Stochastic Variance-Reduced Hamilton Monte Carlo Methods

Difan Zou [* 1] Pan Xu [* 1] Quanquan Gu [1]

Abstract

We propose a fast stochastic Hamilton Monte Carlo (HMC) method, for sampling from a smooth and strongly log-concave distribution. At the core of our proposed method is a variance reduction technique inspired by the recent advance in stochastic optimization. We show that, to achieve ϵ accuracy in 2-Wasserstein distance, our algorithm achieves $\widetilde{O}\big(n + \kappa^2 d^{1/2}/\epsilon + \kappa^{4/3}d^{1/3}n^{2/3}/\epsilon^{2/3}\big)$ gradient complexity (i.e., number of component gradient evaluations), which outperforms the state-of-the-art HMC and stochastic gradient HMC methods in a wide regime. We also extend our algorithm for sampling from smooth and general log-concave distributions, and prove the corresponding gradient complexity as well. Experiments on both synthetic and real data demonstrate the superior performance of our algorithm.

1. Introduction

Past decades have witnessed increasing attention of Markov Chain Monte Carlo (MCMC) methods in modern machine learning problems (Andrieu et al., 2003). An important family of Markov Chain Monte Carlo algorithms, called Langevin Monte Carlo method (Neal et al., 2011), is proposed based on Langevin dynamics (Parisi, 1981). Langevin dynamics was used for modeling of the dynamics of molecular systems, and can be described by the following Itô's stochastic differential equation (SDE) (Øksendal, 2003),

$$dX_t = -\nabla f(X_t)dt + \sqrt{2\beta}dB_t, \qquad (1.1)$$

where X_t is a d-dimensional stochastic process, $t \geq 0$ denotes the time index, $\beta > 0$ is the temperature parameter, and B_t is the standard d-dimensional Brownian motion. Under certain assumptions on the drift coefficient ∇f, Chiang et al. (1987) showed that the distribution of X_t in (1.1)

*Equal contribution [1]Department of Computer Science, University of California, Los Angeles, CA 90095, USA. Correspondence to: Quanquan Gu <qgu@cs.ucla.edu>.

converges to its stationary distribution, a.k.a., the Gibbs measure $\pi_\beta \propto \exp(-\beta f(\mathbf{x}))$. Note that π_β is smooth and log-concave (resp. strongly log-concave) if f is smooth and convex (resp. strongly convex). A typical way to sample from density π_β is applying Euler-Maruyama discretization scheme (Kloeden & Platen, 1992) to (1.1), which yields

$$X_{k+1} = X_k - \nabla f(X_k)\eta + \sqrt{2\eta\beta} \cdot \epsilon_k, \qquad (1.2)$$

where $\epsilon_k \sim N(\mathbf{0}, \mathbf{I}_{d \times d})$ is a standard Gaussian random vector, $\mathbf{I}_{d \times d}$ is a $d \times d$ identity matrix, and $\eta > 0$ is the step size. (1.2) is often referred to as the Langevin Monte Carlo (LMC) method. In total variation (TV) distance, LMC has been proved to be able to produce approximate sampling of density $\pi_\beta \propto e^{-f/\beta}$ under arbitrary precision requirement in Dalalyan (2014); Durmus & Moulines (2016b), with properly chosen step size. The non-asymptotic convergence of LMC has also been studied in Dalalyan (2017); Dalalyan & Karagulyan (2017); Durmus et al. (2017), which shows that the LMC algorithm can achieve ϵ-precision in 2-Wasserstein distance after $\widetilde{O}(\kappa^2 d/\epsilon^2)$ iterations if f is L-smooth and μ-strongly convex, where $\kappa = L/\mu$ is the condition number.

In order to accelerate the convergence of Langevin dynamics (1.1) and improve its mixing time to the unique stationary distribution, Hamiltonian dynamics (Duane et al., 1987; Neal et al., 2011) was proposed, which is also known as underdamped Langevin dynamics and is defined by the following system of SDEs

$$\begin{aligned} d\mathbf{V}_t &= -\gamma \mathbf{V}_t dt - u\nabla f(X_t)dt + \sqrt{2\gamma u}d\mathbf{B}_t, \\ dX_t &= \mathbf{V}_t dt, \end{aligned} \qquad (1.3)$$

where $\gamma > 0$ is the friction parameter, u denotes the inverse mass, $X_t, \mathbf{V}_t \in \mathbb{R}^d$ are the position and velocity of the continuous-time dynamics respectively, and $\mathbf{B}_t$ is the Brownian motion. Let $\mathbf{W}_t = (X_t^\top, \mathbf{V}_t^\top)^\top$, under mild assumptions on the drift coefficient $\nabla f(\mathbf{x})$, the distribution of $\mathbf{W}_t$ converges to an unique invariant distribution $\pi_{\mathbf{w}} \propto e^{-f(\mathbf{x})-\|\mathbf{v}\|_2^2/(2u)}$ (Neal et al., 2011), whose marginal distribution on X_t, denoted by π, is proportional to $e^{-f(\mathbf{x})}$. Similar to the numerical approximation of the Langevin dynamics in (1.2), one can also apply the same Euler-Maruyama discretization scheme to Hamiltonian dynamics in (1.3), which gives rise to Hamiltonian Monte

Carlo (HMC) method

$$\mathbf{v}_{k+1} = \mathbf{v}_k - \gamma\eta\mathbf{v}_k - \eta u\nabla f(\mathbf{x}_k) + \sqrt{2\gamma u\eta}\boldsymbol{\epsilon}_k,$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta\mathbf{v}_k. \tag{1.4}$$

(1.4) provides an alternative way to sample from the target distribution $\pi \propto e^{-f(\mathbf{x})}$. While HMC has been observed to outperform LMC in a number of empirical studies (Chen et al., 2014; 2015), there does not exist a non-asymptotic convergence analysis of the HMC method until very recent work by Cheng et al. (2017)[1]. In particular, Cheng et al. (2017) proposed a variant of HMC based on coupling techniques, and showed that it achieves ϵ sampling accuracy in 2-Wasserstein distance within $\widetilde{O}(\kappa^2 d^{1/2}/\epsilon)$ iterations for smooth and strongly convex function f. This improves upon the convergence rate of LMC by a factor of $\widetilde{O}(d^{1/2}/\epsilon)$.

Both LMC and HMC are gradient based Monte Carlo methods and are effective in sampling from smooth and strongly log-concave distributions. However, they can be slow if the evaluation of the gradient is computationally expensive, especially on large datasets. This motivates using stochastic gradient instead of full gradient in LMC and HMC, which gives rise to Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011; Ahn et al., 2012; Durmus & Moulines, 2016b; Dalalyan, 2017) and Stochastic Gradient Hamilton Monte Carlo (SG-HMC) method (Chen et al., 2014; Ma et al., 2015; Chen et al., 2015) respectively. For smooth and strongly log-concave distributions, Dalalyan & Karagulyan (2017); Dalalyan (2017) proved that the convergence rate of SGLD is $\widetilde{O}(\kappa^2 d\sigma^2/\epsilon^2)$, where σ^2 denotes the upper bound of the variance of the stochastic gradient. Cheng et al. (2017) proposed a variant of SG-HMC and proved that it converges after $\widetilde{O}(\kappa^2 d\sigma^2/\epsilon^2)$ iterations. It is worth noting that although using stochastic gradient evaluations reduces the per-iteration cost, it comes at a cost that the convergence rates of SGLD and SG-HMC are slower than LMC and HMC. Thus, a natural questions is:

Does there exist an algorithm that can leverage stochastic gradients, but also achieve a faster rate of convergence?

In this paper, we answer this question affirmatively, when the function f can be written as the finite sum of n smooth component functions f_i

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}). \tag{1.5}$$

It is worth noting that the finite sum structure is prevalent in machine learning, as the log-likelihood function of a dataset (e.g., f) is the sum of the log-likelihood over each data point (e.g., f_i) in the dataset. We propose a stochastic

---

[1] In Cheng et al. (2017), the sampling method in (1.4) is also called the underdamped Langevin MCMC algorithm.

variance-reduced HMC (SVR-HMC), which incorporates the variance reduction technique into stochastic HMC. Our algorithm is inspired by the recent advance in stochastic optimization (Roux et al., 2012; Johnson & Zhang, 2013; Xiao & Zhang, 2014; Defazio et al., 2014; Allen-Zhu & Hazan, 2016; Reddi et al., 2016; Lei & Jordan, 2016; Lei et al., 2017), which use semi-stochastic gradients to accelerate the optimization of the finite-sum function, and to improve the runtime complexity of full gradient methods. We also notice that the variance reduction technique has already been employed in recent work Dubey et al. (2016); Baker et al. (2017) on SGLD. Nevertheless, it does not show an improvement in terms of dependence on the accuracy ϵ.

In detail, the proposed SVR-HMC uses a multi-epoch scheme to reduce the variance of the stochastic gradient. At the beginning of each epoch, it computes the full gradient or an estimation of the full gradient based on the entire data. Within each epoch, it performs semi-stochastic gradient descent and outputs the last iterate as the warm up starting point for the next epoch. Thorough experiments on both synthetic and real data demonstrate the advantage of our proposed algorithm.

Our Contributions The major contributions of our work are highlighted as follows.

- We propose a new algorithm, SVR-HMC, that incorporates variance-reduction technique into HMC. Our algorithm does not require the variance of the stochastic gradient is bounded. We proved that SVR-HMC has a better gradient complexity than the state-of-the-art LMC and HMC methods for sampling from smooth and strongly log-concave distributions, when the error is measured by 2-Wasserstein distance. In particular, to achieve ϵ sampling error in 2-Wasserstein distance, our algorithm only needs $\widetilde{O}(n + \kappa^2 d^{1/2}/\epsilon + \kappa^{4/3}d^{1/3}n^{2/3}/\epsilon^{2/3})$ number of component gradient evaluations. This improves upon the state-of-the-art result by (Cheng et al., 2017), which is $\widetilde{O}(n\kappa^2 d^{1/2}/\epsilon)$ in a large regime.

- We generalize the analysis of SVR-HMC to sampling from smooth and general log-concave distributions by adding a diminishing regularizer. We prove that the gradient complexity of SVR-HMC to achieve ϵ-accuracy in 2-Wasserstein distance is $\widetilde{O}(n + d^{11/2}/\epsilon^6 + d^{11/3}n^{2/3}/\epsilon^4)$. To the best of our knowledge, this is the first convergence result of LMC methods in 2-Wasserstein distance.

Notation We denote the discrete update by lower case symbol $\mathbf{x}_k$ and the continuous-time dynamics by upper case symbol $\boldsymbol{X}_t$. We denote by $\|\mathbf{x}\|_2$ the Euclidean norm of vector $\mathbf{x} \in \mathbb{R}^d$. For a random vector $\boldsymbol{X}_t \in \mathbb{R}^d$ (or $\mathbf{x}_k \in \mathbb{R}^d$), we denote its probability distribution function by $P(\boldsymbol{X}_t)$ (or $P(\mathbf{x}_k)$). We denote by $\mathbb{E}_u(\boldsymbol{X})$ the expectation of $\boldsymbol{X}$ under

probability measure u. The squared 2-Wasserstein distance between probability measures u and v is

$$\mathcal{W}_2^2(u,v) = \inf_{\zeta \in \Gamma(u,v)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\boldsymbol{X}_u - \boldsymbol{X}_v\|_2^2 \mathrm{d}\zeta(\boldsymbol{X}_u, \boldsymbol{X}_v),$$

where $\Gamma(u,v)$ is the set of all joint distributions with u and v being the marginal distributions. We use $a_n = O(b_n)$ to denote $a_n \leq C b_n$ for some constant $C > 0$ independent of n, and use $a_n = \widetilde{O}(b_n)$ to hide logarithmic terms of b_n. We denote $a_n \lesssim b_n$ ($a_n \gtrsim b_n$) if a_n is less than (larger than) b_n up to a constant. We use $a \wedge b$ to denote $\min\{a, b\}$

2. Related Work

In this section, we briefly review the relevant work in the literature.

Langevin Monte Carlos (LMC) methods (a.k.a, Unadjusted Langevin Algorithms), and its Metropolis adjusted version, have been studied in a number of papers (Roberts & Tweedie, 1996; Roberts & Rosenthal, 1998; Stramer & Tweedie, 1999a;b; Jarner & Hansen, 2000; Roberts & Stramer, 2002), which have been proved to attain asymptotic exponential convergence. In the past few years, there has emerged numerous studies on proving the non-asymptotic convergence of LMC methods. Dalalyan (2014) first proposed the theoretical guarantee for approximate sampling using Langevin Monte Carlo method for strongly log-concave and smooth distributions, where he proved rate $O(d/\epsilon^2)$ for LMC algorithm with warm start in total variation (TV) distance. This result has later been extended to Wasserstein metric by Dalalyan & Karagulyan (2017); Durmus & Moulines (2016b), where the same convergence rate in 2-Wasserstein distance holds without the warm start assumption. Recently, Cheng & Bartlett (2017) also proved an $\widetilde{O}(d/\epsilon)$ convergence rate of the LMC algorithm in KL-divergence. The stochastic gradient based LMC methods, also known as stochastic gradient Langevin dynamics (SGLD), was originally proposed for Bayesian posterior sampling (Welling & Teh, 2011; Ahn et al., 2012). Dalalyan (2017); Dalalyan & Karagulyan (2017) analyzed the convergence rate for SGLD based on both unbiased and biased stochastic gradients. In particular, they proved that the gradient complexity for unbiased SGLD is $O(\kappa^2 d/\epsilon^2)$, and showed that it may not converge to the target distribution if the stochastic gradient has non-negligible bias. The SGLD algorithm has also been applied to nonconvex optimization. Raginsky et al. (2017) analyzed the non-asymptotic convergence rate of SGLD. Zhang et al. (2017) provided the theoretical guarantee of SGLD in terms of the hitting time to a first and second-order stationary point. Xu et al. (2017) provided a analysis framework for the global convergence of LMC, SGLD and its variance-reduced variant based on the ergodicity of the discrete-time algorithm.

In order to improve convergence rates of LMC methods, the Hamiltonian Monte Carlo (HMC) method was proposed Duane et al. (1987); Neal et al. (2011), which introduces a momentum term in its dynamics. To deal with large datasets, stochastic gradient HMC has been proposed for Bayesian learning (Chen et al., 2014; Ma et al., 2015). Chen et al. (2015) investigated the generic stochastic gradient MCMC algorithms with high-order integrators, and provided a comprehensive convergence analysis. For strongly log-concave and smooth distribution, a non-asymptotic convergence guarantee was proved by Cheng et al. (2017) for underdamped Langevin MCMC, which is a variant of stochastic gradient HMC method.

Our proposed algorithm is motivated by the stochastic variance reduced gradient (SVRG) algorithm, was first proposed in Johnson & Zhang (2013), and later extended to different problem setups Xiao & Zhang (2014); Defazio et al. (2014); Reddi et al. (2016); Allen-Zhu & Hazan (2016); Lei & Jordan (2016); Lei et al. (2017). Inspired by this line of research, Dubey et al. (2016) applied the variance reduction technique to stochastic gradient Langevin dynamics, and proved a slightly tighter convergence bound than SGLD. Nevertheless, the dependence of the convergence rate on the sampling accuracy ϵ is not improved. Thus, it remains open whether variance reduction technique can indeed improve the convergence rate of MCMC methods. Our work answers this question in the affirmative and provides rigorously faster rates of convergence for sampling from log-concave and smooth density functions.

For the ease of comparison, we summarize the gradient complexity[2] in 2-Wasserstein distance for different gradient-based Monte Carlo methods in Table 1. Evidently, for sampling from smooth and strongly log-concave distributions, SVR-HMC outperforms all existing algorithms.

Table 1. Gradient complexity of gradient-based Monte Carlo algorithms in 2-Wasserstein distance for sampling from smooth and strong log-concave distributions.

Methods	Gradient Complexity
LMC (Dalalyan, 2017)	$\widetilde{O}\left(\frac{n\kappa^2 d}{\epsilon^2}\right)$
HMC (Cheng et al., 2017)	$\widetilde{O}\left(\frac{n\kappa^2 d^{1/2}}{\epsilon}\right)$
SGLD (Dalalyan, 2017)	$\widetilde{O}\left(\frac{\kappa^2 \sigma^2 d}{\epsilon^2}\right)$
SG-HMC (Cheng et al., 2017)	$\widetilde{O}\left(\frac{\kappa^2 \sigma^2 d}{\epsilon^2}\right)$
SVR-HMC (this paper)	$\widetilde{O}\left(n + \frac{\kappa^2 d^{1/2}}{\epsilon} + \frac{\kappa^{3/4} d^{1/3} n^{2/3}}{\epsilon^{2/3}}\right)$

---

[2]The gradient complexity is defined as number of stochastic gradient evaluations to achieve ϵ sampling accuracy.

3. The Proposed Algorithm

In this section, we propose a novel HMC algorithm that leverages variance reduced stochastic gradient to sample from the target distribution $\pi = e^{-f(\mathbf{x})}/Z$, where $Z = \int e^{-f(\mathbf{x})} \mathrm{d}\mathbf{x}$ is the partition function.

Recall that function $f(\mathbf{x})$ has the finite-sum structure in (1.5). When n is large, the full gradient $1/n \sum_{i=1}^{n} \nabla f_i(\mathbf{x})$ in (1.4) can be expensive to compute. Thus, the stochastic gradient is often used to improve the computational complexity per iteration. However, due to the non-diminishing variance of the stochastic gradient, the convergence rate of gradient-based MC methods using stochastic gradient is often no better than that of gradient MC using full gradient.

In order to overcome the drawback of stochastic gradient, and achieve faster rate of convergence, we propose a Stochastic Variance-Reduced Hamiltonian Monte Carlo algorithm (SVR-HMC), which leverages the advantages of both HMC and variance reduction. The outline of the algorithm is displayed in Algorithm 1. We can see that the algorithm performs in a multi-epoch way. At the beginning of each epoch, it computes the full gradient of the f at some snapshot of the iterate $\widetilde{\mathbf{x}}_j$. Then it performs the following update for both the velocity and the position variables in each epoch

$$
\begin{aligned}
\mathbf{v}_{k+1} &= \mathbf{v}_k - \gamma\eta\mathbf{v}_k - \eta u\mathbf{g}_k + \boldsymbol{\epsilon}_k^v, \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \eta\mathbf{v}_k + \boldsymbol{\epsilon}_k^x,
\end{aligned}
\tag{3.1}
$$

where $\gamma, \eta, u > 0$ are tuning parameters, $\mathbf{g}_k$ is a semi-stochastic gradient that is an unbiased estimator of $\nabla f(\mathbf{x}_k)$ and defined as follows,

$$
\mathbf{g}_k = \nabla f_{i_k}(\mathbf{x}_k) - \nabla f_{i_k}(\widetilde{\mathbf{x}}_j) + \nabla f(\widetilde{\mathbf{x}}_j),
\tag{3.2}
$$

where i_k is uniformly sampled from $\{1, \ldots, n\}$, and $\widetilde{\mathbf{x}}_j$ is a snapshot of $\mathbf{x}_k$ that is only updated every m iterations such that $k = jm + l$ for some $l = 0, \ldots, m - 1$. And $\boldsymbol{\epsilon}_k^v$ and $\boldsymbol{\epsilon}_k^x$ are Gaussian random vectors with zero mean and covariance matrices equal to

$$
\mathbb{E}[\boldsymbol{\epsilon}_k^v(\boldsymbol{\epsilon}_k^v)^\top] = u(1 - e^{-2\gamma\eta}) \cdot \mathbf{I}_{d\times d},
$$

$$
\mathbb{E}[\boldsymbol{\epsilon}_k^x(\boldsymbol{\epsilon}_k^x)^\top] = \frac{u}{\gamma^2}(2\gamma\eta + 4e^{-\gamma\eta} - e^{-2\gamma\eta} - 3) \cdot \mathbf{I}_{d\times d},
$$

$$
\mathbb{E}[\boldsymbol{\epsilon}_k^v(\boldsymbol{\epsilon}_k^x)^\top] = \frac{u}{\gamma}(1 - 2e^{-\gamma\eta} + e^{-2\gamma\eta}) \cdot \mathbf{I}_{d\times d},
\tag{3.3}
$$

where $\mathbf{I}_{d\times d}$ is a $d \times d$ identity matrix.

The idea of semi-stochastic gradient has been successfully used in stochastic optimization in machine learning to reduce the variance of stochastic gradient and obtains faster convergence rates (Johnson & Zhang, 2013; Xiao & Zhang, 2014; Reddi et al., 2016; Allen-Zhu & Hazan, 2016; Lei & Jordan, 2016; Lei et al., 2017). Apart from the semi-stochastic gradient, the second update formula in (3.1) also

differs from the direct Euler-Maruyama discretization (1.4) of Hamiltonian dynamics due to the additional Gaussian noise term $\boldsymbol{\epsilon}_k^x$. This additional Gaussian noise term is pivotal in our theoretical analysis to obtain faster convergence rates of our algorithm than LMC methods. Similar idea has been used in Cheng et al. (2017) to prove the faster rate of convergence of HMC (underdamped MCMC) against LMC.

Algorithm 1 Stochastic Variance-Reduced Hamiltonian Monte Carlo (SVR-HMC)

1: **initialization:** $\widetilde{\mathbf{x}}_0 = \mathbf{0}, \widetilde{\mathbf{v}}_0 = \mathbf{0}$
2: **for** $j = 0, \ldots, \lceil K/m \rceil$
3:     $\widetilde{\mathbf{g}} = \nabla f(\widetilde{\mathbf{x}}_j)$
4:     **for** $l = 0, \ldots, m - 1$
5:         $k = jm + l$
6:         Uniformly sample $i_k \in [n]$
7:         $\mathbf{g}_k = \nabla f_{i_k}(\mathbf{x}_k) - \nabla f_{i_k}(\widetilde{\mathbf{x}}_j) + \widetilde{\mathbf{g}}$
8:         $\mathbf{x}_{k+1} = \mathbf{x}_k + \eta\mathbf{v}_k + \boldsymbol{\epsilon}_k^x$
9:         $\mathbf{v}_{k+1} = \mathbf{v}_k - \gamma\eta\mathbf{v}_k - \eta u\mathbf{g}_k + \boldsymbol{\epsilon}_k^v.$
10:        **if** $l = m - 1$
11:           $\widetilde{\mathbf{x}}_j = \mathbf{x}_{k+1}$
12:        **end**
13:     **end for**
14: **end for**
15: **output:** $\mathbf{x}_K$

---

4. Main Theory

In this section, we analyze the convergence of our proposed algorithm in 2-Wasserstein distance between the distribution of the iterate in Algorithm 1, and the target distribution $\pi \propto e^{-f}$.

Following the recent work Durmus & Moulines (2016a); Dalalyan & Karagulyan (2017); Dalalyan (2017); Cheng et al. (2017), we use the 2-Wasserstein distance to measure the convergence rate of Algorithm 1, since it directly provides the level of approximation of the first and second order moments (Dalalyan, 2017; Dalalyan & Karagulyan, 2017). It is arguably more suitable to characterize the quality of approximate sampling algorithms than the other distance metrics such as total variation distance. In addition, while Algorithm 1 performs update on both the position variable $\mathbf{x}_k$ and the velocity variable $\mathbf{v}_k$, only the convergence rate of the position variable $\mathbf{x}_k$ is of central interest.

4.1. SVR-HMC for Sampling from Strongly Log-concave Distributions

We first present the convergence rate and gradient complexity of SVR-HMC when f is smooth and strongly convex, i.e., the target distribution $\pi \propto e^{-f}$ is smooth and strongly log-concave. We start with the following formal assumptions on the negative log density function.

Assumption 4.1 (Smoothness). There exists a constant $L > 0$, such that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the following holds for any i,

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2.$$

Under Assumption 4.1, it can be easily verified that function $f(\mathbf{x})$ is also L-smooth.

Assumption 4.2 (Strong Convexity). There exists a constant $\mu > 0$, such that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the following holds for any i,

$$f(\mathbf{x}) - f(\mathbf{y}) \geq \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|_2^2. \quad (4.1)$$

Note that the strong convexity assumption is only made on the finite sum function f, instead of the individual component function f_i's.

Theorem 4.3. Under Assumptions 4.1 and 4.2. Let $P(\mathbf{x}_K)$ denote the distribution of the last iterate $\mathbf{x}_K$, and $\pi \propto e^{-f(\mathbf{x})}$ denote the stationary distribution of (1.3). Set $u = 1/L$, $\gamma = 2$, $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{v}_0 = \mathbf{0}$ and $\eta = \widetilde{O}(1/\kappa \wedge 1/(\kappa^{1/3}n^{2/3}))$. Assume $\|\mathbf{x}^*\|_2 \leq R$ for some constant $R > 0$, where $\mathbf{x}^* = \arg\min_x f(\mathbf{x})$ is the global minimizer of function $f(\mathbf{x})$. Then the output of Algorithm 1 satisfies,

$$\mathcal{W}_2\big(P(\mathbf{x}_K), \pi\big) \leq e^{-K\eta/(2\kappa)}w_0 + 4\eta\kappa(2\sqrt{D_1} + \sqrt{D_2})$$
$$+ 2\sqrt{\kappa D_3}m\eta^{3/2}, \quad (4.2)$$

where $w_0 = \mathcal{W}_2\big(P(\mathbf{x}_0), \pi\big)$, $\kappa = L/\mu$ is the condition number, η is the step size, and m denotes the epoch (i.e., inner loop) length of Algorithm 1. D_1, D_2 and D_3 are defined as follows,

$$D_1 = \left(\frac{8\eta^2}{5} + \frac{4}{3}\right)U_v + \frac{4}{3L}U_f + \frac{16d\eta}{3L},$$
$$D_2 = 13U_v + \frac{8U_f}{L} + \frac{28d\eta}{L}, \quad D_3 = U_v + 4ud,$$

in which parameters U_v and U_f are in the order of $O(d/\mu)$ and $O(d\kappa)$, respectively.

Remark 4.4. In existing stochastic Langevin Monte Carlo methods (Dalalyan & Karagulyan, 2017; Zhang et al., 2017) and stochastic Hamiltonian Monte Carlo methods (Chen et al., 2014; 2015; Cheng et al., 2017), their convergence analyses require bounded variance of stochastic gradient, i.e., the inequality $\mathbb{E}_i[\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|_2^2] \leq \sigma^2$ holds uniformly for all $\mathbf{x} \in \mathbb{R}^d$. In contrast, our analysis does not need this assumption, which implies that our algorithm is applicable to a larger class of target density functions.

In the following corollary, by providing a specific choice of step size η, and epoch length m, we present the gradient complexity of Algorithm 1 in 2-Wasserstein distance.

Corollary 4.5. Under the same conditions as in Theorem 4.3, let $m = n$ and $\eta = O\big(\epsilon/(\kappa^{-1}d^{-1/2}) \wedge \epsilon^{2/3}/(\kappa^{1/3}d^{1/3}n^{2/3})\big)$. Then the output of Algorithm 1 satisfies $\mathcal{W}_2\big(P(\mathbf{x}_K), \pi\big) \leq \epsilon$ after

$$\widetilde{O}\left(n + \frac{\kappa^2 d^{1/2}}{\epsilon} + \frac{n^{2/3}\kappa^{4/3}d^{1/3}}{\epsilon^{2/3}}\right) \quad (4.3)$$

stochastic gradient evaluations.

Remark 4.6. Recall that the gradient complexity of HMC is $\widetilde{O}(n\kappa^2 d^{1/2}/\epsilon)$ and the gradient complexity of SG-HMC is $\widetilde{O}(\kappa^2 d\sigma^2/\epsilon^2)$, both of which are recently proved in Cheng et al. (2017). It can be seen from Corollary 4.5 that the gradient complexity of our SVR-HMC algorithm has a better dependence on dimension d.

Note that the gradient complexity of SVR-HMC in (4.3) depends on the relationship between sample size n and precision parameter ϵ. To make a thorough comparison with existing algorithms, we discuss our result for SVR-HMC in the following three regimes:

- When $n \lesssim \kappa d^{1/4}/\epsilon^{1/2}$, the gradient complexity of our algorithm is dominated by $\widetilde{O}(\kappa^2 d^{1/2}/\epsilon)$, which is lower than that of the HMC algorithm by a factor of $\widetilde{O}(n)$ and lower than that of the SG-HMC algorithm by a factor of $\widetilde{O}(d^{1/2}/\epsilon)$.

- When $\kappa d^{1/4}/\epsilon^{1/2} \lesssim n \lesssim \kappa d\sigma^3/\epsilon^2$, the gradient complexity of our algorithm is dominated by $\widetilde{O}(n^{2/3}\kappa^{4/3}d^{1/3}/\epsilon^{2/3})$. It improves that of HMC by a factor of $\widetilde{O}(n^{1/3}\kappa^{2/3}d^{1/6}/\epsilon^{4/3})$, and is lower than that of SG-HMC by a factor of $\widetilde{O}(\kappa^{2/3}d^{2/3}\sigma^2 n^{-2/3}/\epsilon^{4/3})$. Plugging in the upper bound of n into (4.3) yields $\widetilde{O}(\kappa^2 d\sigma^2/\epsilon^2)$ gradient complexity, which still matches that of SG-HMC.

- When $n \gtrsim \kappa^4 d/\epsilon^2$, i.e., the sample size is super large, the gradient complexity of our algorithm is dominated by $\widetilde{O}(n)$. It is still lower than that of HMC by a factor of $\widetilde{O}(\kappa^2 d^{1/2}/\epsilon)$. Nonetheless, our algorithm has a higher gradient complexity than SG-HMC due to the extremely large sample size. This suggests that SG-HMC (Cheng et al., 2017) is the most suitable algorithm in this regime.

Moreover, from Corollary 4.5 we know that the optimal learning rate for SVR-HMC is in the order of $O(\epsilon^{2/3}/(\kappa^{1/3}d^{1/3}n^{2/3}))$, while the optimal learning rate for SG-HMC is in the order of $O(\epsilon^2/(\sigma^2 d\kappa))$, which is smaller than the learning rate of SVR-HMC when $n \leq \kappa d\sigma^3/\epsilon^2$ (Dalalyan, 2017). This observation aligns with the consequence of variance reduction in the field of optimization.

4.2. SVR-HMC for Sampling from General Log-concave Distributions

In this section, we will extend the analysis of the proposed algorithm SVR-HMC to sampling from distributions which are only general log-concave but not strongly log-concave.

In detail, we want to sample from the distribution $\pi \propto e^{-f(\mathbf{x})}$, where f is general convex and L-smooth. We follow the similar idea in Dalalyan (2014) to construct a strongly log-concave distribution by adding a quadratic regularizer to the convex and L-smooth function f, which yields

$$\bar{f}(\mathbf{x}) = f(\mathbf{x}) + \lambda \|\mathbf{x}\|_2^2/2,$$

where $\lambda > 0$ is a regularization parameter. Apparently, $\bar{f}$ is λ-strongly convex and $(L + \lambda)$-smooth. Then we can apply Algorithm 1 to function $\bar{f}$, which amounts to sampling from the modified target distribution $\bar{\pi} \propto e^{-\bar{f}}$. We will obtain a sequence $\{\mathbf{x}_k\}_{k=0,\ldots,K}$, whose distribution converges to a unique stationary distribution of Hamiltonian dynamics (1.3), denoted by $\bar{\pi}$. According to Neal et al. (2011), $\bar{\pi}$ is propositional to $e^{-\bar{f}(\mathbf{x})}$, i.e.,

$$\bar{\pi} \propto \exp\big({-\bar{f}(\mathbf{x})}\big) = \exp\Big({-f(\mathbf{x}) - \frac{\lambda}{2}\|\mathbf{x}\|_2^2}\Big).$$

Denote the distribution of $\mathbf{x}_k$ by $P(\mathbf{x}_k)$. We have

$$\mathcal{W}_2(P(\mathbf{x}_k, \pi)) \leq \mathcal{W}_2(P(\mathbf{x}_k), \bar{\pi}) + \mathcal{W}_2(\bar{\pi}, \pi). \quad (4.4)$$

To bound the 2-Wasserstein distance between $P(\mathbf{x}_k)$ and the desired distribution π, we only need to upper bound the 2-Wasserstein distance between two Gibbs distribution $\bar{\pi}$ and π. Before we present our theoretical characterization on this distance, we first lay down the following assumption.

Assumption 4.7. Regarding distribution $\pi \propto e^{-f}$, its fourth-order moment is upper bounded, i.e., there exists a constant $\bar{U}$ such that $\mathbb{E}_\pi[\|\mathbf{x}\|_2^4] \leq \bar{U}d^2$.

The following theorem spells out the convergence rate of SVR-HMC for sampling from a general log-concave distribution.

Theorem 4.8. Under Assumptions 4.1 and 4.7, in order to sample for a general log-concave density $\pi \propto e^{-f(\mathbf{x})}$, the output of Algorithm 1 when applied to $\bar{f}(\mathbf{x}) = f(\mathbf{x}) + \lambda\|\mathbf{x}\|_2^2/2$ satisfies $\mathcal{W}_2\big(P(\mathbf{x}_k), \pi\big) \leq \epsilon$ after

$$\widetilde{O}\Big(n + \frac{d^{11/2}}{\epsilon^6} + \frac{d^{11/3}n^{2/3}}{\epsilon^4}\Big)$$

gradient evaluations.

Regarding sampling from a smooth and general log-concave distribution, to the best of our knowledge, there is no existing theoretical analysis on the convergence of LMC algorithms in 2-Wasserstein distance. Yet the convergence analyses of LMC methods in total variation distance (Dalalyan,

2014; Durmus et al., 2017) and KL-divergence (Cheng & Bartlett, 2017) have recently been established. In detail, Dalalyan (2014) proved a convergence rate of $\widetilde{O}(d^3/\epsilon^4)$ in total variation distance for LMC with general log-concave distributions, which implies $\widetilde{O}(nd^3/\epsilon^4)$ gradient complexity. Durmus et al. (2017) improved the gradient complexity of LMC in total variation distance to $\widetilde{O}(nd^5/\epsilon^2)$. (Cheng & Bartlett, 2017) proved the convergence of LMC in KL-divergence, which attains $\widetilde{O}(nd/\epsilon^3)$ gradient complexity. It is worth noting that our convergence rate in 2-Wasserstein distance is not directly comparable to the aforementioned existing results.

5. Experiments

In this section, we compare the proposed algorithm (**SVR-HMC**) with the state-of-the-art MCMC algorithms for Bayesian learning. To compare the convergence rates for different MCMC algorithms, we conduct the experiments on both synthetic data and real data.

We compare our algorithm with **SGLD** (Welling & Teh, 2011), **VR-SGLD** (Reddi et al., 2016), **HMC** (Cheng & Bartlett, 2017) and **SG-HMC** (Cheng & Bartlett, 2017).

5.1. Simulation Based on Synthetic Data

On the synthetic data, we construct each component function to be $f_i(\mathbf{x}) = (\mathbf{x} - \boldsymbol{a}_i)^\top \boldsymbol{\Sigma}(\mathbf{x} - \boldsymbol{a}_i)/2$, where $\boldsymbol{a}_i$ is a Gaussian random vector drawn from distribution $\mathcal{N}(2, 4 \times \mathbf{I}_{d \times d})$, and $\boldsymbol{\Sigma}$ is a positive definite symmetric matrix with maximum eigenvalue $L = 3/2$ and minimum eigenvalue $\mu = 2/3$. Note that each random vector $\boldsymbol{a}_i$ leads to a particular component function $f_i(\mathbf{x})$. Then it can be observed that the target density $\pi \propto \exp\big(1/n \sum_{i=1}^n f_i(\mathbf{x})\big) = \exp\big((\mathbf{x} - \bar{\boldsymbol{a}})^\top \boldsymbol{\Sigma}(\mathbf{x} - \bar{\boldsymbol{a}})/2\big)$ is a multivariate Gaussian distribution with mean $\bar{\boldsymbol{a}} = 1/n \sum_{i=1}^n \boldsymbol{a}_i$ and covariance matrix $\boldsymbol{\Sigma}$. Moreover, the negative log density $f(\mathbf{x})$ is L-smooth and μ-strongly convex.

In our simulation, we investigate different dimension d and number of component functions n, and show the 2-Wasserstein distance between the target distribution π and that of the output from different algorithms with respect to the number of data passes. In order to estimate the 2-Wasserstein distance between the distribution of each iterate and the target one, we repeat all algorithms for $20,000$ times and obtain $20,000$ random samples for each algorithm in each iteration. In Figure 1, we present the convergence results for three HMC based algorithms (HMC, SG-HMC and SVR-HMC). It is evident that SVR-HMC performs the best among these three algorithms when n is not large enough, and its performance becomes close to that of SG-HMC when the number of component function is increased. This
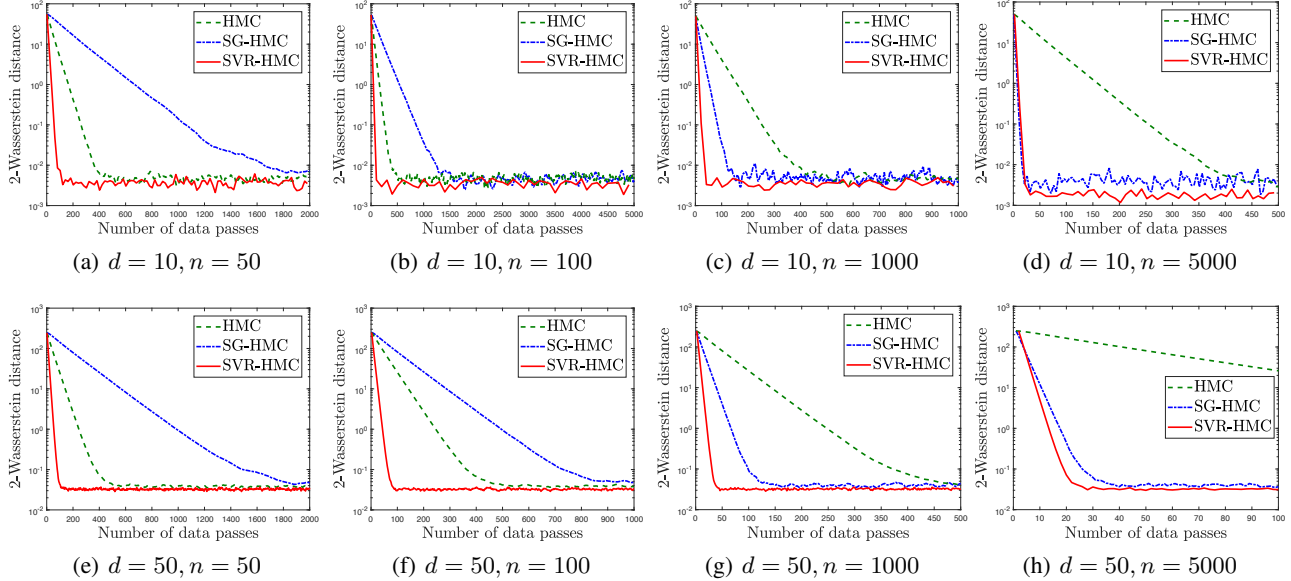
Figure 1. Numerical results for synthetic data, where we compare 3 different algorithms, and show their convergence performance in 2-Wasserstein distance. (a)-(h) represent for different dimensions d and sample sizes n.

phenomenon is well-aligned with our theoretical analysis, since the gradient complexity of our algorithm can be worse than SG-HMC when the sample size n is extremely large.

5.2. Bayesian Logistic Regression for Classification

Table 2. Summary of datasets for Bayesian classification

Dataset	*pima*	*a3a*	*gisette*	*mushroom*
n (training)	384	3185	6000	4062
n (test)	384	29376	1000	4062
d	8	122	5000	112

Now, we apply our algorithm to the Bayesian logistic regression problems. In logistic regression, given n i.i.d. examples $\{\mathbf{a}_i, y_i\}_{i=1,\ldots,n}$, where $\mathbf{a}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$ denote the features and binary labels respectively, the probability mass function of y_i given the feature $\mathbf{a}_i$ is modelled as $p(y_i|\mathbf{a}_i, \mathbf{x}) = 1/(1 + e^{-y_i \mathbf{x}^\top \mathbf{a}_i})$, where $\mathbf{x} \in \mathbb{R}^d$ is the regression parameter. Considering the prior $p(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I})$, the posterior distribution takes the form

$$p(\mathbf{x}|\boldsymbol{A}, \boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{A}, \mathbf{x})p(\mathbf{x}) = \prod_{i=1}^{n} p(y_i|\mathbf{a}_i, \beta)p(\mathbf{x}).$$

where $\boldsymbol{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]^\top$ and $\boldsymbol{Y} = [y_1, y_2, \ldots, y_n]^\top$. The posterior distribution can be written as $p(\mathbf{x}|\boldsymbol{A}, \boldsymbol{Y}) \propto$

$e^{-\sum_{i=1}^{n} f_i(\mathbf{x})}$, where each $f_i(\mathbf{x})$ is in the following form

$$f_i(\mathbf{x}) = n \log\left(1 + \exp(-y_i \mathbf{x}^\top \mathbf{a}_i)\right) + \lambda/2\|\mathbf{x}\|_2^2.$$

We use four binary classification datasets from Libsvm (Chang & Lin, 2011) and UCI machine learning repository (Lichman, 2013), which are summarized in Table 3. Note that *pima* and *mushroom* do not have test data in their original version, and we split them into 50% for training and 50% for test. Following Welling & Teh (2011); Chen et al. (2014; 2015), we report the sample path average and discard the first 50 iterations as burn-in. It is worth noting that we observe similar convergence comparison of different algorithms for larger burn-in period ($= 10^4$). We run each algorithm 20 times and report the averaged results for comparison. Note that variance reduction based algorithms (i.e., VR-SGLD and SVR-HMC) require the first data pass to compute one full gradient. Therefore, in Figure 2, plots of VR-SGLD and VRHMC start from the second data pass while plots of SGLD and SGHMC start from the first data pass. It can be clearly seen that our proposed algorithm is able to converge faster than SGLD and SG-HMC on all datasets, which validates our theoretical analysis of the convergence rate. In addition, although there is no existing non-asymptotic theoretical guarantee for VR-SGLD when the target distribution is strongly log-concave, from Figure 2, we can observe that SVR-HMC also outperforms VR-SGLD on these four datasets, which again demonstrates the superior performance of our algorithm for Bayesian learning. This clearly shows the advantage of our algorithm for Bayesian learning.

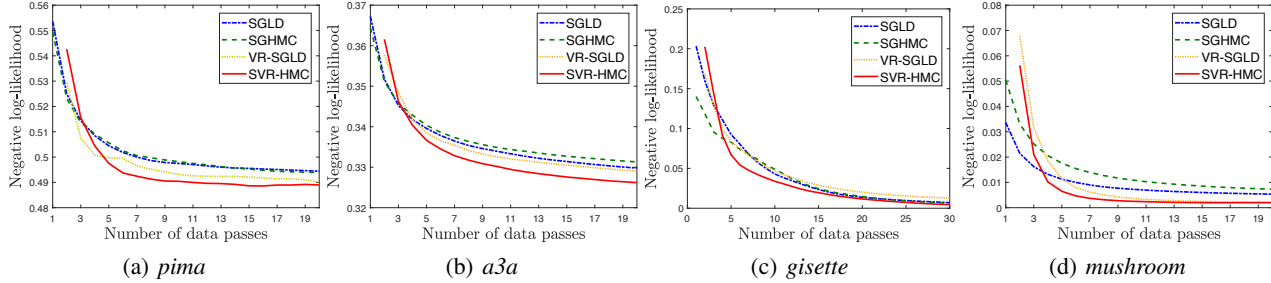(a) *pima* (b) *a3a* (c) *gisette* (d) *mushroom*

Figure 2. Comparison of different algorithms for Bayesian logistic regression, where y axis shows the negative log-likelihood on the test data, and y axis is the number of data passess. (a)-(d) correspond to 4 datasets.
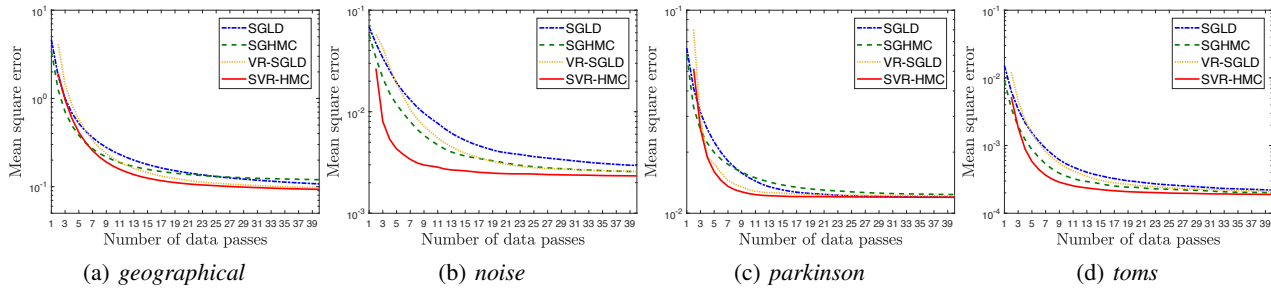

(a) *geographical* (b) *noise* (c) *parkinson* (d) *toms*

Figure 3. Comparison of different algorithms for Bayesian linear regression, where y axis is the mean square errors on the test data, and x axis is the number of data passess. (a)-(d) correspond to 4 datasets.

Table 3. Test error of different algorithms for Bayesian classification after 10 entire data passes on 4 datasets

Dateset	*pima*	*a3a*	*gisette*	*mushroom*
SGLD	0.2314 ± 0.0044	0.1594 ± 0.0018	0.0098 ± 0.0009	$(6.647 + 2.251) \times 10^{-4}$
SGHMC	0.2306 ± 0.0079	0.1591 ± 0.0044	0.0096 ± 0.0006	$(5.916 \pm 2.734) \times 10^{-4}$
VR-SGLD	$0.2299 + 0.0056$	0.1572 ± 0.0012	0.0105 ± 0.0006	$(7.755 \pm 3.231) \times 10^{-4}$
SVR-HMC	0.2289 ± 0.0043	0.1570 ± 0.0019	0.0093 ± 0.0011	$(6.278 \pm 3.149) \times 10^{-4}$

Table 4. Summary of datasets for Bayesian linear regression

Dataset	*geographical*	*noise*	*parkinson*	*toms*
n	1059	1503	5875	45730
d	69	5	21	96

5.3. Bayesian Linear Regression

We also apply our algorithm to Bayesian linear regression, and make comparison with the baseline algorithms. Similar to Bayesian classification, given i.i.d. examples $\{\mathbf{a}_i, y_i\}_{i=1,\ldots,n}$ with $y_i \in \mathbb{R}$, the likelihood of Bayessian linear regression is $p(y_i|\mathbf{a}_i, \mathbf{x}) = \mathcal{N}(\mathbf{x}^\top \mathbf{a}_i, \sigma_a^2)$ and the prior is $\mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I})$. We use 4 datasets, which are summarized in Table 4. In our experiment, we set $\sigma_a^2 = 1$ and $\lambda = 1$, and conduct the normalization of the original data. In addition, we split each dataset into training and test data evenly. Similarly, we compute the sample path average while treating the first 50 iterates as burn in. We report the mean square errors on the test data on these 4 datasets in Figure 3 for different algorithms. It is evident that our

algorithm is faster than all the other baseline algorithms on all the datasets, which further illustrates the advantage of our algorithm for Bayesian learning.

6. Conclusions and Future work

We propose a stochastic variance reduced Hamilton Monte Carlo (HMC) method, for sampling from a smooth and strongly log-concave distribution. We show that, to achieve ϵ accuracy in 2-Wasserstein distance, our algorithm enjoys a faster rate of convergence and better gradient complexity than state-of-the-art HMC and stochastic gradient HMC methods in a wide regime. We also extend our algorithm for sampling from smooth and general log-concave distributions. Experiments on both synthetic and real data verified the superior performance of our algorithm. In the future, we will extend our algorithm to non-log-concave distributions and study the symplectic integration techniques such as Leap-frog integration for Bayesian posterior sampling.

Acknowledgements

References

Ahn, S., Balan, A. K., and Welling, M. Bayesian posterior sampling via stochastic gradient fisher scoring. In *ICML*, 2012.

Allen-Zhu, Z. and Hazan, E. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pp. 699–707, 2016.

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

Baker, J., Fearnhead, P., Fox, E. B., and Nemeth, C. Control variates for stochastic gradient mcmc. *arXiv preprint arXiv:1706.05439*, 2017.

Bakry, D., Gentil, I., and Ledoux, M. *Analysis and geometry of Markov diffusion operators*, volume 348. Springer Science & Business Media, 2013.

Chang, C.-C. and Lin, C.-J. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

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

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

Cheng, X. and Bartlett, P. Convergence of langevin mcmc in kl-divergence. *arXiv preprint arXiv:1705.09048*, 2017.

Cheng, X., Chatterji, N. S., Bartlett, P. L., and Jordan, M. I. Underdamped langevin mcmc: A non-asymptotic analysis. *arXiv preprint arXiv:1707.03663*, 2017.

Chiang, T.-S., Hwang, C.-R., and Sheu, S. J. Diffusion for global optimization in r^n. *SIAM Journal on Control and Optimization*, 25(3):737–753, 1987.

Dalalyan, A. S. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *arXiv preprint arXiv:1412.7392*, 2014.

Dalalyan, A. S. Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent. *arXiv preprint arXiv:1704.04752*, 2017.

Dalalyan, A. S. and Karagulyan, A. G. User-friendly guarantees for the langevin monte carlo with inaccurate gradient. *arXiv preprint arXiv:1710.00095*, 2017.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 1646–1654, 2014.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

Dubey, K. A., Reddi, S. J., Williamson, S. A., Poczos, B., Smola, A. J., and Xing, E. P. Variance reduction in stochastic gradient langevin dynamics. In *Advances in Neural Information Processing Systems*, pp. 1154–1162, 2016.

Durmus, A. and Moulines, E. High-dimensional bayesian inference via the unadjusted langevin algorithm. 2016a.

Durmus, A. and Moulines, E. Sampling from strongly log-concave distributions with the unadjusted langevin algorithm. *arXiv preprint arXiv:1605.01559*, 2016b.

Durmus, A., Moulines, E., et al. Nonasymptotic convergence analysis for the unadjusted langevin algorithm. *The Annals of Applied Probability*, 27(3):1551–1587, 2017.

Eberle, A., Guillin, A., and Zimmer, R. Couplings and quantitative contraction rates for langevin dynamics. *arXiv preprint arXiv:1703.01617*, 2017.

Jarner, S. F. and Hansen, E. Geometric ergodicity of metropolis algorithms. *Stochastic processes and their applications*, 85(2):341–361, 2000.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Kloeden, P. E. and Platen, E. Higher-order implicit strong numerical schemes for stochastic differential equations. *Journal of statistical physics*, 66(1):283–314, 1992.

Lei, L. and Jordan, M. I. Less than a single pass: Stochastically controlled stochastic gradient method. *arXiv preprint arXiv:1609.03261*, 2016.

Lei, L., Ju, C., Chen, J., and Jordan, M. I. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pp. 2345–2355, 2017.

Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Ma, Y.-A., Chen, T., and Fox, E. A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems*, pp. 2917–2925, 2015.

Neal, R. M. et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11), 2011.

Øksendal, B. Stochastic differential equations. In *Stochastic differential equations*, pp. 65–84. Springer, 2003.

Parisi, G. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384, 1981.

Raginsky, M., Rakhlin, A., and Telgarsky, M. Nonconvex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. *arXiv preprint arXiv:1702.03849*, 2017.

Reddi, S. J., Hefny, A., Sra, S., Poczos, B., and Smola, A. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pp. 314–323, 2016.

Roberts, G. O. and Rosenthal, J. S. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

Roberts, G. O. and Stramer, O. Langevin diffusions and metropolis-hastings algorithms. *Methodology and computing in applied probability*, 4(4):337–357, 2002.

Roberts, G. O. and Tweedie, R. L. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, pp. 341–363, 1996.

Roux, N. L., Schmidt, M., and Bach, F. R. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in Neural Information Processing Systems*, pp. 2663–2671, 2012.

Stramer, O. and Tweedie, R. Langevin-type models i: Diffusions with given stationary distributions and their discretizations. *Methodology and Computing in Applied Probability*, 1(3):283–306, 1999a.

Stramer, O. and Tweedie, R. Langevin-type models ii: Self-targeting candidates for mcmc algorithms. *Methodology and Computing in Applied Probability*, 1(3):307–328, 1999b.

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

Xiao, L. and Zhang, T. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Xu, P., Chen, J., Zou, D., and Gu, Q. Global convergence of langevin dynamics based algorithms for nonconvex optimization. *arXiv preprint arXiv:1707.06618*, 2017.

Zhang, Y., Liang, P., and Charikar, M. A hitting time analysis of stochastic gradient langevin dynamics. *arXiv preprint arXiv:1702.05575*, 2017.