Bayesian Posterior Sampling via Stochastic Gradient Descent with Collisions

Quico Spaen

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-169

http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-169.html

December 10, 2019

Acknowledgement

I would like to thank Jongho (Jerry) Park for his contributions to this project.

Bayesian Posterior Sampling via Stochastic Gradient Descent with Collisions

by Quico Pepijn Spaen

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor John Canny
Research Advisor

12/10/19
(Date)

* * * * * * *

Anil J. Aswani

Associate Professor Anil Aswani
Second Reader

12-6-19
(Date)

MSc Technical Report: Bayesian Posterior Sampling via Stochastic Gradient Descent with Collisions

Quico Spaen

Advisor: Professor John Canny

Abstract

Markov Chain Monte Carlo algorithms, with step proposals based on Hamiltonian or Langevin dynamics, are commonly used in Bayesian machine learning and inference methods to sample from the posterior distribution of over model parameters. In addition to providing accurate predictions, these methods quantify parameter uncertainty and are robust to overfitting. Until recently, these methods were limited to small datasets since they require a full pass over the data per update step. New developments have enabled mini-batch updates through the use of a new mini-batch acceptance test and by combining stochastic gradient descent with additional noise to correct the noise distribution.

We propose a novel method that redistributes the stochastic gradient noise across all degrees of freedom via collisions between particles instead of inserting additional noise into the system. Since no additional noise is added to the system, the proposed method has a higher rate of diffusion. This should result in faster convergence as well as improved exploration of the posterior distribution. We observe this behavior in initial experiments on a multivariate Gaussian model with a highly skewed, and correlated distribution.

1 Introduction

The availability of large datasets and more powerful computational tools for machine learning have led to advances in diverse fields, including vision, speech recognition, and reinforcement learning. A crucial algorithmic tool that enabled machine learning on these large datasets is stochastic gradient descent (SGD) [Robbins and Monro, 1951]. In traditional gradient descent algorithms for machine learning, a step is taken in parameter space in the direction opposite to the gradient of the loss function. The loss function is commonly the sum or average of the loss of each observation in the dataset. In stochastic gradient descent, the gradient with respect to the full dataset is replaced with the gradient with regards to a small random subset of the data, known as a (mini-)batch. This reduces the computational complexity of computing the gradient in each step from $O(n)$ to $O(1)$, where n is the number of observations in the dataset.

However, not all gradient-based machine learning methods benefit from the use of stochastic gradient descent. In particular, Bayesian machine learning algorithms that rely on Markov Chain Monte Carlo (MCMC) methods [Hastings, 1970; Metropolis et al., 1953] to estimate the posterior distribution of the parameters did not. Bayesian machine learning methods generate a posterior distribution over the model parameters instead of point estimates as done by more traditional approaches. The posterior distribution measures the uncertainty for parameters, thus providing a method to prevent overfitting for models with many parameters.

MCMC algorithm are methods for sampling from a distribution of interest. They construct a Markov chain whose stationary distribution is the distribution of interest. Sampling from the distribution of interest is then done by sampling from the Markov chain after it has converged to its stationary distribution. Commonly, these algorithm apply a Metropolis-Hastings framework [Hastings, 1970; Metropolis et al., 1953]. In this framework, steps in the Markov chain are generated from a proposal density and are then accepted or rejected based on a test that ensures that the stationary distribution matches the distribution of interest. Traditionally, both the proposal generation and the acceptance test require a full pass over the data, resulting in low sample-efficiency. Both for the proposal step and the acceptance test, mini-batch methods have been proposed.

For the proposal step, Welling and Teh [2011] propose the Stochastic Gradient Langevin Dynamics (SGLD) algorithm. It is a variant of first-order Langevin dynamics that injects additional noise to ensure that noise distribution is correct. They also anneal the stepsize to zero to avoid the use of an acceptance test. Ahn et al. [2012] step away from Langevin Dynamics and propose a method based on Fisher scoring. The method generates samples from a Gaussian approximation of the posterior distribution. Chen et al. [2014] propose instead the Stochastic Gradient Hamiltonian Monte Carle (SGHMC) algorithm. The algorithm builds on a variant of second-order Langevin Dynamics, with momentum, to update the state. Similar to SGLD, they inject additional noise into the system, but they also reduce the effect of the gradient noise.

For the acceptance test, Seita et al. [2018] proposed an efficient mini-batch acceptance test. The test replaces the Metropolis-Hasting's test [Hastings, 1970; Metropolis et al., 1953] with a mini-batch version of the Barker test. They correct for the noise from the mini-batch sampling with a correction random variable.

For the SGLD and SGHMC algorithms to approximate or converge to the posterior distribution, the scale of the injected noise needs to dominate the gradient noise in each direction [Chen et al., 2014; Welling and Teh, 2011]. The scale of the injected noise is thus lower bounded by the largest eigenvalue of the covariance matrix of the gradient noise (after correcting for any pre-conditioning). The injection of additional noise reduces the rate of the diffusion of the samplers, since the increased noise increases the random walk behavior. While the gradients may be conditioned with a pre-conditioning matrix, it is not practical to apply a dense pre-conditioning matrix for models with a large number of parameters.

Commonly-used, diagonal pre-conditioning matrices are helpful for scaling the variance in each dimension, but do not help with correlated features.

We propose a novel SGD method based on second-order Langevin dynamics that does not require the injection of additional noise into the system. Going back to the origin of Langevin dynamics, we rely on a form of collisions between simultaneously trained particles to redistribute the noise across all degrees of freedom. We show experimentally that this whitens the noise inserted at each step and results in faster convergence to the posterior distribution.

The remainder of this manuscript is structured as follows: In section 2, we introduce notation. In section 3, we review stochastic gradient descent, Langevin Dynamics, and their relation. In section 4, we present our proposed approach for stochastic gradient descent with collisions. In section 5, we present an experimental study between three versions of stochastic gradient descent with injected noise and/or collisions. Finally, we conclude the manuscript in section 6.

2 Preliminaries

We are given a dataset X consisting of n observations, $\{x_i\}_{i=1}^n$, drawn independently from a distribution. For a vector of model parameters θ, let $p(\theta)$ denote its prior distribution, and let $p(x|\theta)$ denote the probability of observing data x given model parameters θ. By Bayes rule, the posterior distribution of θ given a dataset X of independent observations is $p(\theta|X) \propto \prod_{i=1}^n p(x_i|\theta)p(\theta)$.

3 Stochastic Gradient Descent & Langevin Dynamics

3.1 Stochastic Gradient Descent

Stochastic gradient descent (SGD) [Robbins and Monro, 1951] is an optimization method for finding a local minimum of a function of a function of the form: $\sum_{i=1}^n f(\theta|x_i)$. At each iteration t, we observe a mini-batch $X_t = \{x_1^t, x_2^t, \ldots, x_b^t\}$, a randomly drawn subsample of observations drawn from X. b is known as the batch-size. The model parameters are updated as follows:

$$\theta(t) = \theta(t-1) - \lambda\frac{n}{b}\sum_{i=1}^{b} \nabla f(\theta|x_i^t), \tag{1}$$

where $\lambda > 0$ is the learning rate. A common extension for stochastic gradient descent is to add a momentum or velocity term $v(\theta)$. The update rule for the model parameters is then:

$$v(t) = \mu v(t-1) - \lambda\frac{n}{b}\sum_{i=1}^{b} \nabla f(\theta|x_i^t), \tag{2}$$

$$\theta(t) = \theta(t-1) + v(t). \tag{3}$$

Here $\mu \in [0, 1)$ can be interpreted as the coefficient of a friction force acting on the particle.

An alternative way to interpret the SGD update is to consider $\nabla \hat{f}_{X_t}(\theta) = \frac{n}{b} \sum_{i=1}^{b} \nabla f(\theta | x_i^t)$ as a sample estimator of $\nabla f_X(\theta) = \sum_{i=1}^{n} \nabla f(\theta | x_i)$. We note that $\mathbb{E}_{X_t}\left[\nabla \hat{f}_{X_t}(\theta)\right] = \nabla f_X(\theta)$. The estimator's covariance matrix is:

$$Q(\theta) = \mathbb{E}_{X_t, X_{t'}}\left[\left(\nabla \hat{f}_{X_t}(\theta) - \nabla f_X(\theta)\right)\left(\nabla \hat{f}_{X_{t'}}(\theta) - \nabla f_X(\theta)\right)^T\right]. \tag{4}$$

By the central limit theorem, we may assume for a sufficiently large batch size b that:

$$\nabla \hat{f}_{X_t}(\theta) = \nabla f_X(\theta) + N\left(0, Q(\theta)\right). \tag{5}$$

A generalization of stochastic gradient descent includes the injection additional Gaussian noise into the system. The generalized update rules are then:

$$v(t) = \mu v(t-1) - \lambda(\nabla f_X(\theta) + \epsilon N(0, V(\theta)), \tag{6}$$

$$\theta(t) = \theta(t-1) + v(t). \tag{7}$$

Here, the covariance matrix $V(\theta)$ captures both the SGD noise and any noise injected into the system, and the parameter ϵ is a scaling parameter for the noise added to the system.

3.2 Langevin Dynamics

Langevin dynamics [Langevin, 1908] describes the motion of a heavy particle emerged in a fluid that is subject to micro-collisions with molecules in the fluid (Brownian motion). The model accounts for a potential, random motion due to collisions, and viscous damping due to the fluid [Pavliotis, 2014]:

$$M\ddot{x} = -\nabla U(x) - \gamma \dot{x} + \sqrt{2\gamma k_B T} W(t). \tag{8}$$

x represents the position of the particle, $\dot{x}$ is its velocity, and $\ddot{x}$ is its acceleration. The single and double dotted x refer to the first and second time derivative respectively. M is the mass of the particle, $U(x)$ is the potential function, $\gamma \geq 0$ is the damping coefficient, k_B is the Boltzmann constant, and T is the temperature. $W(t)$ is a delta-correlated, stationary Gaussian process with zero mean. A delta-correlated stochastic process $Z(t)$ has an auto-covariance function of $\mathrm{cov}(Z(t), Z(t')) = I\delta(t - t')$, where $\delta(x)$ is the Dirac delta function and I is the identity matrix.

The stationary distribution $\pi(x)$ for a particle observing Langevin dynamics is the well-known Boltzmann distribution:

$$\pi(x) \propto \exp\left(-\frac{U(x)}{k_B T}\right). \tag{9}$$

By choosing

$$U(\theta) = -\sum_{i=1}^{n} \log p(x_i | \theta) - \log p(\theta) \tag{10}$$

and setting a temperature of $T = \frac{1}{k_B}$, the stationary distribution of a Langevin particle is equal to the posterior distribution $p(\theta|X)$. By simulating a Langevin particle, we can sample from the posterior distribution of the parameters. How fast the particle moves through the stationary distribution is determined by its diffusion coefficient:

$$D = \frac{T}{\gamma} \tag{11}$$

Within the machine learning literature, this stochastic process is commonly known as second-order Langevin dynamics. A special case is first-order Langevin dynamics, or over-damped Langevin dynamics, where the term $M\ddot{x}$ disappears:

$$0 = -\nabla U(x) - \gamma\dot{x} + \sqrt{2\gamma k_B T} W(t). \tag{12}$$

3.3 The Connection between Langevin Dynamics and Stochastic Gradient Descent

Stochastic gradient descent is closely related to a discretized version of Langevin dynamics. To make this connection apparent, we discretize the second-order Langevin process with unit stepsize:

$$\ddot{x}(t) = v(t) - v(t-1), \quad \dot{x}(t) = v(t-1), \quad x(t) = x(t-1) + v(t). \tag{13}$$

Here, $x(t)$ denotes the position at time t, and $v(t)$ denotes the velocity a time t. Furthermore, we discretized our Gaussian noise process and replace $W(t)$ with $N(0, I)$. After rearranging and dividing the update rule for $v(t)$ by M, we obtain the following update rules:

$$v(t) = \frac{M-\gamma}{M} v(t-1) - \frac{1}{M}\nabla U(x) + \frac{1}{M}\sqrt{2\gamma k_B T} N(0, I), \tag{14}$$

$$x(t) = x(t-1) + v(t). \tag{15}$$

These update rules are similar to those of stochastic gradient descent with momentum given in (6) and (7) after changing from physical parameters (M, γ, T) to SGD parameters (μ, λ, ϵ):

$$M = \frac{1}{\lambda}, \quad \gamma = \frac{(1-\mu)}{\lambda}, \quad T = \frac{\lambda\epsilon^2}{2k_B(1-\mu)} \tag{16}$$

For the update rules to be the same, we require that $V(\theta)$ is the identity matrix. If this were the case, then we could use an appropriately tuned stochastic gradient descent for posterior inference. However, typically $V(\theta)$ is highly-skewed with low rank [Chaudhari and Soatto, 2018].

4 SGD with Collisions

To converge to the posterior distribution and explore it more efficiently, a higher diffusion coefficient D for a given temperature is preferable. This requires a decrease in ϵ^2 proportional to a decrease in γ to maintain the same temperature. One approach to decrease ϵ is to increase the batch size, reducing the sample efficiency and increasing the work per iteration.

Our proposal is to avoid or reduce the addition of noise by randomly perturbing the momentum or velocity through collisions between pairs of particles. Given m particles, we train each particle on an independent copy of the dataset. Every t_{col} iterations, we divide the m particles into $\frac{m}{2}$ random pairs. We simulate a randomized collision between each pair of particle, resulting in an impulse vector and a corresponding velocity change for each particle. Note that we collide pairs of particles regardless of their position and energy.

The collisions should satisfy certain properties to ensure that they only perturb the momentum direction, and do not change the equilibrium of the system. The first condition is that the collisions should satisfy the law of momentum conservation. This guarantees that no energy is added or removed from the system. We also require that the energy per particle is conserved. This ensures that we can collide particles with different energy levels at a distance. Without the requirement that the energy per particle is conserved, the collisions would distort the energy distribution of the system.

Next, we describe the method for colliding the particles. Suppose we have two particles with momentum vectors p and q. Recall from physics that the momentum of an object is defined as its velocity times its mass. The momentum conversation law requires that the total momentum of the system remains constant. The collision impulse for the two particles must thus be c and $-c$ for some impulse vector c. Our second condition requires that the kinetic energy per particle is preserved. Our momentum vectors after collision $p+c$ and $q-c$ must thus satisfy:

$$\|p+c\|_2^2 = \|p\|_2^2 \quad \text{and} \quad \|q-c\|_2^2 = \|q\|_2^2. \tag{17}$$

One impulse vector in $\text{span}(p, q)$ that satisfies these constraints is the vector s in Figure 1. Note however that this solution does not achieve our goal of distributing the momentum across the degrees of freedom. We therefore combine the vector s with a random vector r not in $\text{span}(p, q)$. r is initially drawn from a standard multivariate normal distribution. Subsequently, we compute $\bar{r}$ as the part of r that is orthogonal to both p and q. This requires solving a linear system in two variables. $\bar{r}$ is then rescaled such that $\|\bar{r}\|_2 = \|s\|_2$. Finally, we define $c = \frac{1}{2}(s + \bar{r})$. This choice of c ensures that half of the energy in the collision impulse is distributed into a random degree of freedom. Note that this definition of c still satisfies the energy conservation requirement.

5 Experimental Results

We compare three variants of SGD for the problem of estimating the posterior distribution of the mean parameter of a multivariate Gaussian model with a known, highly-skewed, and correlated covariance matrix.

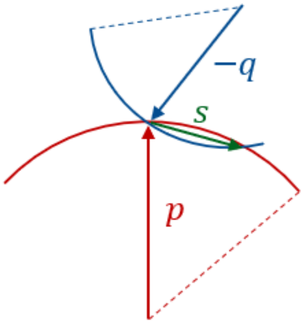


Figure 1: Collision impulse vector in span(p, q).

5.1 Experimental Setup

We consider three different samplers: SGD with momentum (SGD), SGD with momentum and added noise (SGD + Noise), and SGD with momentum and collisions (SGD + Collisions). The SGD + Noise method is inspired by the SGLD and SGHMC algorithms [Chen et al., 2015; Welling and Teh, 2011], but it does not match the methods as described in their respective papers exactly.

We evaluate the samplers on a 10-dimensional, multivariate normal distribution with a highly-skewed, and correlated covariance matrix. The dataset consists of 4800 observations from a 10-dimensional, multivariate normal distribution with zero mean and covariance matrix $\Sigma = RDR^T$. R is an orthonormal rotation matrix drawn from the Haar distribution. D is a diagonal matrix whose i^{th} entry is $2^{-(i-1)}$ for $i = 1, \ldots, 10$. The covariance matrix is thus highly skewed with exponentially decaying eigenvalues. The rotation matrix R ensures that the features are correlated.

The goal is to obtain the posterior distribution for the mean parameter θ. We assume that the covariance matrix is known, but it cannot be incorporated in the algorithm. As prior, we use an improper prior with density function $p(\theta) = 1$. It can be shown analytically that the posterior distribution for θ is a multivariate normal distribution with mean equal to the sample mean of the data and covariance matrix equal to $\frac{\Sigma}{4800}$.

For each sampler, we train 1000 particles for 1250 epochs with a batch size of 32. The parameters for each of the algorithms are listed in Table 1. The learning rates for each algorithm are selected such that the temperature T is $\frac{1}{k_B}$, and the stationary distribution of the particle matches the posterior distribution. For SGD + Noise, the variance of the added noise was selected to be the same order of magnitude as the largest eigenvalue of $Q(\theta)$.

5.2 Experimental Results

Figure 2 provides the covariance matrix of the particles for the last iteration of each of the sampling methods. We observe that both SGD + Collisions and SGD + Noise have converged close to the posterior distribution (see Figure 2d). The velocity and momentum of

Table 1: List of algorithm-specific hyperparameters.

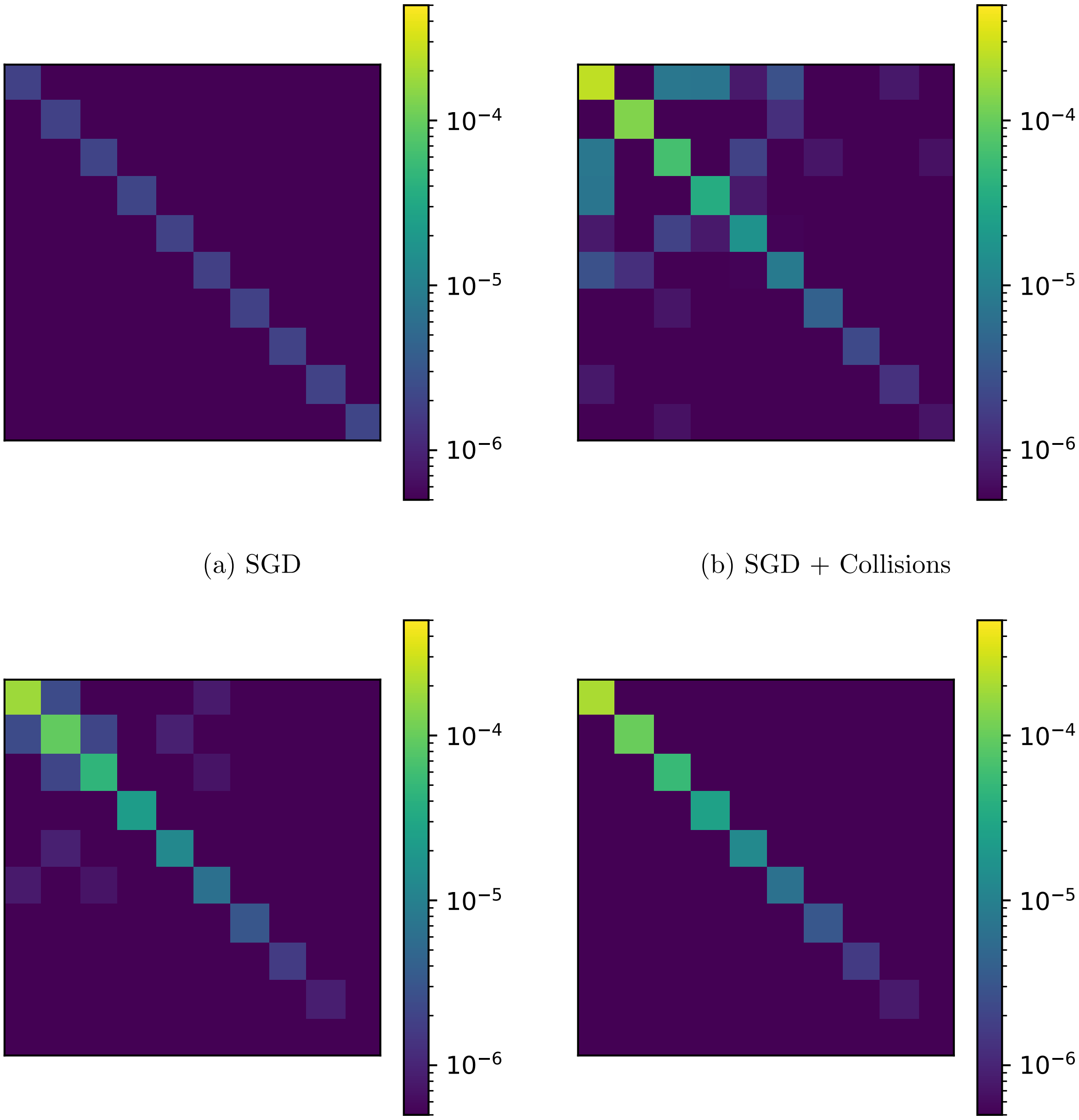
Hyperparameter	SGD	SGD + Collisions	SGD + Noise
Learning rate (λ)	1.39×10^{-9}	1.39×10^{-9}	9.09×10^{-9}
Momentum (μ)	0.95	0.95	0.95
Variance added noise (σ_A^2)	0	0	10^9
Collisions	No	Yes	No
Collision interval (# steps)	N/A	1	N/A

these particles for these sampling methods is distributed almost uniformly across all degrees of freedom as seen in Figure 3. This is expected since the equipartition theorem states that the average energy in each degree of freedom of a system in equilibrium is the same. We do not observe the equipartition of momentum for SGD. Both SGD + Collisions and SGD + Noise final distributions have converged close to the posterior distribution, since both algorithms, despite their different methods, successfully whiten the noise ingested into the system (see Figure 4).

To evaluate how quickly the distribution for each of the sampler converges to the posterior distribution, we provide a variational analysis of the KL divergence between the posterior distribution and the particle distributions for each of the samplers. In this analysis, we assume that the particle distribution for each of the samplers are multivariate Gaussian distributions parametrized by their sample mean vector and their sample covariance matrix. From Figure 5, we observe that SGD + Collisions converges faster than SGD + Noise. In less than 200 epochs, SGD + Collisions stabilizes at the same KL divergence that SGD + Noise reaches after 1250 epochs.The improved convergence for SGD + Collisions is expected based on the higher levels of noise injected into the systems for SGD + Noise, resulting in a lower diffusion coefficient.

We also observe faster convergence for SGD + Collisions when evaluating the error between the average position of the particles at each epoch and the true mean of the posterior distribution (see Figure 6). After approximately 350 epochs, SGD + Collisions has converged to its equilibrium. After all 1250 epochs, the error for SGD + Noise is at least one order of magnitude larger. The error, after convergence, for SGD itself is about an order magnitude lower than that of SGD + Collisions. Although the average particle position is a better estimate of the mean of the posterior distribution, the distribution is not shaped correctly. The distribution is a ball instead o a highly-skewed ellipsoid.

That SGD + Noise explores the distribution more slowly is also evident along the direction with highest posterior variance, the first dimension of eigenspace. For this dimension in particular, we observe in Figure 7 that the SGD + Noise has higher autocorrelation. This indicates that the sampler traverses this dimension more slowly than SGD + Collisions.



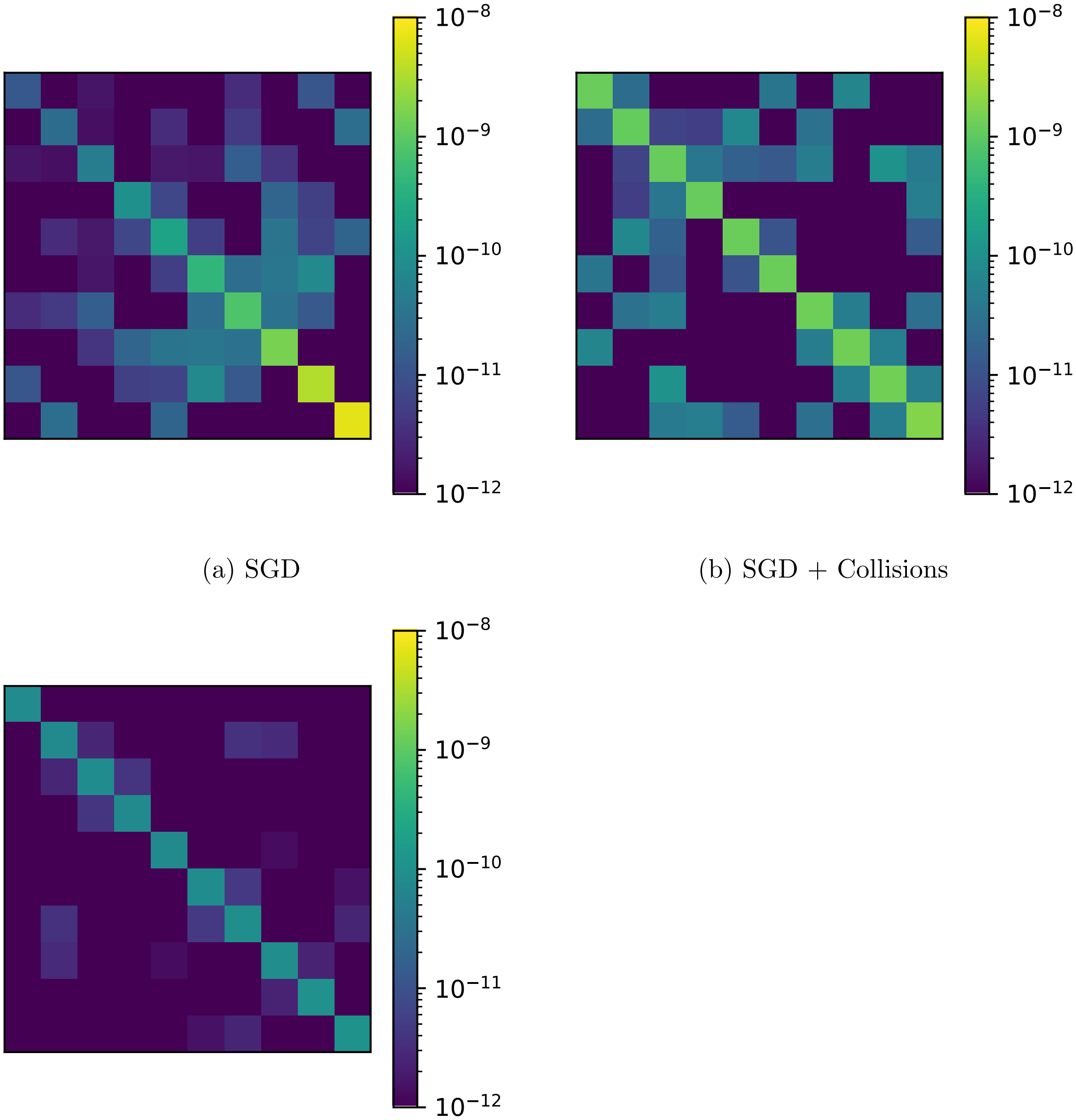
(a) SGD

(b) SGD + Collisions

(c) SGD + Noise

(d) Posterior Distribution

Figure 2: Covariance matrix of particle positions after 1250 epochs. The covariance matrix is projected onto the eigenspace of Σ.

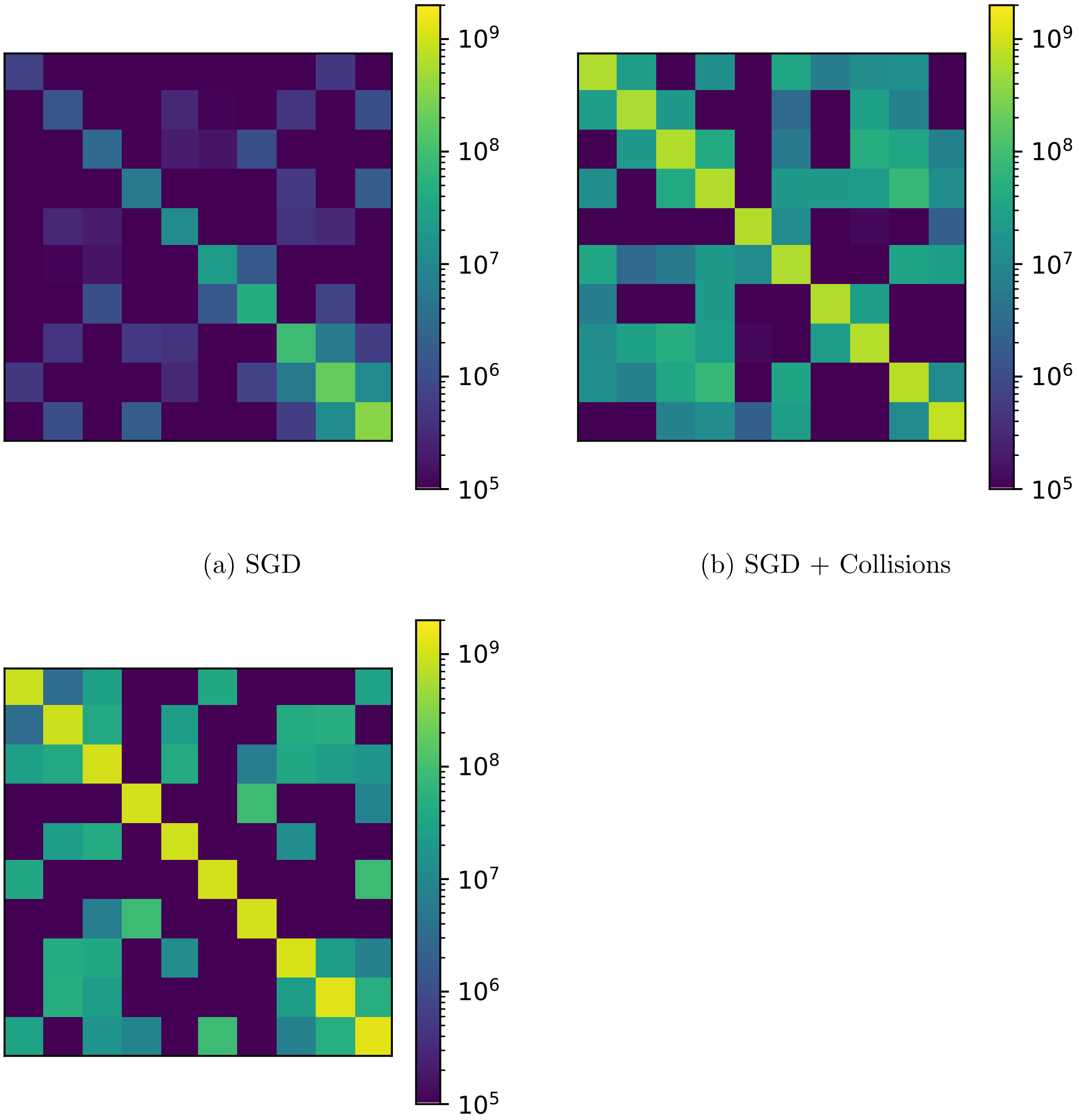


(a) SGD

(b) SGD + Collisions

(c) SGD + Noise

Figure 3: Covariance matrix of particle velocities after 1250 epochs. The covariance matrix is projected onto the eigenspace of Σ.



(a) SGD

(b) SGD + Collisions

(c) SGD + Noise

Figure 4: Covariance matrix of perturbations due to noise and collisions after 1250 epochs. The covariance matrix is projected onto the eigenspace of Σ.

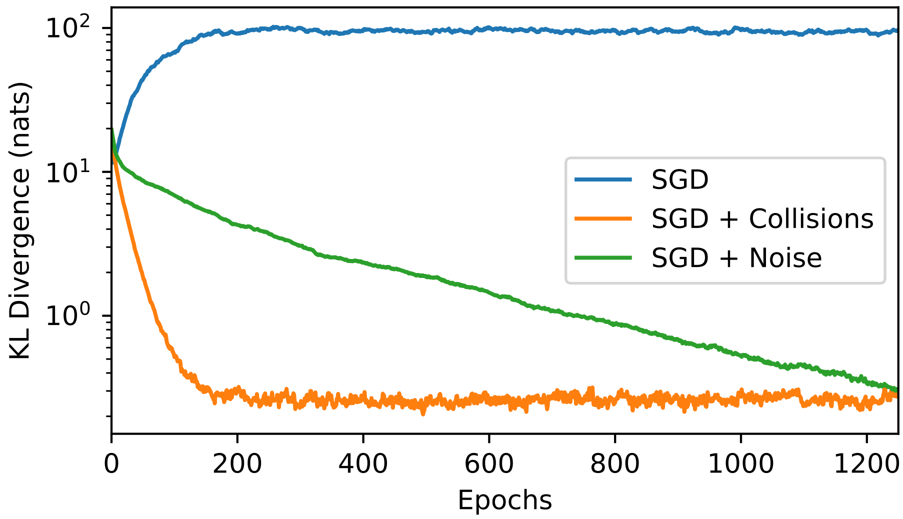


Figure 5: Variational KL divergence.

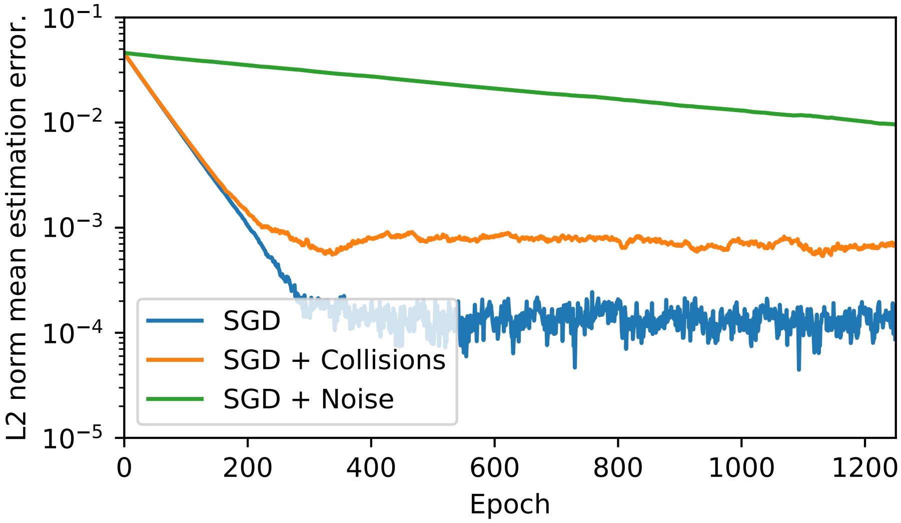
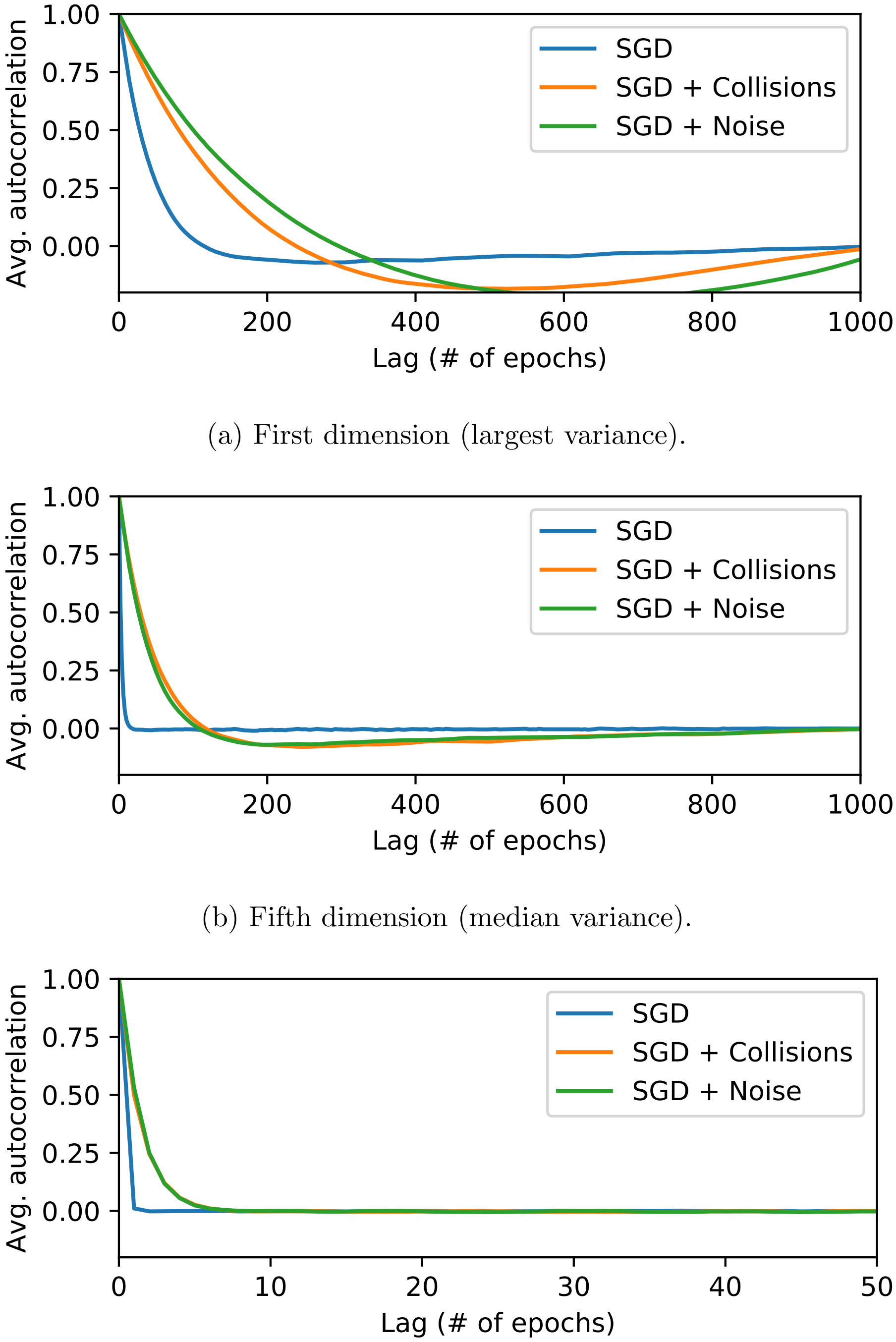


Figure 6: L2 norm of the difference of the average particle position at the end of each epoch and the true mean of the posterior distribution.



(a) First dimension (largest variance).

(b) Fifth dimension (median variance).

(c) Tenth dimension (smallest variance).

Figure 7: Autocorrelation by samplers for up to 1000 lags. Autocorrelations are measured with respect to the basis of the true covariance matrix Σ.

6 Conclusion

We propose to combine stochastic gradient descent with collisions for sampling from posterior distributions. We describe why this method should result in faster diffusion and thus improved exploration of the posterior distribution as compared to SGD-based sampling methods that inject additional noise into the system. We also present initial experimental evidence that the proposed method converges to the posterior distribution faster, but further experimental analysis is required to verify this.

Acknowledgments

I would like to thank Jongho (Jerry) Park for his contributions to this project.

References

S. Ahn, A. Korattikara, and M. Welling. Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1591–1598, 2012.

P. Chaudhari and S. Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2018.

C. Chen, N. Ding, and L. Carin. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pages 2278–2286, 2015.

T. Chen, E. Fox, and C. Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. 1970.

P. Langevin. Sur la théorie du mouvement brownien. *Compt. Rendus*, 146:530–533, 1908.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6): 1087–1092, 1953.

G. A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, volume 60. Springer, 2014.

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

D. Seita, X. Pan, H. Chen, and J. Canny. An Efficient Minibatch Acceptance Test for Metropolis-Hastings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, volume 18, pages 5359–5363, 2018.

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.