)) \tag{2.3}$$

denotes the gradient of log-likelihood function with the whole data set at θ and

$$h_t(\theta) = \nabla\log(\pi(\theta)) + \frac{N}{n}\sum_{i=1}^{N}\nabla\log(p(x_i|\theta)) - g(\theta) \tag{2.4}$$

The stochastic gradient is $g(\theta) + h_t(\theta)$, where $h_t(\theta)$ involves stochasticity of the data points chosen at time t with variance $V(\theta_t)$. Furthermore, by injecting a Gaussian noise, updates of the parameters include a second source of stochasticity with variance ϵ_t.
As $t \to \infty$, $\epsilon_t \to 0$ and the injected Gaussian noise will dominate the stochastic gradient noise. We can also observe that when $\epsilon_t \to \infty$, it will average out the stochaticity in the gradients so that the MH rejection probability will reduce to 0. It then shows that for large t, (2.4) defines a non-stationary Markov chain such that the transition kernel will have the posterior over θ as the equilibrium[7].

Since the Markov chain is non-stationary, another question remains: will θ_t finally converge to the posterior distribution $p(\theta|X)$? The answer is yes. We can find a subsequence $\{t_1, t_2, ...\}$ such that $\sum_{t=t_s+1}^{t_{s+1}} \epsilon_t \to \epsilon_0$ as $s \to \infty$. The subsequence $\{\theta_{t_1}, \theta_{t_2}, ...\}$ will converge to the posterior, and hence the whole sequence converges too. More proof details are presented in [7].

2.4 Experiments with SGLD

Now we can run a simple simulation of the SGLD algorithm.

$$\theta_1 \sim N(0, \sigma_1^2); \quad \theta_2 \sim N(0, \sigma_2^2)$$

$$x_i \sim N(\theta_1 + \theta_2, \sigma_x^2)$$

where $\sigma_1^2 = 5, \sigma_2^2 = 2$ and $\sigma_x^2 = 2$. For the step size, I chose ϵ_t to be $0.01(10+t)^{-0.55}$ which will decrease from 0.002 as t increases.
Based on the setting of the experiments, we will have

$$\Delta\theta_t^1 = -\frac{1}{\sigma_1^2}\theta_t^1 + \frac{N}{n}\sum_{i=1}^{n}\frac{x_i - \theta_t^1 - \theta_t^2}{2}$$

$$\Delta\theta_t^2 = -\frac{1}{\sigma_2^2}\theta_t^2 + \frac{N}{n}\sum_{i=1}^{n}\frac{x_i - \theta_t^1 - \theta_t^2}{2}.$$

When 100 data points are drawn from the setting $\theta_1 = 0$, $\theta_2 = 1$, we conduct the experiment using SGLD and estimate the parameters. As we can see from the Figure 1, the simulation is roughly accurate.

Another 100 data points are drawn from the setting $\theta_1 = 1$, $\theta_2 = -1$. After conducting the simulation, we get the result in Figure 2. We can drawn the same conclusion as previous section that the algorithm converges to the true posterior θ^*.

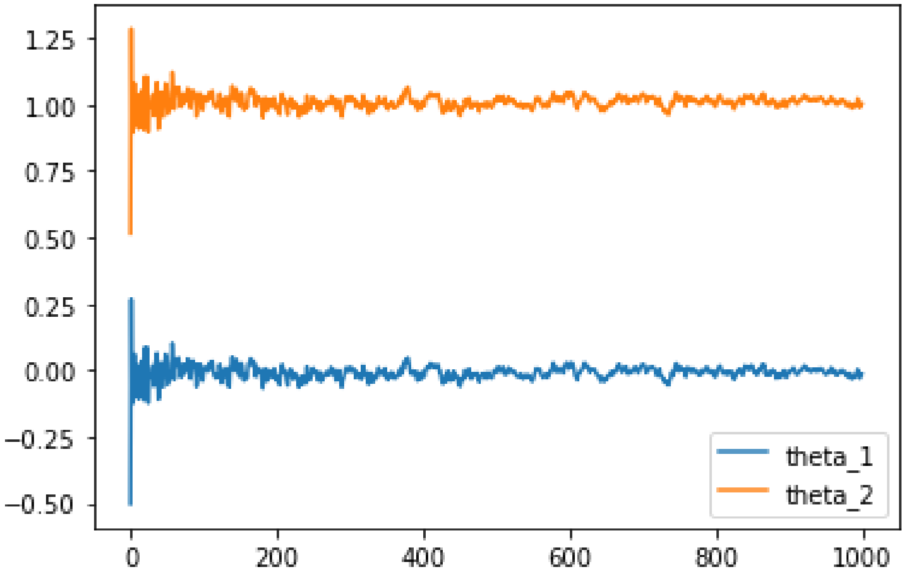

Figure 1 SGLD on $\theta_1 = 0$, $\theta_2 = 1$
With iteration = 1000

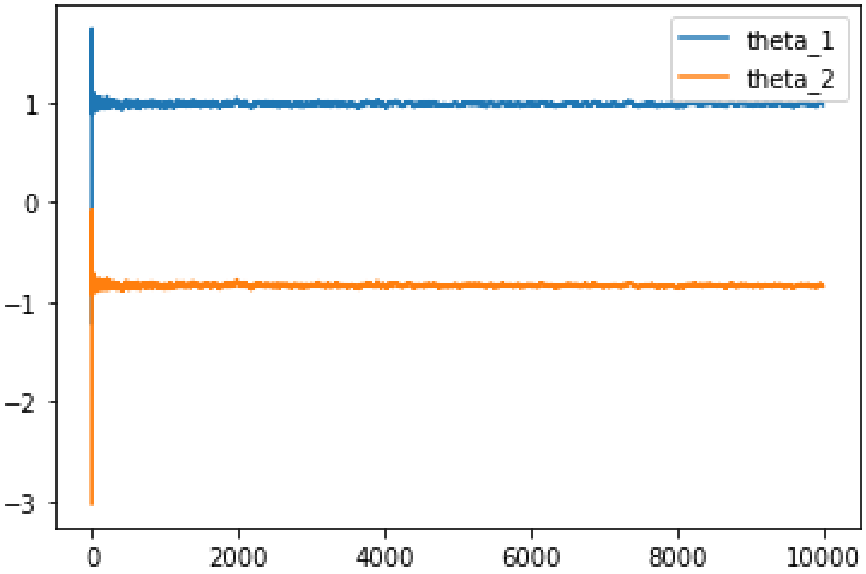

Figure 2 SGLD on $\theta_1 = 1$, $\theta_2 =$ -1
With iteration = 10000

3 Hamiltonian Stochastic Gradient Langevin Dynamics

The key idea of Hamiltonian Monte Carlo(HMC) sampling is to enable more efficient use of the state space and avoid the limitation of local behavior of Random Walk

Metropolis-Hastings approach. However, the method requires gradient computation for simulation of the Hamiltonian dynamics which can be difficult or even impossible when dealing with large sample size of data[2]. One possible remedy is to apply the stochastic gradient method that is introduced in section 2.1, estimating the gradient from a subset of the data at each iteration.

Unfortunately, the naive stochastic gradient HMC no longer leads to Hamiltonian dynamics with the desired distribution. Chen, Fox and Guestrin then propose an alternative method, the HSGLD approach with friction added to the momentum updates[2].

In this section, I will outline the Hamiltonian Monte Carlo algorithm first, and then explore the option of combining HMC with stochastic gradient algorithm and show that the naive Stochastic Gradient HMC method will behave poorly. In section 3.3, I will present a modification to the naive Stochastic Gradient HMC method incorporating friction, that is, the HSGLD method. In section 3.4, I will apply the methods to experiments that illustrate their behaviors.

3.1 Hamiltonian Monte Carlo Methods

Hamiltonian dynamics operates with a position vector q and a momentum vector p. We write our target function as

$$f(q) \propto \exp(-V(q)),$$

where V is the potential energy function. In the setting of this paper, we let $f(q)$ to be the posterior distribution $p(\theta|\{x_i\})$ and V to be the negative log-likelihood function

$$V = -\log(\pi(\theta)) - \sum_{i=1}^{N} \log(p(x_i|\theta)).$$

The Hamiltonian $H(\theta, r)$ is defined as

$$H(\theta, r) = V(\theta) + K(r),$$

where $K(r) = \frac{1}{2} r^T M^{-1} r$ for mass matrix M. Given the above Hamilton, the Hamilton equations are

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial r} = M^{-1} r$$
$$\frac{dr}{dt} = \frac{\partial H}{\partial \theta} = -\nabla V(\theta).$$

In practice, we often choose M to be the identity matrix and apply the leapfrog method which is time reversible and conserves volumes. With step size ϵ, we can proceed with the following steps:

$$r(t + \frac{1}{2}\epsilon) = r(t) - \frac{\epsilon}{2} \nabla V(\theta(t)) \tag{3.1}$$

$$\theta(t + \epsilon) = \theta_t + \epsilon \, r(t + \frac{1}{2}\epsilon) \tag{3.2}$$

$$r(t + \epsilon) = r(t + \frac{1}{2}\epsilon) - \frac{\epsilon}{2} \nabla V(\theta(t + \epsilon)). \tag{3.3}$$

Let $\psi_\lambda(\theta, r)$ be the symplectic numerical integrator that performs steps (3.1)-(3.3) with initial value $(\theta(t), r(t)) = (\theta, r)$ and repeat for $\lfloor \frac{\lambda}{\epsilon} \rfloor$ times.

Algorithm 3.1 Hamiltonian Monte Carlo (HMC)

```
Input: Initialization $\theta_0, r_0$, step size $\epsilon$, duration parameter $\lambda$.
At iteration t:
Step 1: Sample $r_{new} \sim N(0, M)$.
Step 2: Set $(\theta^*, r^*) = \psi_\lambda(\theta_t, r_{new})$
Step 3: Set
    $(\theta_{t+1}, r_{t+1}) = \begin{cases} (\theta^*, r^*), & \text{w.p. } \min\{1, \exp(H(\theta_t, r_{new}) - H(\theta^*, r^*))\} \\ (\theta_t, -r_{new}), & \text{w.p. } 1 - \min\{1, \exp(H(\theta_t, r_{new}) - H(\theta^*, r^*))\} \end{cases}$    (3.4)
Output: $\theta_1$, $\theta_2$, ..., $\theta_N$.
```

Remark 3.1. The HMC algorithm behaves better than the Random Walk Metropolis Hasting algorithms in high dimensional cases. However, the method requires us to compute the gradient of V, which can be difficult or even impossible sometimes.

3.2 Naive Stochastic Gradient HMC Algorithm

As stated in remark 3.1, in practice, it may be overwhelmingly expensive to compute the gradient of the log-likelihood function on the whole data set, so now we only consider a noisy estimate based on a small batch of n data points:

$$\nabla \tilde{V}(\theta) = -\nabla \log(\pi(\theta)) - \frac{N}{n} \sum_{i=1}^{n} \nabla \log(p(x_i|\theta)). \tag{3.5}$$

We assume the observation x are independent and approximate this noisy gradient as

$$\nabla \tilde{V}(\theta) \approx \nabla V(\theta) + N(0, \Sigma(\theta)), \tag{3.6}$$

where $\Sigma(\theta)$ represents the covariance of the stochastic gradient noise[2]. As n increases, the approximation will be more accurate. But we also want n to be small in order to reduce computational costs. Empirically, n being several hundreds would be sufficient to guarantee the algorithm accuracy with central limit theorem, and would also serve our purpose of computation costs reduction[1].

Naive Stochastic Gradient HMC methond simply replaces $\nabla V(\theta)$ in the HMC algorithm with $\tilde{V}(\theta)$. The resulting Hamilton equations are:

$$d\theta = M^{-1} r \; dt \tag{3.7}$$

$$dr = -\nabla V(\theta) \; dt + N(0, 2B(\theta) dt), \tag{3.8}$$

where $B(\theta) = \frac{1}{2}\epsilon\Sigma(\theta)$ is the diffusion matrix of the gradient noise. Next, we want to show that when $B(\theta)$ is nonzero, the joint distribution $p(\theta, r)$ is no longer invariant under the dynamics (3.7) and (3.8).

Theorem 3.2. *Let $p_t(\theta, r)$ denote the joint distribution of θ and r at time t with the dynamics (3.7) and (3.8). Let the entropy of p_t be $h(p_t) = -\int_{\theta,r} f(p_t(\theta, r))d\theta dr$, where $f(x) = x \ln x$. Assume p_t has density and gradient vanishing at infinity, and the gradient vanishes faster than $\frac{1}{\ln p_t}$. Then $h(p_t)$ increases over time with rate*

$$\partial_t h(p_t(\theta, r)) = \int_{\theta,r} f''(p_t)(\nabla_r p_t(\theta, r))^T B(\theta)\nabla_r p_t(\theta, r)d\theta dr.$$

Since $B(\theta)$ is positive semi-definite, $\partial_t h(p_t(\theta, r)) \geq 0$.

Theorem 3.1 is true if we assume that the diffusion matrix $B(\theta)$ is positive definite, and $\nabla_r p_t(\theta, r) \neq 0$ for all t. Since the noise-free Hamiltonian dynamics preserves entropy and the additional noise term strictly increases entropy by our assumption, the entropy function strictly increases over time as well. The detailed proof is provided in [2].

Now for the purpose of reductio, let us suppose that the joint distribution $p(\theta, r) = \frac{1}{Z}\exp(-H(\theta, r))$ is invariant under the dynamics (3.7) and (3.8) and $H(\theta, r) \to \infty$ as $\|\theta\|, \|r\| \to \infty$. Then as $\|\theta\|, \|r\| \to \infty$, $p(\theta, r) \to 0$, and $f(p(\theta, r)) \to 0$. Now we can apply theorem 3.1, and prove that the entropy increases over time. However, this contradicts our supposition that the joint distribution is invariant. We can then conclude that the Naive Stochastic Gradient HMC algorithm will not lead to the desired target distribution.

3.3 Hamiltonian Stochastic Gradient Langevin Dynamics

In order to resolve the problem of the naive stochastic gradient HMC algorithm, we consider to add a "friction" term to the momentum update (3.8):

$$d\theta = M^{-1}r\ dt \tag{3.9}$$

$$dr = -\nabla V(\theta)\ dt - B(\theta)M^{-1}rdt + N(0, 2B(\theta)dt), \tag{3.10}$$

Let $\psi_\lambda(\theta, r)$ be the symplectic numerical integrator that performs steps (3.9) and (3.10) with initial value $(\theta(t), r(t)) = (\theta, r)$ and repeat for $\lfloor \frac{\lambda}{\epsilon} \rfloor$ times. The friction term $-B(\theta)M^{-1}dt$ helps reducing the influence of the noise. Comparing the above equations with Langevin dynamics (2.2), we can see that the formulas are similar to each other. The type of dynamics of (3.8) and (3.9) is usually referred as the second-order Langevin dynamics[6].

Theorem 3.3. $p(\theta, r) \propto \exp(-H(\theta, r))$ *is the stationary distribution of the dynamics described by (3.9) and (3.10).*

Proof. Consider G = $\begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$ and D = $\begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix}$, where D is the symmetric diffusion matrix.
Rewrite (3.9) and (3.10) in matrix multiplication form:

$$d\begin{bmatrix} \theta \\ r \end{bmatrix} = \begin{bmatrix} 0 & -I \\ I & B \end{bmatrix}\begin{bmatrix} \nabla V(\theta) \\ M^{-1}r \end{bmatrix}dt + N(0, 2\tau Ddt) = -[D + G]\nabla H(\theta, r)dt + N(0, 2\tau Ddt).$$

Apply the Fokker-Plank Equation

$$\partial_t p_t(z) = -\nabla^T\{g(z)[p_t(z)] + \nabla^T[D(z)\nabla p_t(z)]\}$$

to the above equation and set $g(z) = -[D+G]\nabla H(\theta, r)$. We get

$$\partial_t p_t(\theta, r) = -\nabla^T\{-[D+G]\nabla H(\theta, r)[p_t(\theta, r)] + \nabla^T[\tau D\nabla p_t(\theta, r)]\}.$$

Note that $\nabla^T[G\nabla p_t(\theta, r)] = 0$. Then the distribution increases with rate:

$$\partial_t p_t(\theta, r) = \nabla^T\{[D+G][\nabla H(\theta, r)p_t(\theta, r) + \tau\nabla p_t(\theta, r)]\}.$$

If we substitute the target distribution $p(\theta, r) = \exp(-\frac{1}{\tau}H(\theta, r))$, we will get

$$\partial_t p(\theta, r) = \exp(-\frac{1}{\tau}H(\theta, r))\nabla H(\theta, r) + \tau\nabla\exp(-\frac{1}{\tau})H(\theta, r) = 0.$$

Therefore, we have shown that $p(\theta, r) \propto \exp(-H(\theta, r))$ is the stationary distribution of the dynamics described by (3.9) and (3.10).

□

Algorithm 3.2 HSGLD algorithm

```
**Input**: Initialization $\theta_0, r_0$, step size $\epsilon$, duration parameter $\lambda$.
**At iteration t:**
**Step 1**: Sample $r_{new} \sim N(0, M)$.
**Step 2**: Let $(\theta^*, r^*) = \psi_\lambda(\theta_t, r_{new})$
**Step 3**: Set $(\theta_{t+1}, r_{t+1}) = (\theta^*, r^*)$ with no Metropolis Hasting step.
**Output**: $\theta_1$, $\theta_2$, ..., $\theta_N$.
```

According to Theorem 3.3, the HSGLD algorithm will behave as desired. In the next subsection, I will evaluate its performance practically using one simple example.

3.4 Experiements on Noise Stochastic Gradient HMC and HSGLD Algorithms

Let $V(\theta) = \frac{1}{2}\theta^2$ and $\epsilon = 0.1$. Simulate (θ, r) using the Hamilton Dynamics for a burn-out period of 14000 steps and 1000 samples. For naive stochastic gradient HMC, we replace $\nabla\tilde{V}(\theta)$ with $\theta + N(0, 4)$, and for HSGLD algorithm we let $B(\theta) = 0$.

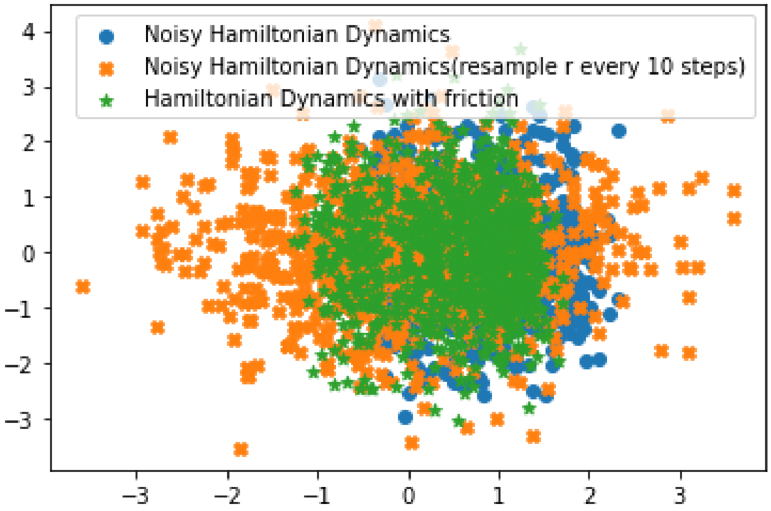

Figure 3 Noisy Hamilton Dynamics v.s. Hamilton Dynamics with friction

The above figure shows the trajectories of the algorithms. As we can see, the naive stochastic gradient HMC (noisy Hamilton dynamics) diverges from our desired stationary distribution. Resampling r for every 10 iterations helps improve the performance, but still fail to converge to the true posterior distribution. After introducing a "friction" term into the dynamics, the HSGLD algorithm corrects the divergence behavior.

Comparing the distribution of θ from HSGLD approach with the true distribution $\exp(-V(\theta))$, we get the following graph. It is not hard to see that the histogram coincides with the true distribution, so we can conclude that the HSGLD algorithm yields the target distribution.

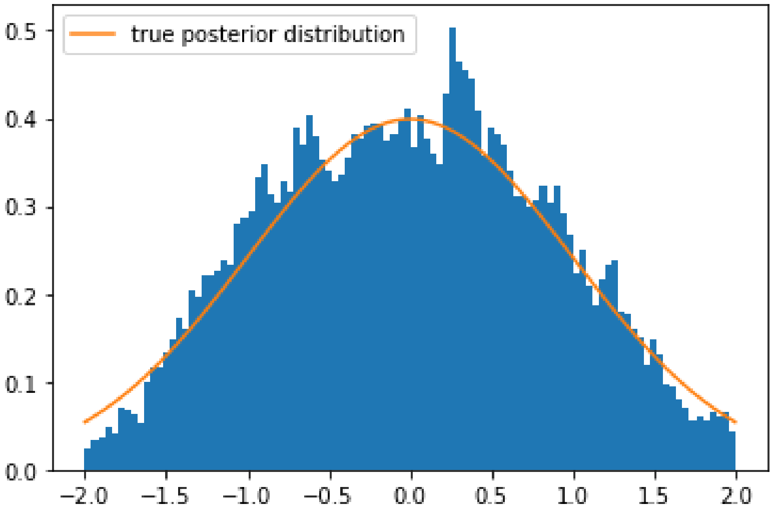

Figure 4 HSGLD result v.s. True posterior
With iteration = 10000

4 Discussion and Bibliography

The consistency of the SGLD algorithms are discussed in more details in [5] and other related articles. The proofs of theorems and corollaries involved in this paper are presented in [7] and [2].

5 Algorithms Implementation and Codes

References

[1] S. Ahn, A. Korattikara, and M. Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. In *Proceedings of the 29 th International Conference on Machine Learning*, 2012.

[2] T. Chen, E. B. Fox, and C. Guerstrin. Stochastic gradient hamiltonian monte carlo. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, 2012.

[3] B. Nicholas and D. Lepingle. *Numerical Methods for stochastic Processess.* New York: John Wiley, 1994.

[4] H. Robins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[5] Y. W. Teh, A. H. Thiery, and S. J. Vollmer. Consistency and fluctuations for stochastic gradient Langevin dynamics. *The Journal of Machine Learning.*

[6] M. Wang and G. Uhlenbeck. On the theory of brownian motion ii. *Reviews of Modern Physics*, 17(323), 1945.

[7] 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.