Variance Reduction in Stochastic Particle-Optimization Sampling

Jianyi Zhang [1] **Yang Zhao** [2] **Ruiyi Zhang** [1] **Lawrence Carin** [1] **Changyou Chen** [2]

Abstract

Stochastic particle-optimization sampling (SPOS) is a recently-developed scalable Bayesian sampling framework unifying stochastic gradient MCMC (SG-MCMC) and Stein variational gradient descent (SVGD) algorithms based on Wasserstein gradient flows. With a rigorous non-asymptotic convergence theory developed, SPOS can avoid the particle-collapsing pitfall of SVGD. However, the variance-reduction effect in SPOS has not been clear. In this paper, we address this gap by presenting several variance-reduction techniques for SPOS. Specifically, we propose three variants of variance-reduced SPOS, called SAGA particle-optimization sampling (SAGA-POS), SVRG particle-optimization sampling (SVRG-POS) and a variant of SVRG-POS which avoids full gradient computations, denoted as SVRG-POS$^+$. Importantly, we provide non-asymptotic convergence guarantees for these algorithms in terms of the 2-Wasserstein metric and analyze their complexities. The results show our algorithms yield better convergence rates than existing variance-reduced variants of stochastic Langevin dynamics, though more space is required to store the particles in training. Our theory aligns well with experimental results on both synthetic and real datasets.

1. Introduction

Sampling has been an effective tool for approximate Bayesian inference, which is becoming increasingly important in modern machine learning. In the setting of big data, recent research has developed scalable Bayesian sampling algorithms such as stochastic gradient Markov Chain Monte Carlo (SG-MCMC) (Welling & Teh, 2011) and Stein variational gradient descent (SVGD) (Liu & Wang, 2016). These methods have facilitated important real-world applications and achieved impressive results, such as in topic modeling (Gan et al., 2015; Liu et al., 2016), matrix factorization (Chen et al., 2014; Ding et al., 2014; Şimşekli et al., 2016), differential privacy (Wang et al., 2015; Li et al., 2017), Bayesian optimization (Springenberg et al., 2016), reinforcement learning (Haarnoja et al., 2018; Zhang et al., 2018a;b; 2019) and deep neural networks (Li et al., 2016). Generally speaking, these methods use gradient information of a target distribution to generate samples, leading to more effective algorithms compared to traditional sampling methods. Recently, (Chen et al., 2018) proposed a particle-optimization Bayesian sampling framework based on Wasserstein gradient flows, which unified SG-MCMC and SVGD in a new sampling framework called particle-optimization sampling (POS). Furthermore, Zhang et al. (2020) discovered that SVGD endows some unintended pitfalls, *i.e.*, particles tend to collapse under some conditions. As a result, a remedy was proposed to inject random noise into SVGD update equations in the POS framework, leading to stochastic particle-optimization sampling (SPOS) algorithms (Zhang et al., 2020). Remarkably, for the first time, non-asymptotic convergence theory was developed for SPOS (SVGD-type algorithms) in (Zhang et al., 2020).

In order to deal with large-scale datasets, many gradient-based methods for optimization and sampling use stochastic gradients calculated on a mini-batch of a dataset, for computational feasibility. Unfortunately, this has the potential of adding extra variance into the algorithms, which may potentially degrade model performance. To address this issue, variance control has been an important and interesting direction of research. Efficient solutions such as SAGA (Defazio et al., 2014) and SVRG (Johnson & Zhang, 2013) were proposed to reduce variance in stochastic optimization. Subsequently, (Dubey et al., 2016) introduced these techniques in SG-MCMC for Bayesian sampling, which also has achieved great success in practice.

Since SPOS has enjoyed the best of both worlds by combining SG-MCMC and SVGD, it is of great value to further reduce its gradient variance. While both the algorithm and theory have been developed for SPOS, no work has been done to investigate its variance-reduction techniques. Compared with research on SG-MCMC, where variance reduction has been well explored by recent work such as (Dubey

[1]Duke University [2]University at Buffalo, SUNY. Correspondence to: Lawrence Carin <lcarin@duke.edu>, Changyou Chen <changyou@buffalo.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

et al., 2016; Chatterji et al., 2018; Zou et al., 2018; Chen et al., 2019; Zou et al., 2019), it is much more challenging for SPOS to control the variance of stochastic gradients. This is because from a theoretical perspective, SPOS corresponds to nonlinear stochastic differential equations (SDE), where fewer existing mathematical tools can be applied for theoretical analysis. Furthermore, the fact that many particles are used in the algorithm makes it difficult to improve its performance by adding modifications to the way they interact with each other.

In this paper, we take a first attempt to study variance-reduction techniques in SPOS and develop corresponding convergence theory. We adopt recent ideas on variance reduction in SG-MCMC and stochastic-optimization algorithms, and propose three variance-reduced SPOS algorithms, denoted as SAGA particle-optimization sampling (SAGA-POS), SVRG particle-optimization sampling (SVRG-POS) and a variant of SVRG-POS without full-gradient computations, denoted as SVRG-POS$^+$. For all these variants, we prove rigorous theoretical results on their non-asymptotic convergence rates in terms of 2-Wasserstein metrics. Importantly, our theoretical results demonstrate significant improvements in convergence rates over standard SPOS. Remarkably, when comparing our convergence rates with those of variance-reduced stochastic gradient Langevin dynamics (SGLD), our theory indicates faster convergence of variance-reduced SPOS when the number of particles is large enough. Our theoretical results are verified by a number of experiments on both synthetic and real datasets.

2. Preliminaries

2.1. Stochastic gradient MCMC

In Bayesian sampling, one aims at sampling from a posterior distribution $p(\boldsymbol{\theta}|\mathbf{x}) \propto p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ represents the model parameter, and $\mathbf{X} \triangleq \{\mathbf{x}_j\}_{j=1}^N$ is the dataset. Let $p(\boldsymbol{\theta}|\mathbf{X}) = (1/Z)\exp(-U(\boldsymbol{\theta}))$, where

$$U(\boldsymbol{\theta}) = -\log p(\mathbf{X}|\boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \triangleq -\sum_{i=1}^{N} \log p(\mathbf{x}_i|\boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$$

is referred to as the potential energy function, and Z is the normalizing constant. We further define the full gradient F and individual gradient F_j used in this paper:

$$F_j(\boldsymbol{\theta}) \triangleq -\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_j|\boldsymbol{\theta}) - \frac{1}{N}\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta})$$

$$\text{and } F(\boldsymbol{\theta}) \triangleq \nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) = \sum_{j=1}^{N} F_j(\boldsymbol{\theta})$$

We can define a stochastic differential equation, an instance of Itó diffusion, whose stationary distribution equals the target posterior distribution $p(\boldsymbol{\theta}|\mathbf{X})$. For example, consider the following 1st-order Langevin dynamic:

$$\mathrm{d}\boldsymbol{\theta}_t = -\beta^{-1}F(\boldsymbol{\theta}_t)\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}\mathcal{W}_t \,, \tag{1}$$

where t is the time index, $\mathcal{W}_t \in \mathbb{R}^d$ is d-dimensional Brownian motion, and β a scaling factor. By the Fokker-Planck equation (Kolmogoroff, 1931; Risken, 1989), the stationary distribution of (1) equals $p(\boldsymbol{\theta}|\mathbf{X})$.

SG-MCMC algorithms are discretized numerical approximations of the Itó diffusions (1). To make algorithms feasible in a big-data setting, the computationally-expensive term F is replaced with its unbiased stochastic approximation via a random subset of the dataset in each iteration, *e.g.* F can be approximated by a stochastic gradient:

$$G_k \triangleq \frac{N}{B}\sum_{j\in I_k} F_j(\theta_k) = -\nabla \log p(\theta_k) - \frac{N}{B}\sum_{j\in I_k} \nabla_{\theta_k} \log p(\mathbf{x}_j|\theta_k) \,,$$

where I_k is a random subset of $\{1, 2, \cdots, N\}$ with size B. The above definition of G_k reflects the fact that we only have information from $B \ll N$ data points in each iteration. This is the source of the variance we seek to reduce. We also note that G_k is used in standard SVGD and SPOS. As an example, SGLD is a numerical solution of (1), with update equation: $\theta_{k+1} = \theta_k - \beta^{-1}G_k h + \sqrt{2\beta^{-1}h}\xi_k$, where h means the step size and $\xi_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

2.2. Stein variational gradient descent

Different from SG-MCMC, SVGD initializes a set of particles, which are updated iteratively to approximate a posterior distribution. Specifically, we consider a set of particles $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^M$ drawn from some distribution q. SVGD tries to update these particles by doing gradient descent on the interactive particle system via

$$\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}^{(i)} + h\phi(\boldsymbol{\theta}^{(i)}), \;\; \phi = \arg\max_{\phi\in\mathcal{F}} \{\frac{\partial}{\partial h}\mathsf{KL}(q_{[h\phi]}||p)\}_{\{h=0\}}$$

where ϕ is a function perturbation direction chosen to minimize the KL divergence between the updated density $q_{[h\phi]}$ induced by the particles and the posterior $p(\boldsymbol{\theta}|\mathbf{X})$. The standard SVGD algorithm considers $\mathcal{F}$ as the unit ball of a vector-valued reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ associated with a kernel $\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}')$. In such a setting, (Liu & Wang, 2016) shows that

$$\phi(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}'\sim q}[\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}')F(\boldsymbol{\theta}') + \nabla_{\boldsymbol{\theta}'}\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}')]. \tag{2}$$

When approximating the expectation $\mathbb{E}_{\boldsymbol{\theta}'\sim q}[\cdot]$ with an empirical distribution formed by a set of particles $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^M$ and adopting stochastic gradients $G_k^{(i)} \triangleq \frac{N}{B}\sum_{j\in I_k} F_j(\theta_k^{(i)})$,

we arrive at the following update for the particles:

$$\theta_{k+1}^{(i)} = \theta_k^{(i)} + \frac{h}{M}\sum_{q=1}^{M}\left[\kappa(\theta_k^{(q)},\theta_k^{(i)})G_k^{(i)} + \nabla_{\theta_k^{(q)}}\kappa(\theta_k^{(q)},\theta_k^{(i)})\right] \tag{3}$$

SVGD applies (3) repeatedly for all the particles.

2.3. Stochastic particle-optimization sampling

In this paper, we focus on the RBF kernel $\kappa(\boldsymbol{\theta},\boldsymbol{\theta}') = \exp(-\frac{\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|^2}{2\eta^2})$ due to its wide use in both theoretical analysis and practical applications. Hence, we can rewrite the kernel $\kappa(\boldsymbol{\theta},\boldsymbol{\theta}')$ with a simpler function $K(\boldsymbol{\theta}) = \exp(-\frac{\|\boldsymbol{\theta}\|^2}{2\eta^2})$. According to the work of Chen et al. (2018); Zhang et al. (2020), the stationary distribution of the ρ_t in the following partial differential equation equals $p(\boldsymbol{\theta}|\mathbf{X})$:

$$\begin{aligned}\partial_t\rho_t =& \nabla_{\boldsymbol{\theta}}\cdot(\rho_t\beta^{-1}F(\boldsymbol{\theta}) + \rho_t E_{Y\sim\rho_t}K(\boldsymbol{\theta}-Y)F(Y)\\ &- \rho_t(\nabla K * \rho_t) + \beta^{-1}\nabla_{\boldsymbol{\theta}}\rho_t)\,.\end{aligned} \tag{4}$$

When approximating the ρ_t in (4) with an empirical distribution formed by a set of particles $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^M$, Zhang et al. (2020) derive the following diffusion process characterizing the SPOS algorithm.

$$\begin{aligned}\mathrm{d}\boldsymbol{\theta}_t^{(i)} =& -\beta^{-1}F(\boldsymbol{\theta}_t^{(i)})\mathrm{d}t - \frac{1}{M}\sum_{q=1}^{M}K(\boldsymbol{\theta}_t^{(i)}-\boldsymbol{\theta}_t^{(q)})F(\boldsymbol{\theta}_t^{(q)})\mathrm{d}t\\ &+ \frac{1}{M}\sum_{q=1}^{M}\nabla K(\boldsymbol{\theta}_t^{(i)}-\boldsymbol{\theta}_t^{(q)})\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}\mathcal{W}_t^{(i)}\ \ \forall i\end{aligned} \tag{5}$$

Note that if we set the initial distribution of all the particles $\boldsymbol{\theta}_0^{(i)}$ to be identical, the system of these M particles is exchangeable. As a result, the distributions of all the $\boldsymbol{\theta}_t^{(i)}$ are identical and can be denoted as ρ_t. When solving the above diffusion process with a numerical method and adopting stochastic gradients $G_k^{(i)}$, one arrives at the SPOS algorithm of Zhang et al. (2020), with the following update equation:

$$\begin{aligned}\theta_{k+1}^{(i)} =& \theta_k^{(i)} - h\beta^{-1}G_k^{(i)} - \frac{h}{M}\sum_{j=1}^{M}K(\theta_k^{(i)}-\theta_k^{(j)})G_k^{(j)}\\ &+ \frac{h}{M}\sum_{j=1}^{M}\nabla K(\theta_k^{(i)}-\theta_k^{(j)}) + \sqrt{2\beta^{-1}h}\xi_k^{(i)}\end{aligned} \tag{6}$$

where $\xi_k^{(i)} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. SPOS applies an update of (6) repeatedly for all the particles $\theta_k^{(i)}$. Detailed theoretical results for SPOS are reviewed in the Supplementary Material (SM).

3. Variance Reduction in SPOS

In standard SPOS, each particle is updated by adopting $G_k^{(i)} \triangleq \frac{N}{B}\sum_{j\in I_k}F_j(\boldsymbol{\theta}_k^{(i)})$. Because one can only access $B \ll N$ data points in each step, the increased variance of the "noisy gradient" $G_k^{(i)}$ causes a slower convergence rate. A simple way to alleviate this is to increase B by using larger minibatches. Unfortunately, this brings more computational costs, an undesired side effect. Thus more effective variance-reduction methods are needed for SPOS. Inspired by recent work on variance reduction in SGLD, *e.g.*, (Dubey et al., 2016; Chatterji et al., 2018; Zou et al., 2018), we develop three different variance-reduction algorithms for SPOS based on SAGA (Defazio et al., 2014) and SVRG (Johnson & Zhang, 2013) from stochastic optimization.

3.1. SAGA-POS

SAGA-POS generalizes the idea of SAGA (Defazio et al., 2014) to an interactive particle-optimization system. For each particle $\theta_k^{(i)}$, we use $\{g_{k,j}^{(i)}\}_{j=1}^N$ as an approximation for each individual gradient $F_j(\theta_k^{(i)})$. An unbiased estimate of the full gradient $F(\theta_k^{(i)})$ is calculated as:

$$G_k^{(i)} = \sum_{j=1}^{N}g_{k,j}^{(i)} + \frac{N}{B}\sum_{j\in I_k}(F_j(\theta_k^{(i)}) - g_{k,j}^{(i)}),\ \forall i \tag{7}$$

where I_k represents the set of data in mini-batch k. In each iteration, $\{g_{k,j}^{(i)}\}_{j=1}^N$ will be partially updated under the following rule: $g_{k+1,j}^{(i)} = F_j(\theta_k^{(i)})$ if $j \in I_k$, and $g_{k+1,j}^{(i)} = g_{k,j}^{(i)}$ otherwise. The algorithm is described in Algorithm 3.1.

Compared with standard SPOS, SAGA-POS also enjoys high computational efficiency, as it does not require calculation of each $F_j(\theta_k^{(i)})$ to get the full gradient $F(\theta_k^{(i)})$ in each iteration. Hence, the computation time of SAGA-POS is almost the same as that of POS. However, our analysis in Section 4 shows that SAGA-POS enjoys a better convergence rate.

From another perspective, SAGA-POS has the same drawback as SAGA-based algorithms, which requires memory scaling at a rate of $\mathcal{O}(MNd)$ in the worst case. For each particle $\theta_k^{(i)}$, one needs to store N gradient approximations $\{g_{k,j}^{(i)}\}_{j=1}^N$. Fortunately, as pointed out by Dubey et al. (2016); Chatterji et al. (2018), in some applications the memory cost scales only as $\mathcal{O}(N)$ for SAGA-LD, which corresponds to $\mathcal{O}(MN)$ for SAGA-POS as M particles are used.

Remark 1 *When compared with SAGA-LD, we note M particles are used in both SPOS and SAGA-POS. This makes the memory complexity is M times worse than SAGA-LD in training, thus SAGA-POS does not seem to bring any advantages over SAGA-LD. However, this intuition is misleading. As indicated by our theoretical results in Section 4, when the number of particles M is large enough, the convergence*

Algorithm 1 SAGA-POS

Input: A set of initial particles $\{\theta_0^{(i)}\}_{i=1}^M$, each $\theta_0^{(i)} \in \mathbb{R}^d$, step size h_k, batch size B.

Initialize $\{g_{0,j}^{(i)}\}_{j=1}^N = \{F_j(\theta_0^{(i)})\}_{j=1}^N$ for all $i \in \{1, ..., M\}$;

1: **for** iteration k= 0,1,...,T **do**
2: Uniformly sample I_k from $\{1, 2, ..., N\}$ randomly with replacement such that $|I_k| = B$;
3: Sample $\xi_k^{(i)} \sim N(0, I_{d\times d})$, $\forall i$;
4: Update $G_k^{(i)} \leftarrow \sum_{j=1}^N g_{k,j}^{(i)} + \frac{N}{B}\sum_{j\in I_k}(F_j(\theta_k^{(i)}) - g_{k,j}^{(i)})$, $\forall i$;
5: Update each $\theta_k^{(i)}$ with Eq.(6);
6: Update $\{g_{k,j}^{(i)}\}_{j=1}^N$, $\forall i$: if $j \in I_k$, set $g_{k+1,j}^{(i)} \leftarrow F_j(\theta_k^{(i)})$; else, set $g_{k+1,j}^{(i)} \leftarrow g_{k,j}^{(i)}$
7: **end for**

Output: $\{\theta_T^{(i)}\}_{i=1}^M$

rates of our algorithms are actually better than those of variance-reduced SGLD counterparts.

3.2. SVRG-POS

Under limited memory, we propose SVRG-POS, which is based on the SVRG method of Johnson & Zhang (2013). For each particle $\theta_k^{(i)}$, one needs to store a stale parameter $\widetilde{\theta}^{(i)}$, and update it occasionally every τ iterations. At each update, we need to further conduct a global evaluation of full gradients at $\widetilde{\theta}^{(i)}$, *i.e.*, $\widetilde{G}^{(i)} \leftarrow F(\theta_k^{(i)}) = F(\widetilde{\theta}^{(i)})$. An unbiased gradient estimate is then calculated by leveraging both $\widetilde{G}^{(i)}$ and $\widetilde{\theta}^{(i)}$ as:

$$G_k^{(i)} \leftarrow \widetilde{G}^{(i)} + \frac{N}{B}\sum_{j\in I_k}[F_j(\theta_k^{(i)}) - F_j(\widetilde{\theta}^{(i)})] \qquad (8)$$

The algorithm is depicted in Algorithm 3.2, where one only needs to store $\widetilde{\theta}^{(i)}$ and $\widetilde{G}^{(i)}$, instead of gradient estimates of all the individual F_j. Hence the memory cost scales as $\mathcal{O}(Md)$, almost the same as that of standard SPOS.

Although SVRG-POS alleviates the storage requirements of SAGA-POS significantly, it also has the downside that the full gradients, $F(\widetilde{\theta}^{(i)}) = \sum_{j=1}^N F(\widetilde{\theta}^{(i)})$, need to be re-computed every τ iterations, leading to high computation cost in a big-data scenario.

Remark 2 *i) Similar to SAGA-POS, according to our theory in Section 4, SVRG-POS enjoys a faster convergence rate than SVRD-LD – its SGLD counterpart, although M times more space is required for the particles. This provides a trade-off between convergence rates and space complexity. ii) Previous work has shown that SAGA typically outperforms SVRG (Dubey et al., 2016; Chatterji et al., 2018) in terms of convergence speed. This conclusion applies to our case, which will be verified both by theoretical analysis in Section 4 and experiments in Section 5.*

Algorithm 2 SVRG-POS

Input: A set of initial particles $\{\theta_0^{(i)}\}_{i=1}^M$, each $\theta_0^{(i)} \in \mathbb{R}^d$, step size h, epoch length τ, batch size B.

Initialize $\{\widetilde{\theta}^{(i)}\} \leftarrow \{\theta_0^{(i)}\}, \widetilde{G}^{(i)} \leftarrow F(\theta_0^{(i)})$, $\forall i$;

1: **for** iteration k= 0,1,...,T **do**
2: **if** k mod τ =0 **then**
3: **Option I** *i*)Sample $l \sim unif(0, 1, .., \tau - 1)$
ii)Update $\widetilde{\theta}^{(i)} \leftarrow \theta_{k-l}^{(i)}$
Update $\theta_k^{(i)} \leftarrow \widetilde{\theta}^{(i)}$, $\forall i$
iii)Update $\widetilde{G}^{(i)} \leftarrow F(\theta_k^{(i)})$, $\forall i$
4: **Option II** *i*) Update $\widetilde{\theta}^{(i)} \leftarrow \theta_k^{(i)}$
ii)Update $\widetilde{G}^{(i)} \leftarrow F(\theta_k^{(i)})$, $\forall i$
5: **end if**
6: Uniformly sample I_k from $\{1, 2, ..., N\}$ randomly with replacement such that $|I_k| = B$;
7: Sample $\xi_k^{(i)} \sim N(0, I_{d\times d})$, $\forall i$;
8: Update $G_k^{(i)} \leftarrow \widetilde{G}^{(i)} + \frac{N}{B}\sum_{j\in I_k}[F_j(\theta_k^{(i)}) - F_j(\widetilde{\theta}^{(i)})]$, $\forall i$;
9: Update each $\theta_k^{(i)}$ with Eq.(6)
10: **end for**

Output: $\{\theta_T^{(i)}\}_{i=1}^M$

3.3. SVRG-POS$^+$

The need for full gradient computation in SVRG-POS motivates the development of SVRG-POS$^+$. Our algorithm is also inspired by the recent work of SVRG-LD$^+$ on reducing the computational cost in SVRG-LD (Zou et al., 2018). The main idea in SVRG-POS$^+$ is to replace the full gradient computation every τ iterations with a subsampled gradient, *i.e.*, to uniformly sample $|J_k| = b$ data points where J_k are random samples from $\{1, 2, ..., N\}$ with replacement. Given the sub-sampled data, $\widetilde{\theta}^{(i)}$ and $\widetilde{G}^{(i)}$ are updated as: $\widetilde{\theta}^{(i)} = \theta_k^{(i)}$, $\widetilde{G}^{(i)} = \frac{N}{b}\sum_{j\in J_k} F_j(\theta_k^{(i)})$. The full algorithm is shown in Algorithm 3.3.

4. Convergence Analysis

We prove non-asymptotic convergence rates for the SAGA-POS, SVRG-POS and SVRG-POS$^+$ algorithms under the 2-Wasserstein metric, defined as

$$\mathcal{W}_2(\mu, \nu) = \left(\inf_{\zeta\in\Gamma(\mu,\nu)}\int_{\mathbb{R}^d\times\mathbb{R}^d}\|X_\mu - X_\nu\|^2 d\zeta(X_\mu, X_\nu)\right)^{\frac{1}{2}}$$

Algorithm 3 SVRG-POS$^+$

Input : A set of initial particles $\{\theta_0^{(i)}\}_{i=1}^M$, each $\theta_0^{(i)} \in \mathbb{R}^d$, step size h, epoch length τ, batch size B.

Initialize $\{\widetilde{\theta}^{(i)}\} \leftarrow \{\theta_0^{(i)}\}, \widetilde{G}^{(i)} \leftarrow F(\theta_0^{(i)}),\ \forall i$;

1: **for** iteration k= 0,1,...,T **do**
2: **if** k mod τ =0 **then**
3: *i*) Uniformly sample J_k from $\{1, 2, ..., N\}$ with replacement such that $|J_k| = b$;
ii) Update $\widetilde{\theta}^{(i)} \leftarrow \theta_k^{(i)}$ $\widetilde{G}^{(i)} \leftarrow \frac{N}{b}\sum_{j\in J_k} F_j(\theta_k^{(i)}),\ \forall i$;
4: **end if**
5: Uniformly sample I_k from $\{1, 2, ..., N\}$ with replacement such that $|I_k| = B$;
6: Sample $\xi_k^{(i)} \sim N(0, I_{d\times d}),\ \forall i$;
7: Update $G_k^{(i)} \leftarrow \widetilde{G}^{(i)} + \frac{N}{B}\sum_{j\in I_k}[F_j(\theta_k^{(i)}) - F_j(\widetilde{\theta}^{(i)})],\ \forall i$;
8: Update each $\theta_k^{(i)}$ with Eq.(6)
9: **end for**

Output: $\{\theta_T^{(i)}\}_{i=1}^M$

where $\Gamma(\mu, \nu)$ is the set of all coupling of μ and ν on $\mathbb{R}^d \times \mathbb{R}^d$, with marginal distributions matching μ and ν. Let μ^* denote our target distribution, and μ_T the distribution of $\frac{1}{M}\sum_{i=1}^M \theta_T^{(i)}$ derived via (5) after T iterations. Our analysis aims to bound $\mathcal{W}_2(\mu_T, \mu^*)$. We first introduce our assumptions.

Assumption 1 *F and K satisfy the following conditions:*

- *There exists a positive constant m_F such that $\langle F(\boldsymbol{\theta}) - F(\boldsymbol{\theta}'), \boldsymbol{\theta} - \boldsymbol{\theta}'\rangle \geq m_F\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2$; F is L_F-Lipschitz continuous, i.e., $\|F(\boldsymbol{\theta}) - F(\boldsymbol{\theta}')\| \leq L_F\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$;*
- *K is L_K-Lipschitz continuous for some $L_K \geq 0$, i.e., $\|K(\boldsymbol{\theta}) - K(\boldsymbol{\theta}')\| \leq L_K\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$; and ∇K is $L_{\nabla K}$-Lipschitz continuous for some $L_{\nabla K} \geq 0$, i.e., $\|\nabla K(\boldsymbol{\theta}) - \nabla K(\boldsymbol{\theta}')\| \leq L_{\nabla K}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$;*
- *K is an even function, i.e., $K(-\boldsymbol{\theta}) = K(\boldsymbol{\theta})$;*
- *The initial probability law of each particle has a bounded and strictly positive density ν_0 with respect to the Lebesgue measure on $\mathbb{R}^d$, and $\gamma_0 \triangleq \log\int_{\mathbb{R}^d} e^{\|\theta\|^2}\nu_0(\theta)d\theta < \infty$*

Assumption 2 *There exists a constant $D_F > 0$ such that $\|\nabla F(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta}')\| \leq D_F\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$.*

Assumption 3 *There exits a constant σ such that for all $j \in \{1, 2, ..., N\}$,*

$$\mathbb{E}[\|F_j(\boldsymbol{\theta}) - \frac{1}{N}\sum_{j=1}^N F_j(\boldsymbol{\theta})\|^2] \leq d\sigma^2/N^2$$

Assumption 4 *There exist some positive constants H_1, H_2 such that $\|F(\boldsymbol{\theta}_2)K(\boldsymbol{\theta}_1' - \boldsymbol{\theta}_1) - F(\boldsymbol{\theta}_2)K(\boldsymbol{\theta}_2' - \boldsymbol{\theta}_2)\| \leq H_1L_K\|\boldsymbol{\theta}_1' - \boldsymbol{\theta}_1 - (\boldsymbol{\theta}_2' - \boldsymbol{\theta}_2))\|$ and $\|F(\boldsymbol{\theta}_2)\nabla K(\boldsymbol{\theta}_1' - \boldsymbol{\theta}_1) - F(\boldsymbol{\theta}_2)\nabla K(\boldsymbol{\theta}_2' - \boldsymbol{\theta}_2)\| \leq H_2L_{\nabla K}\|\boldsymbol{\theta}_1' - \boldsymbol{\theta}_1 - (\boldsymbol{\theta}_2' - \boldsymbol{\theta}_2))\|$*

Remark 3 *i) Assumption 1 is adopted from (Zhang et al., 2020). The first bullet of Assumption 1 suggests $U(\cdot)$ is a strongly convex function, which is the general assumption in analyzing SGLD (Dalalyan & Karagulyan, 2017; Durmus & Moulines, 2016) and its variance-reduced variants (Zou et al., 2018; Chatterji et al., 2018). Although some work has been done on investigating the non-convex case, the convex case is a more common case, which is more instructive and meaningful for addressing practical issues (Dalalyan & Karagulyan, 2017; Durmus & Moulines, 2016; Zou et al., 2018; Chatterji et al., 2018). ii) All of the m_F, L_F and D_F can scale linearly with N. iii) $K(\boldsymbol{\theta}) = \exp(-\frac{\|\boldsymbol{\theta}\|^2}{2\eta^2})$ can satisfy the above assumptions by setting the bandwidth large enough. K can also be expressed as Hessian Lipschitz with some positive constant $D_{\nabla^2 K}$, and $\|\nabla K\|$ can be bounded by some positive constant $H_{\nabla K}$. iv) Assumption 4 can be viewed as an extension to the Lipschitz continuity mentioned in Assumption 1, and it is used to bridge the work of (Chatterji et al., 2018) and (Zhang et al., 2020). We allow H_1, H_2 to be related to F, which can scale linearly with N.*

Now we present a convergence analysis for our algorithms, where α is some positive constant independent of T.

Theorem 1 *Let μ_T denote the distribution of the particles after T iterations with SAGA-POS, and consider step size $h < \frac{B}{8C_2N}$ and batch size $B \geq 9$. Under Assumptions 1 and 2, the convergence rate of SAGA-POS is bounded as*

$$\mathcal{W}_2(\mu_T,\mu^*) \leq \frac{C_1}{\sqrt{M}} + 5\exp(-\frac{C_3h}{4}T)\mathcal{W}_2(\mu_0, \mu^*) + \frac{2hC_4dM^{1/2-\alpha}}{C_3} + \frac{2hC_2{}^{\frac{3}{2}}\sqrt{d}}{C_3M^\alpha} + \frac{24hC_2\sqrt{dN}}{M^\alpha\sqrt{C_3B}}, \tag{9}$$

where (C_1, C_2, C_3, C_4) are some positive constant presented in the appendix.

Theorem 2 *Let μ_T denote the distribution of the particles after T iterations with SVRG-POS in Algorithm 3.2. Under Assumptions 1 and 2, if we choose Option I and set the step size $h < \frac{1}{8C_2}$, the batch size $B \geq 2$ and the epoch length $\tau = \frac{4}{hC_3(1-2hC_2(1+2/B))}$, the convergence rate of SVRG-POS is bounded, for all T such that T mod τ equal*

0, as

$$\mathcal{W}_2(\mu_T,\mu^*) \le \frac{C_1}{\sqrt{M}} + \exp(-\frac{C_3 h}{56}T)\frac{\sqrt{C_2}}{\sqrt{C_3}}\mathcal{W}_2(\mu_0,\mu^*) \quad (10)$$
$$+ \frac{2hC_4 dM^{1/2-\alpha}}{C_3} + \frac{2hC_2{}^{\frac{3}{2}}\sqrt{d}}{C_3 M^\alpha} + \frac{64C_2{}^{\frac{3}{2}}\sqrt{hd}}{M^\alpha\sqrt{B}C_3}.$$

If we choose Option II and set the step size $h < \frac{\sqrt{B}}{4\tau C_2}$, the convergence rate of SVRG-POS is bounded for all T as

$$\mathcal{W}_2(\mu_T,\mu^*) \le \frac{C_1}{\sqrt{M}} + \exp(-\frac{C_3 h}{4}T)\mathcal{W}_2(\mu_0,\mu^*) \quad (11)$$
$$+ \frac{\sqrt{2}hC_4 dM^{1/2-\alpha}}{C_3} + \frac{5hC_2{}^{\frac{3}{2}}\sqrt{d}}{C_3 M^\alpha} + \frac{9hC_2\tau\sqrt{d}}{M^\alpha\sqrt{BC_3}}.$$

Theorem 3 *Let μ_T denote the distribution of particles after T iterations with SVRG-POS$^+$. Under Assumptions 1, 2 and 3, if we set the step size $h \le \min\{(\frac{BC_3}{24C_2{}^4\tau^2})^{\frac{1}{3}}, \frac{1}{6\tau(C_5{}^2/b+C_2)}\}$, the convergence rate of SVRG-POS$^+$ is bounded for all T as*

$$\mathcal{W}_2(\mu_T,\mu^*) \le \frac{C_1}{\sqrt{M}} + (1-hC_3/4)^T\mathcal{W}_2(\mu_0,\mu^*) \quad (12)$$
$$+ \frac{3C_5 d^{1/2}}{M^\alpha C_3 b^{1/2}}\mathbf{1}(b\le N) + \frac{2h(C_4 dM^{1/2-\alpha})}{C_3}$$
$$+ \frac{2hC_2^{3/2}d^{1/2}}{C_3 M^\alpha} + \frac{4hC_2(\tau d)^{1/2}\wedge 3h^{1/2}d^{1/2}C_5}{M^\alpha\sqrt{BC_3}}.$$

Since the complexity has been discussed in Section 3, we mainly focus on discussing the convergence rates here. Due to space limits, we move the comparison between convergence rates of the standard SPOS and its variance-reduced counterparts (such as SAGA-POS) into the SM. Specifically, adopting the standard framework of comparing different variance-reduction techniques in SGLD (Dubey et al., 2016; Chatterji et al., 2018; Zou et al., 2018), we focus on the scenario where m_f, L_F, H_F and D_F all scale linearly with N where $N \gg d$. In this case, the last term in Theorem 1 dominates for SAGA-POS, $\mathcal{O}(\frac{hC_2\sqrt{d}}{M^\alpha B}) \approx \mathcal{O}(\frac{hN\sqrt{d}}{M^\alpha B})$. Thus, to achieve an accuracy of ε, we need a stepsize of $h_{ag} = \mathcal{O}(\frac{\varepsilon M^\alpha B}{N\sqrt{d}})$. For SVRG-POS, the dominate term in Theorem 2 is $\mathcal{O}(\frac{\sqrt{hNd}}{M^\alpha\sqrt{B}})$ for Option I and $\mathcal{O}(\frac{\tau hN\sqrt{d}}{M^\alpha\sqrt{B}})$ for Option II. Hence, for an accuracy of ε, the corresponding step sizes are $h_{vr1} = \mathcal{O}(\frac{\varepsilon^2 M^{2\alpha}B}{Nd})$ and $h_{vr2} = \mathcal{O}(\frac{\varepsilon M^\alpha\sqrt{B}}{\tau N\sqrt{d}})$, respectively. Due to the fact that the mixing time T for these methods is roughly proportional to the reciprocal of step size (Chatterji et al., 2018), it is seen that when ε is small enough, one can have $h_{vr1} \ll h_{ag}$, which causes SAGA-POS to converge faster than SVRG-POS (Option I). Similar results hold for Option II since the factor $\frac{1}{\sqrt{B}\tau}$ in h_{vr2} would make the step size even smaller. More theoretical results are given in the SM.

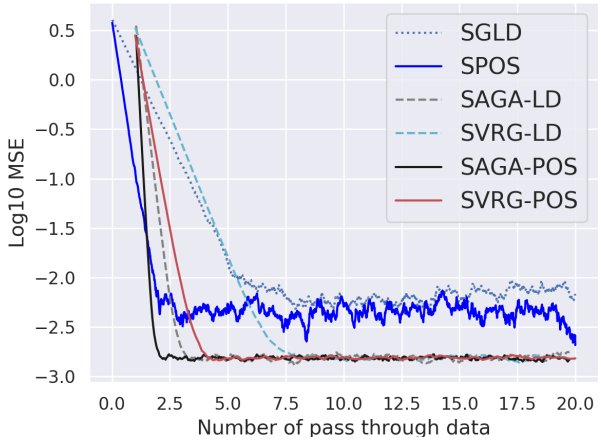


Figure 1. Log-MSE of the mean parameter versus the number of dataset pass.

Remark 4 *We have provided a theoretical analysis to support the statement of i) in Remark 2. Moreover, we should also notice in SAGA-POS, stepsize $h_{ag} = \mathcal{O}(\frac{\varepsilon M^\alpha B}{N\sqrt{d}})$ has an extra factor, M^α, compared with the step size $\mathcal{O}(\frac{\varepsilon B}{N\sqrt{d}})$ used in SAGA-LD (Chatterji et al., 2018)[1]. This means SAGA-POS with more particles (M is large) could outperform SAGA-LD. SVRG-POS and SVRG-POS$^+$ yield similar conclusions. This provides theoretical support for the statements of Remark 1 and i) in Remark 2. Furthermore, an interesting result from the above discussion is that when $h_{vr1} = \mathcal{O}(\frac{\varepsilon^2 M^{2\alpha}B}{Nd})$ in SVRG-POS, there is an extra factor M compared to the stepsize $\mathcal{O}(\frac{\varepsilon^2 B}{Nd})$ in SVRG-LD (Chatterji et al., 2018). Since the order of $M^{2\alpha}$ is higher than M^α, one expects that the improvement of SVRG-POS over SVRG-LD is more significant than that of SAGA-POS over SAGA-LD. This conclusion is verified in our experiments.*

5. Experiments

We conduct experiments to verify our theory, and compare SAGA-POS, SVRG-POS and SVRG-POS$^+$ with existing representative Bayesian sampling methods with/without variance-reduction techniques, *e.g.* SGLD and SPOS without variance reduction; SAGA-LD, SVRG-LD and SVRG-LD$^+$ with variance reduction. For SVRG-POS, we focus on Option I in Algorithm 3.2 to verify our theory.

5.1. Synthetic log-normal distribution

We first evaluate our proposed algorithms on log-normal synthetic data, drawn from $p(\mathbf{x}|\boldsymbol{\mu}) = \frac{1}{\mathbf{x}\sqrt{2\pi}}\exp(-\frac{(\ln\mathbf{x}-\boldsymbol{\mu})^2}{2})$, where $\mathbf{x}, \boldsymbol{\mu} \in \mathbb{R}^{10}$. Like other variance-reduction algorithms (Chatterji et al., 2018), we calculate log-MSE of the sampled "mean" w.r.t. the true value, and plot the log-

[1]For fair comparisons with our algorithms, we consider variance-reduced versions of SGLD with M independent chains.

MSE versus number of passes through data. The results are plotted in Figure 1, which shows that SAGA-POS and SVRG-POS converge the fastest among all algorithms. It is also interesting to see SPOS even outperforms both SAGA-LD and SVRG-LD.

5.2. Bayesian logistic regression

Following related work in (Dubey et al., 2016), we test the proposed algorithms for Bayesian-logistic-regression (BLR) on four publicly available datasets from the UCI machine learning repository: $Australian$ (690-14), $Pima$ (768-8), $Diabetic$ (1151-20) and $Susy$ (100000-18), where $(N-d)$ means a dataset of N data points with dimensionality d. The first three datasets are relatively small, and the last one is large and is suitable for evaluating scalable Bayesian sampling algorithms.

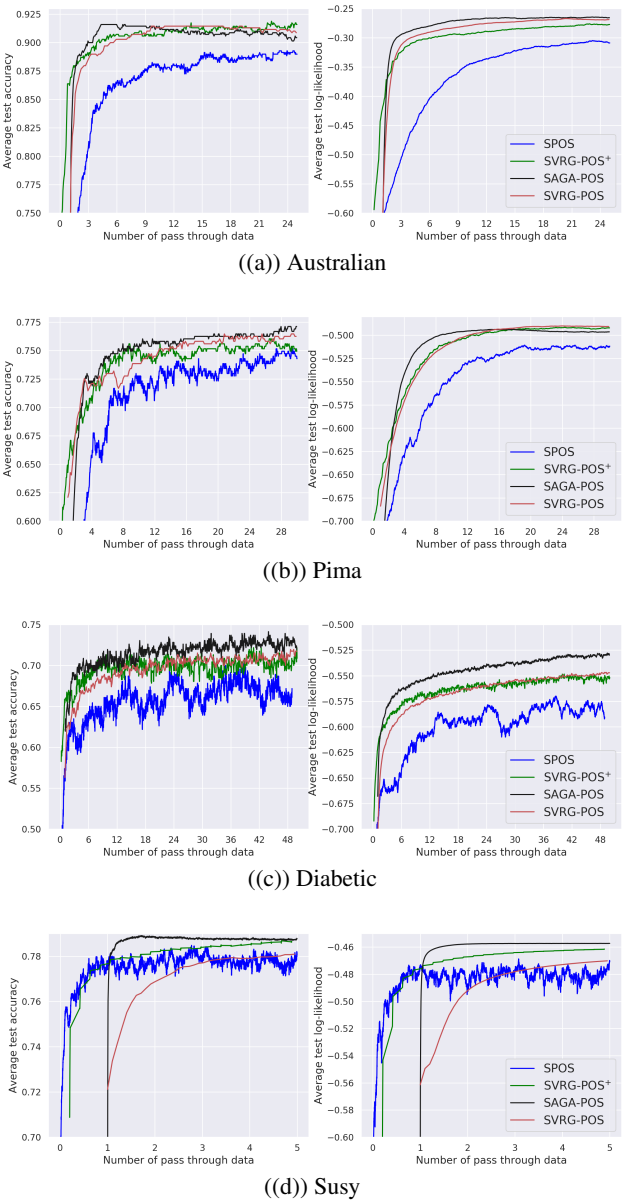
Consider a dataset $\{x_i, y_i\}_{i=1}^N$ with N samples, where $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. The likelihood of a BLR model is written as $p(y_i = 1|X_i, \alpha) = \textbf{sigmoid}(\alpha^T X_i)$ with regression coefficient $\alpha \in \mathbb{R}^d$, which for simplicity is assumed to be sampled from a standard multivariate Gaussian prior $\mathcal{N}(0, I)$. The datasets are split into 80% training data and 20% testing data. Optimized constant stepsizes are applied for each algorithm via grid search. Following existing work, we report testing accuracy and log-likelihood versus the number of data passes for each dataset, averaging over 10 runs with 50 particles. The minibatch size is set to 15 for all experiments.

5.2.1. Variance-reduced SPOS versus SPOS

We first compare SAGA-POS, SVRG-POS and SVRG-POS$^+$ with SPOS without the variance reduction proposed in (Zhang et al., 2020). The testing accuracies and log-likelihoods versus number of passes through data on the four datasets are plotted in Figure 2. It is observed that SAGA-POS converges faster than both SVRG-POS and SVRG-POS$^+$, all of which significantly outperform SPOS. On the largest dataset *SUSY*, SAGA-POS starts only after one pass through the data, which then converges quickly, outperforming the other algorithms. SVRG-POS$^+$ outperforms SVRG-POS because the dataset *SUSY* is large that SVRG-POS$^+$ only requires minibatch calculations. All of these phenomena are supported by our theory.

5.2.2. Variance-reduced SPOS versus variance-reduced SGLD

Next we compare the three variance-reduced SPOS algorithms with SGLD counterparts, *i.e.*, SAGA-LD, SVRG-LD and SVRG-LD$^+$. The results are plotted in Figure 3. Similar phenomena are observed, where both SAGA-POS and SVRG-POS outperform SAGA-LD and SVRG-LD, respectively, consistent with our theoretical results discussed



((a)) Australian

((b)) Pima

((c)) Diabetic

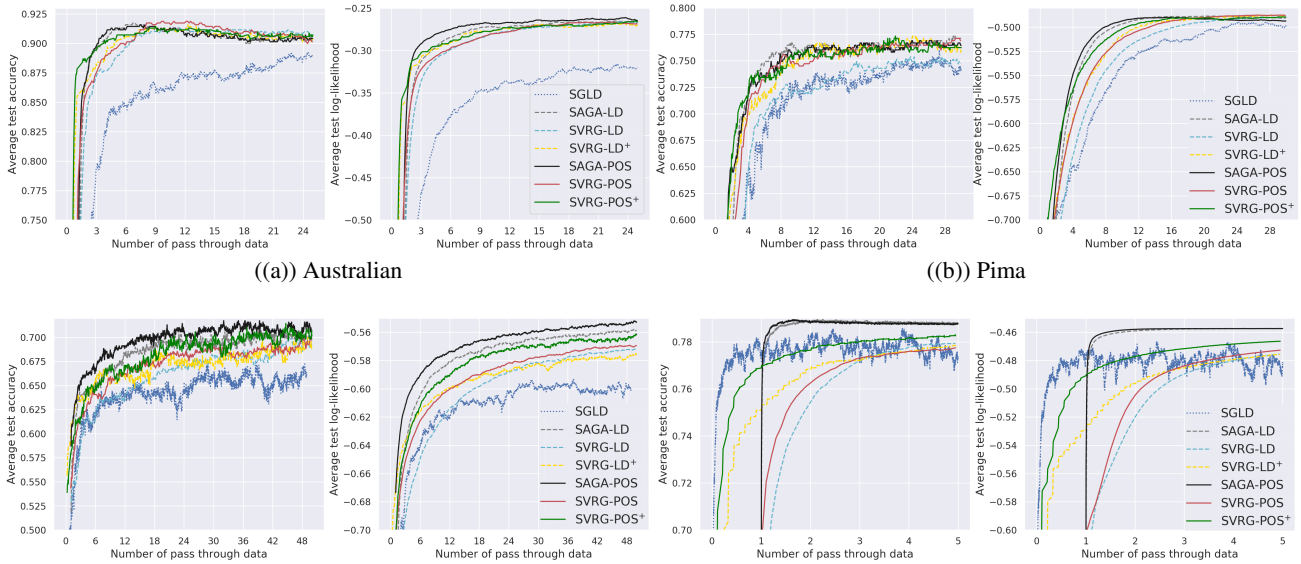
((d)) Susy

Figure 2. Testing accuracy and log-likelihood vs. the number of data pass for SPOS and its variance-reduction variants.

in Remarks 1 and 2. Interestingly, for the PIMA dataset, SVRG-LD is observed to perform even worse (converges slower) than standard SGLD. Furthermore, as discussed in Remark 4, our theory indicates that the improvement of SVRG-POS over SVRG-LD is more significant than that of SAGA-POS over SAGA-LD. This is indeed verified by inspecting the plots in Figure 3.

5.2.3. Impact of number of particles

Finally, we examine the impact of the number of particles on the convergence rates. As indicated by Theorems 1-3, for a fixed number of iterations T, the convergence error in terms



((a)) Australian ((b)) Pima

((c)) Diabetic ((d)) Susy

Figure 3. Testing accuracy and log-likelihood versus the number of dataset pass for variance-reduced SPOS and SGLD.

of 2-Wasserstein distance decreases with increasing number of particles. To verify this, we run SAGA-POS and SVRG-POS for BLR with the number of particles ranging among $\{1, 2, 4, 8, 16\}$. The test log-likelihoods versus iteration numbers are plotted in Figure 4. The results indeed are consistent with our theory.

6. Additional Theoretical Discussion for SAGA-POS, SVRG-POS and SVRG-POS$^+$

We discuss the mixing time and gradient complexity of our algorithms. The mixing time is the number of iterations needed to provably have error less than ε, measured in $\mathcal{W}_2$ distance (Chatterji et al., 2018). The gradient complexity (Zou et al., 2018), which is almost the same as the computational complexity in (Chatterji et al., 2018), is defined as the required number of stochastic gradient evaluations to achieve a target accuracy ε. In Table 1 we present the mixing time and gradient complexity of several related algorithms. We focus on Option I of SVRG-POS. Our results for SVRG-LD$^+$ and SVRG-POS$^+$ may be a little different from those reported in (Zou et al., 2018) since we adopt different definitions for F_j.

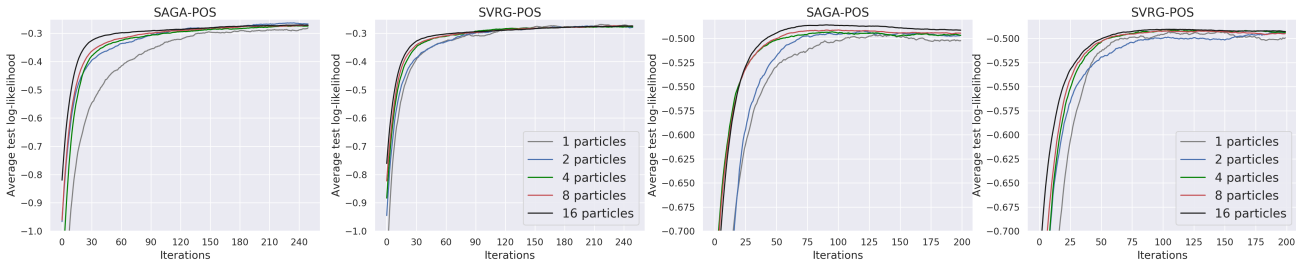
Note that the result for SVRG-POS$^+$ is derived by adopting $B = 1$ and $b = \mathcal{O}(d\sigma^2/\mu^2\varepsilon^2)$ from (Zou et al., 2018), which also sheds light on the optimal choice of b and B in our SVRG-POS$^+$. For fair comparisons with our algorithms, we consider variance-reduced versions of SGLD with M independent chains, which can be shown to have the same convergence rate in terms of total number of updates (Chen et al., 2015). Hence, the gradient complexities of the SAGA-LD, SVRG-LD and SVRG-LD$^+$ need to be scaled by M, consistent with the discussion in Section 3 and our experiment results. Since the convergence guarantees in Theorems 1, 2 and 3 are developed with respect to both iteration T and the number M, we define the "threshold-particle," which means the number of particles needed to provably have error less than ε measured in $\mathcal{W}_2$ distance. We use "threshold-particle" for our algorithms.

Note that the M in the mixing time from Table 1 also should satisfy the result that $M \geq C_1^2/\varepsilon^2$. In practice, since $C_1 = \frac{2(H_{\nabla K}+H_\theta)}{\sqrt{M}(\beta^{-1}-\frac{5}{2}H_\theta L_K - L_F - 2L_{\nabla K})}$, we set β to be small enough to avoid the threshold-particle to be too large. However, in our experiments, since H_F,L_K and L_F are not large, this issue seems to be less of a problem.

Finally, we give explanations concerning the SAGA-POS algorithm. From Algorithm 3.1, it may be noted that one must store all elements like $G_k^{(i)}$ in each iteration. It is known that $\{G_k^{(i)}\}_{i=1}^M$ only scales as $\mathcal{O}(Md)$. Taking the dataset *Susy* as an example, we have $N = 10000$ and $d = 18$. In practice, we only use $M \leq 40$ particles, resulting in a not-so-large $G_k^{(i)}$ term. Hence, we do not take $\{G_k^{(i)}\}_{i=1}^M$ into consideration.

7. Conclusions

We propose several variance-reduction techniques for stochastic particle-optimization sampling. For the first time, we develop nonasymptotic convergence theory for the algorithms in terms of 2-Wasserstein metrics. Our theoretical



((a)) Australian ((b)) Pima

Figure 4. Testing log-likelihood versus number of iterations with different number of particles for variance-reduced SPOS.

Table 1. Mixing Time and Gradient Complexity

Algorithm	**Mixing time**	**Gradient complexity**
SAGA-LD	$\mathcal{O}(\frac{(L_F/m_F)^{3/2}\sqrt{d}}{B\varepsilon})$	$\mathcal{O}(N+\frac{(L_F/m_F)^{3/2}\sqrt{d}}{\varepsilon})$
SAGA-POS	$\mathcal{O}(\frac{(C_2/C_3)^{3/2}\sqrt{d}}{BM^{\alpha}\varepsilon})$	$\mathcal{O}(NM+\frac{(C_2/C_3)^{3/2}\sqrt{d}M^{1-\alpha}}{\varepsilon})$
SVRG-LD	$\mathcal{O}(\frac{(L_F/m_F)^{3}d}{B\varepsilon^2})$	$\mathcal{O}(N+\frac{(L_F/m_F)^{3}\sqrt{d}}{\varepsilon^2})$
SVRG-POS	$\mathcal{O}(\frac{(C_2/C_3)^{3}d}{BM^{2\alpha}\varepsilon^2})$	$\mathcal{O}(NM+\frac{(C_2/C_3)^{3}\sqrt{d}M^{1-2\alpha}}{\varepsilon^2})$
SVRG-LD$^+$	$\mathcal{O}(\frac{\sigma^2 d}{{m_F}^2\varepsilon^2})$	$\mathcal{O}(\frac{\sigma^2 d}{{m_F}^2\varepsilon^2}\wedge(N+\frac{(L_F/m_F)^{3/2}\sqrt{d}}{\varepsilon}))$
SVRG-POS$^+$	$\mathcal{O}(\frac{C_4^2 d}{M^{2\alpha}{C_3}^2\varepsilon^2})$	$\mathcal{O}(\frac{C_4^2 d}{{C_3}^2\varepsilon^2}\wedge(NM+\frac{(C_2/C_3)^{3/2}\sqrt{d}M^{1-2\alpha}}{\varepsilon}))$

Table 2. Threshold-particle

Algorithm	Threshold-particle
SAGA-POS	C_1^2/ε^2
SVRG-POS	C_1^2/ε^2
SVRG-POS$^+$	C_1^2/ε^2

results indicate the improvement of convergence rates for the proposed variance-reduced SPOS compared to both standard SPOS and the variance-reduced SGLD algorithms. Our theory is verified by a number of experiments on both synthetic data and real data for Bayesian logistic regression. Leveraging both our theory and empirical findings, we recommend the following algorithm choices in practice: *i*) SAGA-POS is preferable when storage is not a concern; *ii*) SVRG-POS is a better choice when storage is a concern and full gradients are feasible to calculate; and *iii*) SVRG-POS$^+$ is a good choice and works well in practice when one faces with both computation and storage limitations.

Acknowledgements

The research performed at Duke University was supported in part by DARPA, DOE, NSF and ONR.

References

Chatterji, N., Flammarion, N., Ma, Y.-A., Bartlett, P., and Jordan, M. On the theory of variance reduction for stochastic gradient monte carlo. *ICML*, 2018.

Chen, C., Ding, N., and Carin, L. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *NIPS*, 2015.

Chen, C., Zhang, R., Wang, W., Li, B., and Chen, L. A unified particle-optimization framework for scalable Bayesian sampling. In *UAI*, 2018.

Chen, C., Wang, W., Zhang, Y., Su, Q., and Carin, L. A convergence analysis for a class of practical variance-reduction stochastic gradient MCMC. *Science China Information Sciences*, 62, 2019.

Chen, T., Fox, E. B., and Guestrin, C. Stochastic gradient Hamiltonian Monte Carlo. In *ICML*, 2014.

Şimşekli, U., Badeau, R., Cemgil, A. T., and Richard, G. Stochastic Quasi-Newton Langevin Monte Carlo. In *ICML*, 2016.

Dalalyan, A. S. and Karagulyan, A. User-friendly guarantees for the langevin monte carlo with inaccurate gradient. *arxiv preprint arxiv:1710.00095v2*, 2017.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Nips*, 2014.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. Bayesian sampling using stochastic gradient thermostats. In *NIPS*, 2014.

Dubey, A., Reddi, S. J., Póczos, B., Smola, A. J., and Xing, E. P. Variance reduction in stochastic gradient Langevin dynamics. In *NIPS*, 2016.

Durmus, A. and Moulines, E. High-dimensional bayesian inference via the unadjusted langevin algorithm. *arXiv preprint arXiv:1605.01559*, 2016.

Gan, Z., Chen, C., Henao, R., Carlson, D., and Carin, L. Scalable deep Poisson factor analysis for topic modeling. In *ICML*, 2015.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *ICML*, 2018.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *NIPS*, 2013.

Kolmogoroff, A. Some studies in machine learning using the game of checkers. *Mathematische Annalen*, 104(1): 415–458, 1931.

Li, B., Chen, C., Liu, H., and Carin, L. On connecting stochastic gradient MCMC and differential privacy. Technical Report arXiv:1712.09097, 2017. URL http://arxiv.org/abs/1712.09097.

Li, C., Chen, C., Carlson, D., and Carin, L. Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *AAAI*, 2016.

Liu, C., Zhu, J., and Song, Y. Stochastic gradient geodesic MCMC methods. In *NIPS*, 2016.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NIPS*, 2016.

Risken, H. *The Fokker-Planck equation*. Springer-Verlag, New York, 1989.

Soheil Feizi, Changho Suh, F. X. and Tse, D. Understanding gans: the lqg setting. *https://arxiv.org/abs/1710.10793*.

Springenberg, J. T., Klein, A., Falkner, S., and Hutter, F. Bayesian optimization with robust Bayesian neural networks. In *NIPS*, 2016.

Wang, Y. X., Fienberg, S. E., and Smola, A. Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In *ICML*, 2015.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *ICML*, 2011.

Zhang, J., Zhang, R., and Chen, C. Stochastic particle-optimization sampling and the non-asymptotic convergence theory. In *AISTATS*, 2020.

Zhang, R., Chen, C., Li, C., and Duke, L. C. Policy optimization as wasserstein gradient flows. In *ICML*, 2018a.

Zhang, R., Li, C., Chen, C., and Carin, L. Learning structural weight uncertainty for sequential decision-making. In *AISTATS*, 2018b.

Zhang, R., Wen, Z., Chen, C., and Carin, L. Scalable thompson sampling via optimal transport. In *AISTATS*, 2019.

Zou, D., Xu, P., and Gu, Q. Subsampled stochastic variance-reduced gradient langevin dynamics. *UAI*, 2018.

Zou, D., Xu, P., and Gu, Q. Stochastic gradient hamiltonian monte carlo methods with recursive variance reduction. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 3835–3846. 2019.