Subsampled Stochastic Variance-Reduced Gradient Langevin Dynamics

Difan Zou[*]
Department of Computer Science
University of California
Los Angeles, CA 90095, USA

Pan Xu[*]
Department of Computer Science
University of California
Los Angeles, CA 90095, USA

Quanquan Gu
Department of Computer Science
University of California
Los Angeles, CA 90095, USA

Abstract

Stochastic variance-reduced gradient Langevin dynamics (SVRG-LD) was recently proposed to improve the performance of stochastic gradient Langevin dynamics (SGLD) by reducing the variance of the stochastic gradient. In this paper, we propose a variant of SVRG-LD, namely SVRG-LD$^+$, which replaces the full gradient in each epoch with a subsampled one. We provide a nonasymptotic analysis of the convergence of SVRG-LD$^+$ in 2-Wasserstein distance, and show that SVRG-LD$^+$ enjoys a lower gradient complexity[1] than SVRG-LD, when the sample size is large or the target accuracy requirement is moderate. Our analysis directly implies a sharper convergence rate for SVRG-LD, which improves the existing convergence rate by a factor of $\kappa^{1/6}n^{1/6}$, where κ is the condition number of the log-density function and n is the sample size. Experiments on both synthetic and real-world datasets validate our theoretical results.

1 INTRODUCTION

Markov chain Monte Carlo (MCMC) methods used for posterior sampling have achieved great successes in Bayesian machine learning and Bayesian statistics. Recently, a family of gradient-based MCMC algorithms derived from Langevin dynamics (Parisi, 1981) has become a research hotspot in both Bayesian sampling (Welling & Teh, 2011; Ahn et al., 2012; Wang et al., 2013; Dalalyan, 2014) and optimization (Raginsky et al., 2017; Zhang et al., 2017; Xu et al., 2017). The Langevin dynamics is defined by the following stochastic differential equation (SDE)

$$\mathrm{d}\boldsymbol{X}_t = -\nabla f(\boldsymbol{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{B}_t, \tag{1.1}$$

where $\boldsymbol{X}_t \in \mathbb{R}^d$ is a d-dimensional stochastic process, $\boldsymbol{B}_t \in \mathbb{R}^d$ represents the standard d-dimensional Brownian motion and $-\nabla f(\mathbf{x})$ is called the drift coefficient. It can be shown that the Langevin dynamics converges to an invariant stationary distribution $\pi \propto \exp(-f)$ (Chiang et al., 1987). Based on this observation, various Langevin dynamics based numerical algorithms (Roberts & Tweedie, 1996; Mattingly et al., 2002) have been designed to sample from the target distribution π. Directly applying Euler-Maruyama discretization (Kloeden & Platen, 1992) to SDE (1.1) gives rise to

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \nabla f(\mathbf{x}_k)\eta + \sqrt{2\eta}\boldsymbol{\epsilon}_k, \tag{1.2}$$

where η denotes the step size, and $\boldsymbol{\epsilon}_k \sim N(0, \mathbf{I}_{d\times d})$ is a d-dimensional standard Gaussian random vector. The sampling algorithm using (1.2) as its update formula is typically known as the Langevin Monte Carlo (LMC) algorithm, which has been extensively studied when the target distribution is both log-smooth and strongly log-concave, or even log-Hessian-Lipschitz (Dalalyan, 2014; Durmus & Moulines, 2016; Dalalyan, 2017; Dalalyan & Karagulyan, 2017).

On the other hand, modern machine learning problems often involve an extremely large amount of data. Suppose the dataset consists of n observations, it is often assumed that the function f in the drift term of (1.1) can be written as an average of n finite component functions, i.e.,

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1.3}$$

where each f_i is smooth and f is strongly convex. When n is very large, the LMC algorithm can be inefficient

[*]Equal contribution

[1]Gradient complexity is defined as the required number of stochastic gradient evaluations to reach a target accuracy.

since the gradient evaluation is computationally very expensive. Following the same idea in stochastic optimization, Welling & Teh (2011) proposed the stochastic gradient Langevin dynamics (SGLD) algorithm by replacing the full gradient in (1.2) with a stochastic gradient computed only on a minibatch of data. The SGLD algorithm has been successfully applied to Bayesian learning (Welling & Teh, 2011; Ahn et al., 2012) and training deep neural networks (Chaudhari et al., 2016; Ye et al., 2017), because it can dramatically decrease the number of stochastic gradient evaluations and save a lot computation in practice. Nevertheless, the convergence rate of SGLD is much slower than LMC, which may lead to a worse runtime complexity in certain regime. Regarding the true computational cost of SGLD, Nagapetyan et al. (2017) argued that SGLD is at most better by a constant factor relative to an Euler discretization with full gradients, and raised questions about the good performance of SGLD under the big-data setting. In order to fairly evaluate the performances of stochastic algorithms, one often uses gradient complexity to indicate the efficiency of a sampling algorithm in large scale machine learning problems. When f is smooth, strongly convex and Hessian Lipschitz, Dalalyan & Karagulyan (2017) proved that the gradient complexity of LMC to converge to the stationary distribution π in 2-Wasserstein distance is $\widetilde{O}(n\kappa^2 d^{1/2}/\epsilon)$, where ϵ represents the target accuracy and κ is the condition number of f. In comparison, the gradient complexity of SGLD is $\widetilde{O}(\kappa^2 d\sigma^2/\epsilon^2)$ (Dalalyan, 2017; Dalalyan & Karagulyan, 2017), which is slower than LMC when $n \lesssim d^{1/2}\sigma^2/\epsilon$, where $d\sigma^2$ is an upper bound on the variance of the stochastic gradient.

In order to achieve the best of both worlds, i.e., save the gradient computation of LMC as well as boost the convergence rate of SGLD, Dubey et al. (2016) proposed stochastic variance-reduced gradient Langevin dynamics (SVRG-LD) and stochastic average gradient Langevin dynamics (SAGA-LD), which adapts the idea of variance reduction in stochastic optimization such as SVRG (Johnson & Zhang, 2013; Allen-Zhu & Hazan, 2016; Reddi et al., 2016) and SAGA (Defazio et al., 2014) to gradient-based Monte Carlo methods. However, Dubey et al. (2016) only investigated the performance of both algorithms in terms of mean square error (MSE) of the averaged sample path. Baker et al. (2017) applied zero variance control variates to stochastic MCMC method, and showed that such technique is able to reduce the computational cost of stochastic gradient Langevin dynamics to $O(1)$. Recently, Chatterji et al. (2018) analyzed the convergence rates of SVRG-LD and SAGA-LD to the stationary distribution in 2-Wasserstein distance, and showed that SAGA-LD has a lower gradient complexity compared with SVRG-LD. However, they also observed that when considering low target accuracy regime or the samples size is very large, both of these variance reduction based LMC algorithms perform worse than SGLD, which can converge even within a single data pass. However, their theoretical results suggest that SAGA-LD attains a faster convergence rate than SVRG-LD, which is not consistent with the convergence analyses of SAGA and SVRG for optimization, where both methods have been proved to have the same gradient complexity (Johnson & Zhang, 2013; Defazio et al., 2014). Therefore, Chatterji et al. (2018) raised a question that whether SVRG is less suited than SAGA to work with sampling methods.

In this paper, in order to overcome the shortcomings of SVRG-LD and SAGA-LD, we propose a variant of SVRG-LD, namely SVRG-LD$^+$, by replacing the full gradient computation in the outer loop of SVRG-LD with a subsampled one. The idea of using subsampled gradient instead of full gradient in variance reduction algorithms is originated from the recent work on variance reduction for stochastic optimization (Harikandeh et al., 2015; Lei & Jordan, 2016; Lei et al., 2017), and has also been adopted to Langevin based algorithm by Chen et al. (2017). It is worthy noting that the algorithm proposed in Chen et al. (2017), namely practical vrSG-MCMC, is similar to our algorithm. Nevertheless, the practical SVRG-LD algorithm needs to output all the iterates because its theoretical guarantee is on the sample path. In contrast, our algorithm only needs to output the last iterate, because our theory holds for the last iterate.

1.1 OUR CONTRIBUTIONS

We highlight the major contributions of our work as follows.

- We propose the SVRG-LD$^+$ algorithm and analyze its convergence rate to the target distribution in Wasserstein distance. Specifically, we prove that the SVRG-LD$^+$ algorithm requires $\widetilde{O}\big((n + \kappa^{3/2}n^{1/2}d^{1/2}/\epsilon) \wedge \kappa^2 d\sigma^2/\epsilon^2\big)$ stochastic gradient evaluations to converge to the target distribution in 2-Wasserstein distance within ϵ-accuracy. Our result suggests that when the sample size n is large or the target accuracy ϵ is moderate, the gradient complexity of SVRG-LD$^+$ is better than that of SVRG-LD and SAGA-LD (Chatterji et al., 2018). In addition, the gradient complexity of SVRG-LD$^+$ is never worse than that of SGLD.
- Since SVRG-LD is a special case of SVRG-LD$^+$ when the subsampled gradient is chosen to be the full gradient, our analysis of SVRG-LD$^+$ directly implies a sharp convergence rate of SVRG-LD,

which improves the recent result in Chatterji et al. (2018) by a factor of $\kappa^{1/6}n^{1/6}$, and matches the convergence rate of SAGA-LD (Chatterji et al., 2018). This suggests that both SVRG and SAGA are equally suited to work with sampling methods, and therefore answers the question raised in (Chatterji et al., 2018). Our experiments on both synthetic and real data also show that SVRG-LD and SAGA-LD have comparable performance, which verifies our theory.

We summarize the gradient complexities of existing LMC methods in Table 1, from which we can see that SVRG-LD$^+$ achieves the lowest gradient complexity among all methods. Detailed discussions will be provided in the main theory section.

1.2 ADDITIONAL RELATED WORK

Another line of research that is related to LMC is Hamiltonian Monte Carlo (HMC) method (Neal, 2011), which is based on Hamiltonian dynamics by introducing fictitious momentum variables. Recently, the HMC method has been widely studied and developed both experimentally and theoretically. Specifically, Chen et al. (2014) proposed a stochastic gradient HMC (SG-HMC) algorithm and demonstrated its better performance than SGLD in learning Bayesian neural networks. Chen et al. (2015) conducted a comprehensive analysis for a family of SG-MCMC algorithms including SG-HMC in terms of MSE, and showed that SG-HMC attains a better performance than SGLD if adopting an appropriate discretization method. Ma et al. (2015) proposed a general framework to design samplers from the target distribution, and generated a new state-adaptive sampler on the Riemannian manifold. The nonasymptotic convergence analysis of HMC and SG-HMC was provided in Cheng et al. (2017), where the authors analyzed an underdamped Langevin MCMC algorithm and proved the convergence guarantees in 2-Wasserstein distance. Zou et al. (2018) proposed a stochastic variance-reduced HMC algorithm and proved its convergence rate in 2-Wasserstein distance. Li et al. (2018) analyzed the mean square error of the HMC based algorithm for different discretization schemes. Our work is focused on LMC and is complementary to this line of research.

[2]The convergence of SGLD does not require the Hessian Lipschitz condition. However, Dalalyan & Karagulyan (2017) proved that the convergence rate of SGLD remains the same even with additional Hessian Lipschitz condition.

1.3 NOTATION

We use $[n]$ to denote the index set $\{1, \dots, n\}$. For a random vector $\mathbf{x}_k \in \mathbb{R}^d$, we denote its probability distribution function by $P(\mathbf{x}_k)$. The 2-Wasserstein distance between two probability measures u and v is defined as follows,

$$\mathcal{W}_2(u, v) = \bigg(\inf_{\zeta \in \Gamma(u,v)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\boldsymbol{X}_u - \boldsymbol{X}_v\|_2^2 \mathrm{d}\zeta(\boldsymbol{X}_u, \boldsymbol{X}_v)\bigg)^{1/2},$$

where the infimum is over all joint distributions ζ. We use $a_n = O(b_n)$ to denote that $a_n \leq Cb_n$ for some constant $C > 0$ independent of n, and use $a_n = \widetilde{O}(b_n)$ to hide the logarithmic terms of b_n. We also make use of the notation $a_n \lesssim b_n$ ($a_n \gtrsim b_n$) if a_n is less than (larger than) b_n up to a constant. We use $a \wedge b$ and $a \vee b$ to denote $\min\{a, b\}$ and $\max\{a, b\}$ respectively .

2 ALGORITHM

In this section, we present our SVRG-LD$^+$ algorithm, which is displayed in Algorithm 1.

The algorithm contains multiple epochs. At the beginning of the j-th epoch, we uniformly choose B samples from all training data and obtain a gradient estimator:

$$\widetilde{\mathbf{g}}_j = \nabla f_{\mathcal{I}_j}(\widetilde{\mathbf{x}}_j) = \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} \nabla f_i(\widetilde{\mathbf{x}}_j), \qquad (2.1)$$

where $|\mathcal{I}_j| = B$. At the l-th iteration in the j-th epoch, we define the semi-stochastic gradient as $\mathbf{g}_k = \nabla f_{\widetilde{\mathcal{I}}_k}(\mathbf{x}_k) - \nabla f_{\widetilde{\mathcal{I}}_k}(\widetilde{\mathbf{x}}_j) + \widetilde{\mathbf{g}}_j$, where $k = jm + l$ is the total iteration of the algorithm, m is the length of each epoch, and $\nabla f_{\widetilde{\mathcal{I}}_k}(\mathbf{x}) = 1/|\widetilde{\mathcal{I}}_k| \sum_{i \in \widetilde{\mathcal{I}}_k} \nabla f_i(\mathbf{x})$. Then we perform the following update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \mathbf{g}_k + \sqrt{2\eta}\boldsymbol{\epsilon}_k,$$

where η is the step size and $\boldsymbol{\epsilon}_k \sim N(0, \mathbf{I}_{d \times d})$ is a Gaussian random vector.

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \mathbf{g}_k + \sqrt{2\eta}\boldsymbol{\epsilon}_k,$$

where η is the step size and $\boldsymbol{\epsilon}_k \sim N(0, \mathbf{I}_{d \times d})$ is a Gaussian random vector.

It is worth noting that the major difference between SVRG-LD$^+$ and SVRG-LD (Dubey et al., 2016) is that we replace the full gradient computation in the beginning of each epoch with a subsampled one. On one hand,

Table 1: Gradient complexity of gradient-based Monte Carlo algorithms in 2-Wasserstein distance for sampling from log-smooth, log-Hessian-Lipschitz and strongly log-concave distributions. For the ease of comparison, we follow Chatterji et al. (2018) that assumes $n \lesssim d\sigma^2/(\mu^2\epsilon^2)$ and treats $1/M$ and μ as constants of order $O(1)$.

METHOD	GRADIENT COMPLEXITY	HESSIAN LIPSCHITZ
LMC (Dalalyan & Karagulyan, 2017)	$\widetilde{O}\Big(\frac{n\kappa^2 d^{1/2}}{\epsilon}\Big)$	Yes
SGLD (Dalalyan, 2017)	$\widetilde{O}\Big(\frac{\kappa^2 d\sigma^2}{\epsilon^2}\Big)$	No[2]
SAGA-LD (Chatterji et al., 2018)[3]	$\widetilde{O}\Big(n + \frac{\kappa^{3/2} n^{1/2} d^{1/2}}{\epsilon}\Big)$	Yes
SVRG-LD (Chatterji et al., 2018)	$\widetilde{O}\Big(n + \frac{\kappa^{5/3} n^{2/3} d^{1/2}}{\epsilon}\Big)$	Yes
SVRG-LD (this paper)	$\widetilde{O}\Big(n + \frac{\kappa^{3/2} n^{1/2} d^{1/2}}{\epsilon}\Big)$	Yes
SVRG-LD$^+$ (this paper)	$\widetilde{O}\Big(\big(n + \frac{\kappa^{3/2} n^{1/2} d^{1/2}}{\epsilon}\big) \wedge \frac{\kappa^2 d\sigma^2}{\epsilon^2}\Big)$	Yes

this leads to the consequence that the stochastic gradient $\mathbf{g}_k$ is not an unbiased estimator of the true gradient $\nabla f(\mathbf{x})$, which introduces extra error that poses additional challenge in the analysis. On the other hand, compared with SVRG-LD, it saves gradient computations especially when the sample size n is large. Therefore, the crucial idea of SVRG-LD$^+$ is to make an appropriate trade-off between extra error and saving gradient computation, and the batch size B is a vital parameter which should be carefully designed.

Algorithm 1 SVRG-LD$^+$

input: initial point $\mathbf{x}_0$, step size η, batch size B, mini-batch size b, epoch length m
initialization: $\widetilde{\mathbf{x}}_0 = \mathbf{x}_0$
for $j = 0, \ldots, \lceil K/m \rceil$
  Uniformly sample $\mathcal{I}_j \subseteq [n]$ with $|\mathcal{I}_j| = B$
  $\widetilde{\mathbf{g}}_j = \nabla f_{\mathcal{I}_j}(\widetilde{\mathbf{x}}_j)$
  for $l = 0, \ldots, m-1$
    $k = jm + l$
    Uniformly sample $\widetilde{\mathcal{I}}_k \subseteq [n]$ where $|\widetilde{\mathcal{I}}_k| = b$
    $\mathbf{g}_k = \nabla f_{\widetilde{\mathcal{I}}_k}(\mathbf{x}_k) - \nabla f_{\widetilde{\mathcal{I}}_k}(\widetilde{\mathbf{x}}_j) + \widetilde{\mathbf{g}}_j$
    $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \mathbf{g}_k + \sqrt{2\eta}\boldsymbol{\epsilon}_k$
  end for
  $\widetilde{\mathbf{x}}_{j+1} = \mathbf{x}_{(j+1)m-1}$
end for
output: $\mathbf{x}_K$

3 MAIN THEORY

In this section, we are going to present our main theoretical results on the convergence rate of Algorithm 1 in 2-Wasserstein distance. We will first establish the convergence guarantees of Algorithm 1. Then, we will show that SVRG-LD$^+$ reduces to SVRG-LD when choosing $B = n$, and our analysis leads to a sharp convergence result of SVRG-LD that improves the recent result in Chaudhari et al. (2016).

For the target distribution $\pi \propto e^{-f}$, we first lay down the following assumptions on function $f(\mathbf{x})$, which are required in our analysis.

Assumption 3.1 (Smoothness)**.** There exists a positive constant M such that for each component function $f_i(\mathbf{x})$, the following holds for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\|_2 \le M\|\mathbf{x} - \mathbf{y}\|_2.$$

Note that Assumption 3.1 immediately implies that the function f is also M-smooth, and consequently the target distribution π is M-log-smooth.

Assumption 3.2 (Strong convexity)**.** There exists a positive constant μ such that for function f, the following holds for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

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

The above assumption states that function f is strongly convex, which indicates that the distribution $\pi \propto e^{-f}$ is strongly log-concave.

Assumption 3.3 (Hessian Lipschitz)**.** There exists a positive constant L such that for function f, the following holds for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\|_2 \le L\|\mathbf{x} - \mathbf{y}\|_2.$$

This assumption is essential and useful for proving a faster convergence rate of Langevin Monte Carlo methods (Dalalyan & Karagulyan, 2017; Chatterji et al., 2018).

[3] Different from the definition in (1.3), the finite-sum function f is defined as $f = \sum_{i=1}^n f_i(\mathbf{x})$ in Chatterji et al. (2018), which leads to a difference in the results by a factor of n. To make a fair comparison, we translate their results with the same definition in (1.3).

Assumption 3.4 (Bounded Variance)**.** There exists a constant σ, such that the following holds for all $\mathbf{x} \in \mathbb{R}^d$,

$$\mathbb{E}_i[\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|_2^2] \leq d\sigma^2.$$

Assumption 3.4 is necessary and widely made in stochastic Langevin dynamics based methods such as SGLD (Dalalyan, 2017; Dalalyan & Karagulyan, 2017) and SGHMC (Cheng et al., 2017). However, it should be noted that this assumption is only required for the analysis of the SVRG-LD$^+$ algorithm but not required for SVRG-LD.

In what follows, we will present the convergence results of Algorithm 1. Following the literature (Dalalyan, 2017; Dalalyan & Karagulyan, 2017; Cheng & Bartlett, 2017; Zou et al., 2018; Chatterji et al., 2018), we will focus on the 2-Wasserstein distance between the target distribution $\pi \propto e^{-f}$ and the distribution of the k-th iterate in Algorithm 1. Specifically, we have the following theorem for SVRG-LD$^+$.

Theorem 3.5. Under Assumptions 3.1-3.4, let $P(\mathbf{x}_k)$ denote the distribution of the k-th iterate $\mathbf{x}_k$ in Algorithm 1. Set the step size η to satisfy

$$\eta \leq \min\bigg\{\bigg(\frac{b\mu}{24M^4m^2}\bigg)^{1/3}, \frac{1}{6m(\sigma^2/B+M)}\bigg\}.$$

The 2-Wasserstein distance between $P(\mathbf{x}_k)$ and π is bounded by

$$\begin{aligned}
&\mathcal{W}_2\big(P(\mathbf{x}_k),\pi\big)\\
&\leq (1-\eta\mu/4)^k\mathcal{W}_2(P(\mathbf{x}_0),\pi) + \frac{3\sigma d^{1/2}}{\mu B^{1/2}}\,\mathbb{1}(B\leq n)\\
&\quad + \frac{2\eta(Ld+M^{3/2}d^{1/2})}{\mu}\\
&\quad + \frac{4\eta M(md)^{1/2}\wedge 3\eta^{1/2}d^{1/2}\sigma}{(b\mu)^{1/2}}.
\end{aligned}$$

It is worth noting that the mini-batch size b and the batch size B are two independent parameters in the algorithm that can be chosen separately. In practice, one typically chooses $b \ll B$ (see for example, Harikandeh et al. (2015); Lei & Jordan (2016); Chen et al. (2017) in order to obtain a good convergence result. If we intentionally choose $b > B$ in the algorithm, the evaluation of the semi-stochastic gradient will be even more expensive than that of the subsampled/full gradient in the outer loop, which makes variance reduction techniques no longer effective. The optimal choices of b and B in two different regimes of sample size n will be specified in the following corollaries.

Theorem 3.5 implies that in order to achieve ϵ accuracy in 2-Wasserstein distance, the step size η should be set to be sufficiently small, and the batch size B should be sufficiently large. To address these requirements, we present the following corollaries to show the optimal selections of η and B, and compute the gradient complexity of SVRG-LD$^+$ under different regimes.

We first consider the regime where $n \gtrsim d\sigma^2/(\mu^2\epsilon^2)$.

Corollary 3.6. Under the same assumptions as in Theorem 3.5, suppose the sample size satisfies $n \gtrsim d\sigma^2/(\mu^2\epsilon^2)$, if we set $B = O(d\sigma^2\mu^{-2}\epsilon^{-2})$, $b = O(1)$, $m = O(B)$ and $\eta = O\big(\mu\epsilon^2/(d\sigma^2)\big)$, Algorithm 1 achieves ϵ accuracy in 2-Wasserstein distance after

$$T = \widetilde{O}\bigg(\frac{d\sigma^2}{\mu^2\epsilon^2}\bigg) \tag{3.1}$$

stochastic gradient evaluations.

Remark 3.7. According to Corollary 3.6, if $n \gtrsim d\sigma^2/(\mu^2\epsilon^2)$, then the gradient complexity of SVRG-LD$^+$ in (3.1) matches that of SGLD (Dalalyan, 2017; Dalalyan & Karagulyan, 2017). Note that the gradient complexities of LMC, SAGA-LD and SVRG-LD are at least $\widetilde{O}(n)$ due to the use of full gradients, which indicates that SVRG-LD$^+$ achieves lower gradient complexity than LMC, SAGA-LD and SVRG-LD in this regime.

When the sample size satisfies $n \lesssim d\sigma^2/(\mu^2\epsilon^2)$, we choose the batch size B to be n, i.e., compute the full gradient in the beginning of each epoch in Algorithm 1. In this regime, Algorithm 1 reduces to SVRG-LD (Dubey et al., 2016), and its gradient complexity is characterized by the following corollary.

Corollary 3.8. Under the same assumptions as in Theorem 3.5, suppose the sample size satisfies $n \lesssim d\sigma^2/(\mu^2\epsilon^2)$, if we set $b = 1$ and

$$\eta = \min\bigg\{\frac{\mu\epsilon}{Ld+M^{2/3}d^{1/2}}, \frac{\mu^{1/2}\epsilon}{Md^{1/2}n^{1/2}}\bigg\},$$

SVRG-LD$^+$ achieves ϵ-accuracy in 2-Wasserstein distance after

$$T = \widetilde{O}\bigg(n + \frac{Ld+M^{3/2}d^{1/2}}{\mu^2\epsilon} + \frac{Md^{1/2}n^{1/2}}{\mu^{3/2}\epsilon}\bigg) \tag{3.2}$$

stochastic gradient evaluations.

Remark 3.9. According to Corollary 3.6, if $n \lesssim d\sigma^2/(\mu^2\epsilon^2)$, following Chatterji et al. (2018), if we further assume $n \gtrsim L^2d/(M^2\mu)+\kappa$, and treat $1/M$ and μ as constants of order $O(1)$, then the complexity in (3.2) can be simplified as

$$T = \widetilde{O}\bigg(n + \frac{\kappa^{3/2}d^{1/2}n^{1/2}}{\epsilon}\bigg).$$

It is worth noting that in this regime, Algorithm 1 does not need Assumption 3.4. Moreover, combining the results in Corollaries 3.6 and 3.8, the gradient complexity of SVRG-LD$^+$ can be derived as follows

$$\widetilde{O}\bigg(\Big(n+\frac{\kappa^{3/2}n^{1/2}d^{1/2}}{\epsilon}\Big)\wedge\frac{\kappa^2 d\sigma^2}{\epsilon^2}\bigg), \tag{3.3}$$

where $O(1/\mu^2) = O(\kappa^2/M^2) = O(\kappa^2)$ as $1/M = O(1)$.

Remark 3.10. Corollary 3.8 essentially provides the gradient complexity for SVRG-LD, which is lower than that proved in Chatterji et al. (2018). Recall that their target distribution takes the form

$$\pi \propto \exp\bigg(-\sum_{i=1}^n f_i(\mathbf{x})\bigg) \triangleq \exp\big(-F(\mathbf{x})\big),$$

where the exponent term is different from our definition of f in (1.3) by a factor of $1/n$. In order to make their result comparable to ours, we translate their result to the same definition of f in (1.3), which gives rise to $\widetilde{O}\big(n+\kappa^{5/3}n^{2/3}d^{1/2}/\epsilon\big)$ gradient complexity of SVRG-LD (Chatterji et al., 2018). It is evident that our result improves the gradient complexity of SVRG-LD by a factor of $(\kappa n)^{1/6}$. Last but not the least, our proved gradient complexity of SVRG-LD matches that of SAGA-LD (Chatterji et al., 2018), which suggests that SVRG-LD and SAGA-LD enjoy the same performance.

4 PROOF OF THE MAIN THEORY

In this section we provide the proof for our main theory. We first define an operator $\mathcal{L}$ derived from the Langevin dynamics. Specifically, let $\mathbf{x}_0$ be any starting position, and we denote by $\mathcal{L}_t\mathbf{x}_0$ the random position of the Markov process generated by Langevin dynamics (1.1) after time t. Let $\mathbf{x}^\pi$ denote the random variable that satisfies the stationary distribution $\pi \propto e^{-f}$. In addition, we define $\Delta_k = \mathcal{L}_\eta^k\mathbf{x}^\pi - \mathbf{x}_k$, where $\mathcal{L}_{k\eta} = \mathcal{L}_\eta\circ\mathcal{L}_\eta\circ\cdots\circ\mathcal{L}_\eta = \mathcal{L}_\eta^k$ due to the Markov property of $\mathcal{L}$. Then the following holds trivially

$$\begin{aligned}\Delta_{k+1} &= \mathcal{L}_\eta^{k+1}\mathbf{x}^\pi - \mathbf{x}_{k+1}\\ &= \mathcal{L}_\eta^{k}\mathbf{x}^\pi - \mathbf{x}_{k} + \mathcal{L}_\eta^{k+1}\mathbf{x}^\pi - \mathcal{L}_\eta^{k}\mathbf{x}^\pi - (\mathbf{x}_{k+1}-\mathbf{x}_k).\end{aligned}$$

Consider two synchronously coupled Markov processes $\mathcal{L}_\eta^k\mathbf{x}^\pi$ and $\mathbf{x}_k$ which have shared Brownian motion term in updates $\mathcal{L}_\eta^k\mathbf{x}^\pi \to \mathcal{L}_\eta^{k+1}\mathbf{x}^\pi$ and $\mathbf{x}_k \to \mathbf{x}_{k+1}$, we further have

$$\begin{aligned}\Delta_{k+1} &= \Delta_k + \eta\mathbf{g}_k - \int_0^\eta \nabla f(\mathcal{L}_{k\eta+t}\mathbf{x}^\pi)\mathrm{d}t\\ &= \Delta_k + \eta\big(\mathbf{g}_k - \nabla f(\mathbf{x}_k)\big) - \eta(\nabla f(\mathcal{L}_\eta^k\mathbf{x}^\pi) - \nabla f(\mathbf{x}_k))\\ &\quad - \int_0^\eta \big(\nabla f(\mathcal{L}_{k\eta+t}\mathbf{x}^\pi) - \nabla f(\mathcal{L}_{k\eta}\mathbf{x}^\pi)\big)\mathrm{d}t\\ &= \Delta_k + \eta\boldsymbol{\Phi}_k - \eta\boldsymbol{U}_k - \boldsymbol{S}_k - \boldsymbol{V}_k,\end{aligned} \tag{4.1}$$

where we define

$$\begin{aligned}\boldsymbol{\Phi}_k &= \mathbf{g}_k - \nabla f(\mathbf{x}_k),\\ \boldsymbol{U}_k &= \nabla f(\mathcal{L}_\eta^k\mathbf{x}^\pi) - \nabla f(\mathbf{x}_k),\\ \boldsymbol{S}_k &= \sqrt{2}\int_0^\eta\int_0^t \nabla^2 f(\mathcal{L}_{k\eta+s}\mathbf{x}^\pi)\mathrm{d}\boldsymbol{B}_s\mathrm{d}t,\\ \boldsymbol{V}_k &= \int_0^\eta \big(\nabla f(\mathcal{L}_{k\eta+t}\mathbf{x}^\pi) - \nabla f(\mathcal{L}_{k\eta}\mathbf{x}^\pi)\big)\mathrm{d}t - \boldsymbol{S}_k.\end{aligned}$$

Note that in Algorithm 1, the semi-stochastic gradient $\mathbf{g}_k$ has the following property

$$\begin{aligned}\mathbb{E}[\mathbf{g}_k|\widetilde{\mathbf{x}}_j] &= \mathbb{E}[\nabla f_{\widetilde{\mathcal{I}}_k}(\mathbf{x}_k) - \nabla f_{\widetilde{\mathcal{I}}_k}(\widetilde{\mathbf{x}}_j) + \nabla f_{\mathcal{I}_j}(\widetilde{\mathbf{x}}_j)|\widetilde{\mathbf{x}}_j]\\ &= \mathbb{E}[\nabla f(\mathbf{x}_k) - \nabla f(\widetilde{\mathbf{x}}_j) + \nabla f_{\mathcal{I}_j}(\widetilde{\mathbf{x}}_j)].\end{aligned}$$

Then we can decompose $\boldsymbol{\Phi}_k$ as follows

$$\begin{aligned}\boldsymbol{\Phi}_k &= \mathbf{g}_k - \nabla f(\mathbf{x}_k)\\ &= \underbrace{\nabla f_{\widetilde{\mathcal{I}}_k}(\mathbf{x}_k) - \nabla f_{\widetilde{\mathcal{I}}_k}(\widetilde{\mathbf{x}}_j) - \big(\nabla f(\mathbf{x}_k) - \nabla f(\widetilde{\mathbf{x}}_j)\big)}_{\boldsymbol{\Psi}_k}\\ &\quad + \big(\underbrace{\nabla f_{\mathcal{I}_j}(\widetilde{\mathbf{x}}_j) - \nabla f(\widetilde{\mathbf{x}}_j)}_{\boldsymbol{e}_j}\big).\end{aligned} \tag{4.2}$$

Submitting the above equation into (4.1) yields

$$\Delta_{k+1} = \Delta_k - \eta\boldsymbol{U}_k + \eta\boldsymbol{\Psi}_k + \eta\boldsymbol{e}_j - \boldsymbol{S}_k - \boldsymbol{V}_k. \tag{4.3}$$

Now, we have already obtained the recursive update of Δ_k. In what follows, we will upper bound the ℓ_2-norm of each term on the R.H.S of (4.3). To begin with, we provide the following technical lemmas.

Lemma 4.1. (Dalalyan & Karagulyan, 2017) Under Assumptions 3.1 and 3.2, we have

$$\mathbb{E}[\|\Delta_k - \eta\boldsymbol{U}_k\|_2^2] \le (1-\eta\mu)^2\mathbb{E}[\|\Delta_k\|_2^2],$$

where η denotes the step size, μ is the strongly convex parameter on function $f(\mathbf{x})$.

Lemma 4.2. Under Assumptions 3.1 and 3.2, we have the following upper bound on $\|\boldsymbol{\Psi}_k\|_2^2$.

$$\begin{aligned}&\mathbb{E}[\|\boldsymbol{\Psi}_k\|_2^2]\\ &\le \frac{4d\sigma^2}{b}\wedge\frac{M^2}{b}\big(6m^2\eta^2M^2\mathbb{E}[\|\Delta_{jm}\|_2^2] + G_j\big)e^{2m^2M^2\eta^2},\end{aligned} \tag{4.4}$$

where

$$G_j = 6m^2\eta^2(\mathbb{E}[\|\boldsymbol{e}_j\|_2^2] + Md) + 2md\eta.$$

Lemma 4.3. Under Assumption 3.4, $\|\boldsymbol{e}_j\|_2$ is bounded as follows,

$$\mathbb{E}[\|\boldsymbol{e}_j\|_2^2] = \mathbb{E}[\|\nabla f_{\mathcal{I}_j}(\widetilde{\mathbf{x}}_j) - \nabla f(\widetilde{\mathbf{x}}_j)\|_2^2] \le \frac{d\sigma^2}{B}.$$

In addition, if $B = n$, $\mathbb{E}[\|\boldsymbol{e}_j\|_2^2] = 0$.

Lemma 4.4. (Dalalyan, 2017) Under Assumptions 3.1 and 3.3, regarding to terms $\boldsymbol{S}_k$ and $\boldsymbol{V}_k$ in (4.3), we have the following uniformly upper bound on their ℓ_2-norms

$$\mathbb{E}[\|\boldsymbol{V}_k\|_2^2] \le \frac{\eta^4}{2}(L^2d^2 + M^3d),$$
$$\mathbb{E}[\|\boldsymbol{S}_k\|_2^2] \le \frac{\eta^3M^2d}{3},$$

where M and L denotes the smoothness and Hessian Lipschitz parameters respectively.

Now, we are ready to present the proof for Theorem 3.5.

Proof of Theorem 3.5. Note that

$$\mathbb{E}[\boldsymbol{\Psi}_k|\mathbf{x}_k, \mathcal{L}_\eta^k\mathbf{x}^\pi, \mathcal{L}_\eta^{k+1}\mathbf{x}^\pi] = \mathbf{0},$$

which immediately implies

$$\begin{aligned}
&\mathbb{E}[\|\Delta_{k+1}\|_2^2] \\
&= \mathbb{E}[\|\Delta_k - \eta\boldsymbol{U}_k + \eta\boldsymbol{e}_j - \boldsymbol{S}_k - \boldsymbol{V}_k\|_2^2] + \eta^2\mathbb{E}[\|\boldsymbol{\Psi}_k\|_2^2] \\
&\le (1+\alpha)\mathbb{E}[\|\Delta_k - \eta\boldsymbol{U}_k - \boldsymbol{S}_k\|_2^2] \\
&\quad + (1+1/\alpha)\mathbb{E}[\|\eta\boldsymbol{e}_j - \boldsymbol{V}_k\|_2^2] + \eta^2\mathbb{E}[\|\boldsymbol{\Psi}_k\|_2^2] \\
&= (1+\alpha)\mathbb{E}[\|\Delta_k - \eta\boldsymbol{U}_k\|_2^2 + \|\boldsymbol{S}_k\|_2^2] \\
&\quad + (1+1/\alpha)\mathbb{E}[\|\eta\boldsymbol{e}_j - \boldsymbol{V}_k\|_2^2] + \eta^2\mathbb{E}[\|\boldsymbol{\Psi}_k\|_2^2], \\
&\le (1+\alpha)\mathbb{E}[\|\Delta_k - \eta\boldsymbol{U}_k\|_2^2 + \|\boldsymbol{S}_k\|_2^2] \\
&\quad + 2(1+1/\alpha)\mathbb{E}[\eta^2\|\boldsymbol{e}_j\|_2^2 + \|\boldsymbol{V}_k\|_2^2] + \eta^2\mathbb{E}[\|\boldsymbol{\Psi}_k\|_2^2],
\end{aligned}$$

where $\alpha > 0$ is an arbitrary chosen parameter, the first inequality is by Young's inequality, and the second equality follows from the fact $\mathbb{E}[\boldsymbol{S}_k|\Delta_k, \boldsymbol{U}_k] = \mathbf{0}$. Applying Lemmas 4.1 and 4.2, we have

$$\begin{aligned}
&\mathbb{E}[\|\Delta_{k+1}\|_2^2] \\
&\le (1+\alpha)(1-\eta\mu)^2\mathbb{E}[\|\Delta_k\|_2^2] + (1+\alpha)\mathbb{E}[\|\boldsymbol{S}_k\|_2^2] \\
&\quad + 2(1+1/\alpha)\mathbb{E}[\eta^2\|\boldsymbol{e}_j\|_2^2 + \|\boldsymbol{V}_k\|_2^2] + \eta^2\mathbb{E}[\|\boldsymbol{\Psi}_k\|_2^2 \\
&\le \bigg[(1+\alpha)(1-\eta\mu)^2 + \frac{6\eta^4M^4m^2}{b}e^{2\eta^2m^2M^2}\bigg] \\
&\quad \times \max\big\{\mathbb{E}[\|\Delta_k\|_2^2], \mathbb{E}[\|\Delta_{jm}\|_2^2]\big\} + \Omega_1 + \Omega_2,
\end{aligned} \tag{4.5}$$

where

$$\begin{aligned}
\Omega_1 &= (1+\alpha)\mathbb{E}[\|\boldsymbol{S}_k\|_2^2] \\
&\quad + 2(1+1/\alpha)\mathbb{E}[\eta^2\|\boldsymbol{e}_j\|_2^2 + \|\boldsymbol{V}_k\|_2^2], \\
\Omega_2 &= \frac{4d\sigma^2\eta^2}{b} \wedge \frac{M^2\eta^2G_j}{b}e^{2m^2M^2\eta^2}.
\end{aligned} \tag{4.6}$$

Note that the step size η satisfies $\eta \le \min\big\{(b\mu/(24M^4m^2))^{1/3}, 1/(6m\sigma^2/B + 6mM)\big\}$, we have $\exp(2m^2M^2\eta^2) \le 2$ and $6\eta^4M^4m^2\exp(2m^2M^2\eta^2)/b \le \eta\mu/2$. We choose $\alpha = \eta\mu$, which implies $(1+\alpha)(1-\eta\mu)^2 \le 1-\eta\mu$. Thus, (4.5) can be further rewritten as follows,

$$\begin{aligned}
\mathbb{E}[\|\Delta_{k+1}\|_2^2] &\le (1-\eta\mu/2)\max\big\{\mathbb{E}[\|\Delta_k\|_2^2], \mathbb{E}[\|\Delta_{jm}\|_2^2]\big\} \\
&\quad + \Omega_1 + \Omega_2.
\end{aligned} \tag{4.7}$$

In order to obtain the upper bound of $\mathbb{E}[\|\Delta_k\|_2^2]$, we need to recursively call (4.7). Note that since $jm \le k$, the number of calls to (4.7) must be smaller than k, thus we have

$$\mathbb{E}[\|\Delta_k\|_2^2] \le (1-\eta\mu/2)^k\mathbb{E}[\|\Delta_0\|_2^2\|] + \frac{\Omega_1+\Omega_2}{\eta\mu/2}. \tag{4.8}$$

In what follows, we are going to upper bound Ω_1 and Ω_2. Note that $m \ge 1$ and $M \ge \mu$, we have $\eta\mu \le 1$. Then by the application of Lemmas 4.3 and 4.4, we have

$$\begin{aligned}
\Omega_1 &\le \frac{(1+\alpha)\eta^3M^2d}{3} + 2(1+1/\alpha) \\
&\qquad \times \bigg(\frac{\eta^2d\sigma^2}{B}\mathbb{1}(B<n) + \frac{\eta^4(L^2d^2+M^3d)}{2}\bigg) \\
&\le \frac{2\eta^3M^2d}{3} + \frac{4\eta d\sigma^2}{B\mu} + \frac{2\eta^3(L^2d^2+M^3d)}{\mu}, \\
\Omega_2 &\le \frac{4d\sigma^2\eta^2}{b} \wedge \frac{6M^2md\eta^3}{b}.
\end{aligned}$$

Then we substitute the above upper bounds of Ω_1 and Ω_2 into (4.8), and obtain

$$\begin{aligned}
&\mathbb{E}[\|\Delta_k\|_2^2] \\
&\le (1-\eta\mu/2)^k\mathbb{E}[\|\Delta_0\|_2^2\|] + \frac{\Omega_1+\Omega_2}{\eta\mu/2} \\
&\le (1-\eta\mu/2)^k\mathbb{E}[\|\Delta_0\|_2^2\|] + \frac{8d\sigma^2}{B\mu^2}\mathbb{1}(B<n) \\
&\quad + \frac{4\eta^2(L^2d^2+M^3d)}{\mu^2} + \frac{14\eta^2mM^2d}{b\mu} \wedge \frac{8d\sigma^2\eta}{b\mu}.
\end{aligned}$$

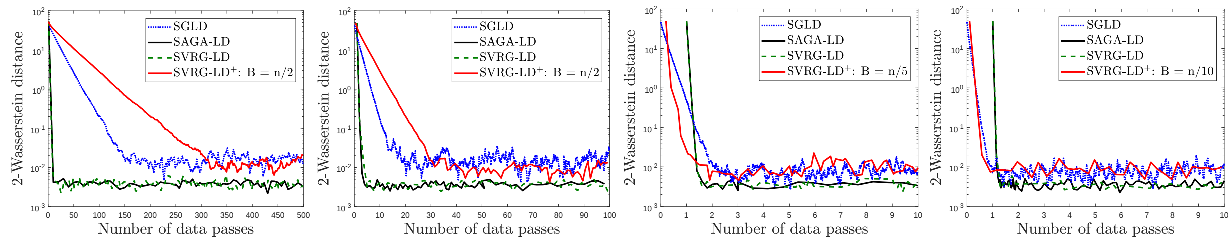
Based on the definition of 2-Wasserstein distance, we have $\mathcal{W}_2^2\big(P(\mathbf{x}_k), \pi\big) \le \mathbb{E}[\|\Delta_k\|_2^2]$. Applying the inequality that $x^2+y^2+z^2 \le (|x|+|y|+|z|)^2$ for all $x, y, z \in \mathbb{R}$, we complete the proof of Theorem 3.5. □

5 EXPERIMENTS

In this section, we are going to verify our theoretical results and evaluate the performances of different Langevin based algorithms on both synthetic and real datasets.

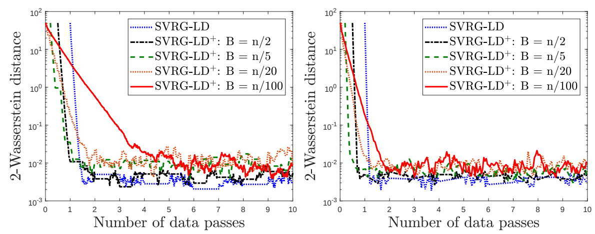
5.1 SIMULATION ON SYNTHETIC DATA

We first validate our theoretical results based on synthetic data. In this simulation, we consider function $f(\mathbf{x}) =$



(a) $n = 100$ (b) $n = 1000$ (c) $n = 20000$ (d) $n = 50000$

Figure 1: Comparison of different algorithms, where y-axis represents the 2-Wasserstein distance computed based on synthetic data, and x-axis is the number of data passes. (a) - (d) represent different sample sizes n.



(a) $n = 20000$ (b) $n = 50000$

Figure 2: Comparison of SVRG-LD$^+$ with different batch size B, where y-axis represents the 2-Wasserstein distance computed based on synthetic data, and x-axis is the number of data passes. (a) and (b) represent different sample sizes n.

$1/n \sum_{i=1}^n f_i(\mathbf{x}) = 1/n \sum_{i=1}^n (\mathbf{x} - \boldsymbol{\theta}_i)^\top \boldsymbol{\Sigma}(\mathbf{x} - \boldsymbol{\theta}_i)/2$, where $\boldsymbol{\Sigma}$ is a symmetric matrix having largest eigenvalue $M = 2$ and smallest eigenvalue $\mu = 1/2$, and $\boldsymbol{\theta}_i$ is drawn from standard multivariate Gaussian distribution.

We first compare the convergence rates of four different algorithms (i.e., SGLD, SVRG-LD, SAGA-LD and SVRG-LD$^+$) to the target distribution in 2-Wasserstein distance, which are reported in Figure 1. It can be seen that there is no obvious difference between the convergence rates of SAGA-LD and SVRG-LD, which verifies our theoretical result of SVRG-LD. Moreover, Figures 1(a) and 1(b) demonstrate that the best choice of B is $B = n$ when the sample size n is small, while Figures 1(c) and 1(d) show that using subsampled gradient $\widetilde{\mathbf{g}}$ in SVRG-LD$^+$ (i.e., $B < n$) is able to improve the performance of SVRG-LD when n is relatively large. This is well aligned with our theoretical analysis that the optimal batch size B is in the order of $O(d\sigma^2/(\epsilon\mu)^2 \wedge n)$.

In Figure 2, we further compare different choices of batch size B in SVRG-LD$^+$ when n is large. Note that since the optimal batch size $B = n$ when the sample size is small, we only perform this experiment on the synthetic datasets with big sample size $n = 20000$ and $n = 50000$. It can be inferred from Figure 2 that if we set $\epsilon = 10^{-2}$, the optimal B in SVRG-LD$^+$ for datasets with sample sizes $n = 20000$ and $n = 50000$ are both $B = 20000/2 = 50000/5 = 10000$. This phenomenon agrees with our theory that for a large $n \gtrsim d\sigma^2/(\epsilon\mu)^2$, the optimal batch size $B = O(d\sigma^2/(\epsilon\mu)^2)$ is independent of n.

5.2 BAYESIAN LOGISTIC REGRESSION

We also collaborate our theoretical results with Bayesian logistic regression. Suppose we are given a dataset with n examples $\{\boldsymbol{X}_i, \mathbf{y}_i\}_{i=1,2,\ldots,n}$, where $\boldsymbol{X}_i \in \mathbb{R}^d$ denotes the d-dimensional feature of the i-th sample, and $\mathbf{y}_i \in \{-1, 1\}$ denotes the corresponding binary label. In Bayesian logistic regression, we assume that the input examples are independent, then the probability distribution of $\mathbf{y}_i$ given features $\boldsymbol{X}_i$ and regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^d$ has the following form

$$p(\mathbf{y}_i|\boldsymbol{X}_i, \boldsymbol{\beta}) = \frac{1}{1 + e^{-\mathbf{y}_i \boldsymbol{\beta}^\top \boldsymbol{X}_i}}.$$

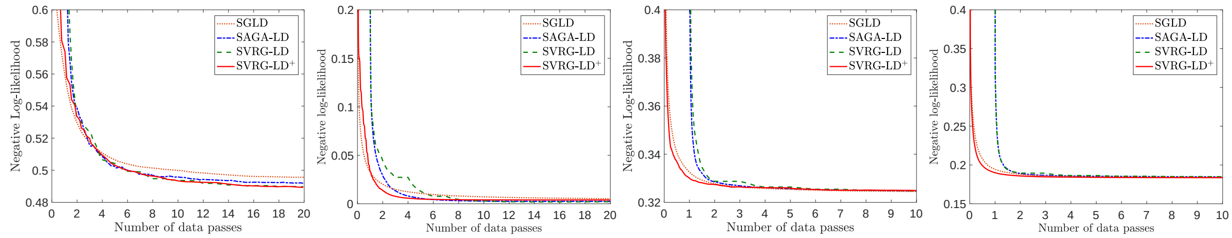
Moreover, the prior of $\boldsymbol{\beta}$ is typically modelled as a Gaussian distribution with zero mean (Dubey et al., 2016; Chatterji et al., 2018), i.e., $\boldsymbol{\beta} \sim N(0, \lambda \mathbf{I}_{d\times d})$. Then we apply the Langevin based method to sample from the posterior distribution of $\boldsymbol{\beta}$, i.e., $p(\boldsymbol{\beta}|\boldsymbol{X}, \mathbf{y}) \propto p(\boldsymbol{\beta}) \prod_{i=1}^n p(\mathbf{y}_i|\boldsymbol{X}_i, \boldsymbol{\beta})$, which implies that the component function $f_i(\boldsymbol{\beta})$ can be written as

$$f_i(\boldsymbol{\beta}) = n \log\left(1 + e^{-\mathbf{y}_i \boldsymbol{\beta} \boldsymbol{X}_i}\right) + \frac{\|\boldsymbol{\beta}\|_2^2}{\lambda}.$$

We apply the Langevin based algorithm to four datasets: *pima*, *mushroom*, *a9a* and *ijcnn1*, which are available at UCI repository[4] and Libsvm website[5]. It is worth noting that *pima* and *mushroom* do not have test datasets like *a9a* and *ijcnn1*, thus we manually partition them into train and test datasets. The basic information of all the datasets is summarized in Table 2. Again, we evaluate the performance of four different algorithms: SGLD,

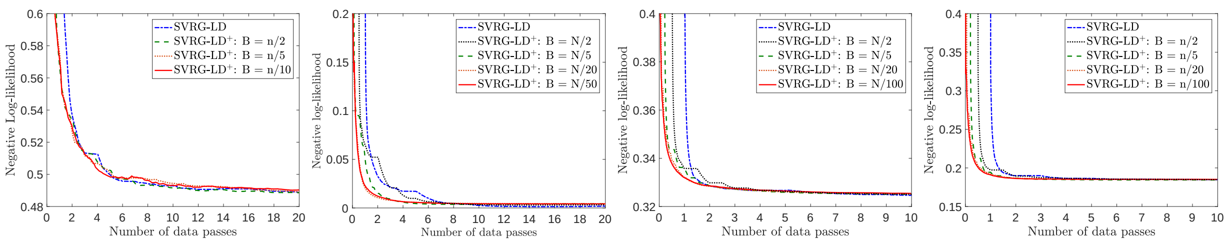
[4] `https://archive.ics.uci.edu/ml/`

[5] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`



(a) *pima* (b) *mushroom* (c) *a9a* (d) *ijcnn1*

Figure 3: Comparison of different algorithms for Bayesian logistic regression, where y axis shows the negative log-likelihood on the test data, and x axis is the number of data passes. (a)-(d) correspond to 4 datasets.



(a) *pima* (b) *mushroom* (c) *a9a* (d) *ijcnn1*

Figure 4: Bayesian Logistic regression results of SVRG-LD$^+$ using different batch size B, where y axis shows the negative log-likelihood on the test data, and x axis is the number of data passes. (a)-(d) correspond to 4 datasets.

SVRG-LD, SAGA-LD and SVRG-LD$^+$, and perform sample path average to estimate the optimal $\boldsymbol{\beta}$, where the minibatch size for each algorithm is set to be 1.

Figure 3 shows the negative log-likelihood of the test examples on these 4 datasets, where each algorithm has been run 10 times to calculate the averaged result. It can be seen that there exists a lag of one data pass for SAGA-LD and SVRG-LD, since they need to scan the entire dataset to compute a full gradient in the beginning. As we can see from the results in Figure 3, SVRG-LD$^+$ converges faster than the other methods, which validates the superior performance of SVRG-LD$^+$. In detail, SVRG-LD$^+$ has a similar convergence rate as SAGA-LD and SVRG-LD when n is small, e.g. datasets *pima*, and performs close to SGLD for relatively large datasets, e.g., *a9a* and *ijcnn1*. This is also consistent with our theoretical results, since the convergence rate of SVRG-LD$^+$ matches that of SGLD when $n \gtrsim d\sigma^2/(\epsilon\mu)^2$.

Table 2: Summary of datasets for Bayesian logistic regression.

Dataset	**pima**	**mushroom**	**a9a**	**ijcnn1**
# training	600	6000	32561	49990
# test	168	2124	16281	91701
d	8	112	123	22

Next, we evaluate the performance of SVRG-LD$^+$ algorithms when choosing different batch sizes B, which are reported in Figure 4. It can be observed that when the batch size is chosen appropriately, SVRG-LD$^+$ converges faster than SVRG-LD, but leading to a slightly higher error. Based on these observations, we can conclude that for Bayesian logistic regression, SVRG-LD$^+$ is more suitable than SVRG-LD when the dataset size is relatively large, and the required accuracy is moderate.

6 CONCLUSIONS

We propose the SVRG-LD$^+$ algorithm and analyze its convergence rate in 2-Wasserstein distance when the target distribution is log-smooth, strongly log-concave and log-Hessian-Lipschitz. Our result implies a sharper convergence analysis of SVRG-LD that improves the state-of-the-art. Experiments on synthetic and real data back up the theoretical results of this paper.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This research was sponsored in part by the National Science Foundation IIS-1618948, IIS-1652539 and SaTC CNS-1717950. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Ahn, Sungjin, Balan, Anoop Korattikara, and Welling, Max. Bayesian posterior sampling via stochastic gradient fisher scoring. In *ICML*, 2012.

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

Baker, Jack, Fearnhead, Paul, Fox, Emily B, and Nemeth, Christopher. Control variates for stochastic gradient MCMC. *arXiv preprint arXiv:1706.05439*, 2017.

Chatterji, Niladri S, Flammarion, Nicolas, Ma, Yi-An, Bartlett, Peter L, and Jordan, Michael I. On the theory of variance reduction for stochastic gradient monte carlo. *arXiv preprint arXiv:1802.05431*, 2018.

Chaudhari, Pratik, Choromanska, Anna, Soatto, Stefano, and LeCun, Yann. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

Chen, Changyou, Ding, Nan, and Carin, Lawrence. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pp. 2278–2286, 2015.

Chen, Changyou, Wang, Wenlin, Zhang, Yizhe, Su, Qinliang, and Carin, Lawrence. A convergence analysis for a class of practical variance-reduction stochastic gradient MCMC. *arXiv preprint arXiv:1709.01180*, 2017.

Chen, Tianqi, Fox, Emily, and Guestrin, Carlos. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, pp. 1683–1691, 2014.

Cheng, Xiang and Bartlett, Peter. Convergence of langevin MCMC in KL-divergence. *arXiv preprint arXiv:1705.09048*, 2017.

Cheng, Xiang, Chatterji, Niladri S, Bartlett, Peter L, and Jordan, Michael I. Underdamped langevin MCMC: A non-asymptotic analysis. *arXiv preprint arXiv:1707.03663*, 2017.

Chiang, Tzuu-Shuh, Hwang, Chii-Ruey, and Sheu, Shuenn Jyi. Diffusion for global optimization in r^n. *SIAM Journal on Control and Optimization*, 25(3): 737–753, 1987.

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

Dalalyan, Arnak S. Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent. *arXiv preprint arXiv:1704.04752*, 2017.

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

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. 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.

Dubey, Kumar Avinava, Reddi, Sashank J, Williamson, Sinead A, Poczos, Barnabas, Smola, Alexander J, and Xing, Eric P. Variance reduction in stochastic gradient langevin dynamics. In *Advances in Neural Information Processing Systems*, pp. 1154–1162, 2016.

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

Harikandeh, Reza, Ahmed, Mohamed Osama, Virani, Alim, Schmidt, Mark, Konečnỳ, Jakub, and Sallinen, Scott. Stopwasting my gradients: Practical SVRG. In *Advances in Neural Information Processing Systems*, pp. 2251–2259, 2015.

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

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

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

Lei, Lihua, Ju, Cheng, Chen, Jianbo, and Jordan, Michael I. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pp. 2345–2355, 2017.

Li, Zhize, Zhang, Tianyi, and Li, Jian. Stochastic gradient hamiltonian monte carlo with variance reduction for bayesian inference. *arXiv preprint arXiv:1803.11159*, 2018.

Ma, Yi-An, Chen, Tianqi, and Fox, Emily. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, pp. 2917–2925, 2015.

Mattingly, Jonathan C, Stuart, Andrew M, and Higham, Desmond J. Ergodicity for sdes and approximations:

locally lipschitz vector fields and degenerate noise. *Stochastic processes and their applications*, 101(2): 185–232, 2002.

Nagapetyan, Tigran, Duncan, Andrew B, Hasenclever, Leonard, Vollmer, Sebastian J, Szpruch, Lukasz, and Zygalakis, Konstantinos. The true cost of stochastic gradient langevin dynamics. *arXiv preprint arXiv:1706.02692*, 2017.

Neal, Radford M. MCMC using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11), 2011.

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

Raginsky, Maxim, Rakhlin, Alexander, and Telgarsky, Matus. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. *arXiv preprint arXiv:1702.03849*, 2017.

Reddi, Sashank J, Hefny, Ahmed, Sra, Suvrit, Poczos, Barnabas, and Smola, Alex. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pp. 314–323, 2016.

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

Wang, Ziyu, Mohamed, Shakir, and Freitas, Nando. Adaptive hamiltonian and riemann manifold monte carlo. In *International Conference on Machine Learning*, pp. 1462–1470, 2013.

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

Xu, Pan, Chen, Jinghui, Zou, Difan, and Gu, Quanquan. Global convergence of langevin dynamics based algorithms for nonconvex optimization. *arXiv preprint arXiv:1707.06618*, 2017.

Ye, Nanyang, Zhu, Zhanxing, and Mantiuk, Rafal K. Langevin dynamics with continuous tempering for high-dimensional non-convex optimization. *arXiv preprint arXiv:1703.04379*, 2017.

Zhang, Yuchen, Liang, Percy, and Charikar, Moses. A hitting time analysis of stochastic gradient langevin dynamics. *arXiv preprint arXiv:1702.05575*, 2017.

Zou, Difan, Xu, Pan, and Gu, Quanquan. Stochastic variance-reduced hamilton monte carlo methods. *arXiv preprint arXiv:1802.04791*, 2018.