On the Theory of Variance Reduction for Stochastic Gradient Monte Carlo

Niladri S. Chatterji [1] **Nicolas Flammarion** [2] **Yi-An Ma** [2] **Peter L. Bartlett** [2 3] **Michael I. Jordan** [2 3]

Abstract

We provide convergence guarantees in Wasserstein distance for a variety of variance-reduction methods: SAGA Langevin diffusion, SVRG Langevin diffusion and control-variate underdamped Langevin diffusion. We analyze these methods under a uniform set of assumptions on the log-posterior distribution, assuming it to be smooth, strongly convex and Hessian Lipschitz. This is achieved by a new proof technique combining ideas from finite-sum optimization and the analysis of sampling methods. Our sharp theoretical bounds allow us to identify regimes of interest where each method performs better than the others. Our theory is verified with experiments on real-world and synthetic datasets.

1. Introduction

One of the major themes in machine learning is the use of stochasticity to obtain procedures that are computationally efficient and statistically calibrated. There are two very different ways in which this theme has played out—one frequentist and one Bayesian. On the frequentist side, gradient-based optimization procedures are widely used to obtain point estimates and point predictions, and stochasticity is used to bring down the computational cost by replacing expensive full-gradient computations with unbiased stochastic-gradient computations. On the Bayesian side, posterior distributions provide information about uncertainty in estimates and predictions, and stochasticity is used to represent those distributions in the form of Monte Carlo (MC) samples. Despite the different conceptual frameworks, there are overlapping methodological issues. In particular, Monte Carlo sampling must move from an out-of-equilibrium configuration towards the posterior distribution and must do so quickly, and thus optimization ideas are relevant. Frequentist inference often involves sampling and resampling, so that efficient approaches to Monte Carlo sampling are relevant.

Variance control has been a particularly interesting point of contact between the two frameworks. In particular, there is a subtlety in the use of stochastic gradients for optimization: Although the per-iteration cost is significantly lower by using stochastic gradients; extra variance is introduced into the sampling procedure at every step so that the total number of iterations is required to be larger. A natural question is whether there is a theoretically-sound way to manage this tradeoff. This question has been answered affirmatively in a seminal line of research (Schmidt et al., 2017; Shalev-Shwartz & Zhang, 2013; Johnson & Zhang, 2013) on variance-controlled stochastic optimization. Theoretically these methods enjoy the best of the gradient and stochastic gradient worlds—they converge at the fast rate of full gradient methods while making use of cheaply-computed stochastic gradients.

A parallel line of research has ensued on the Bayesian side in a Monte Carlo sampling framework. In particular, stochastic-gradient Markov chain Monte Carlo (SG-MCMC) algorithms have been proposed in which approximations to Langevin diffusions make use of stochastic gradients instead of full gradients (Welling & Teh, 2011). There have been a number of theoretical results that establish mixing time bounds for such Langevin-based sampling methods when the posterior distribution is well behaved (Dalalyan, 2017a; Durmus & Moulines, 2017; Cheng & Bartlett, 2017; Dalalyan & Karagulyan, 2017). Such results have set the stage for the investigation of variance control within the SG-MCMC framework (Dubey et al., 2016; Durmus et al., 2016; Bierkens et al., 2016; Baker et al., 2017; Nagapetyan et al., 2017; Chen et al., 2017). Currently, however, the results of these investigations are inconclusive. (Dubey et al., 2016) obtain mixing time guarantees for SAGA Langevin diffusion and SVRG Langevin diffusion (two particular variance-reduced sampling methods) under the strong assumption that the log-posterior has the norm of its gradients uniformly bounded by a constant. Another approach that has been explored involves calculating the mode of the log posterior to construct a control variate for the gradient estimate (Baker et al., 2017; Nagapetyan et al., 2017), an approach that makes rather different assumptions. Indeed, the experimental results from these two lines of work are contradictory,

[1]Department of Physics, [2]Division of Computer Science, [3]Department of Statistics, University of California Berkeley. Correspondence to: Niladri S. Chatterji <chatterji@berkeley.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

reflecting the differences in assumptions.

In this work we aim to provide a unified perspective on variance control for SG-MCMC. Critically, we identify two regimes: we show that when the target accuracy is small, variance-reduction methods are effective, but when the target accuracy is not small (a low-fidelity estimate of the posterior suffices), stochastic gradient Langevin diffusion (SGLD) performs better. These results are obtained via new theoretical techniques for studying stochastic gradient MC algorithms with variance reduction. We improve upon the techniques used to analyze Langevin Diffusion (LD) and SGLD (Dalalyan, 2017a; Dalalyan & Karagulyan, 2017; Durmus & Moulines, 2017) to establish non-asymptotic rates of convergence (in Wasserstein distance) for variance-reduced methods. We also apply control-variate techniques to underdamped Langevin MCMC (Cheng et al., 2017), a second-order diffusion process (CV-ULD). Inspired by proof techniques for variance-reduction methods for stochastic optimization, we design a Lyapunov function to track the progress of convergence and we thereby obtain better bounds on the convergence rate. We make the relatively weak assumption that the log posteriors are Lipschitz smooth, strongly convex and Hessian Lipschitz—a relaxation of the strong assumption that the gradient of the log posteriors are globally bounded.

As an example of our results, we are able to show that when using a variance-reduction method $\tilde{\mathcal{O}}(N+\sqrt{d}/\epsilon)$ steps are required to obtain an accuracy of ϵ, versus the $\tilde{\mathcal{O}}(d/\epsilon^2)$ iterations required for SGLD, where d is the dimension of the data and N is the total number of samples. As we will argue, results of this kind support our convention that when the target accuracy ϵ is *small*, variance-reduction methods outperform SGLD.

Main Contributions. We provide sharp convergence guarantees for a variety of variance-reduction methods—SAGA-LD, SVRG-LD, and CV-ULD under the *same set* of *realistic* assumptions (see Sec. 4). This is achieved by a new proof technique that yields bounds on Wasserstein distance. Our bounds allow us to identify windows of interest where each method performs better than the others (see Fig. 1). The theory is verified with experiments on real-world datasets. We also test the effects of breaking the central limit theorem using synthetic data, and find that in this regime variance-reduced methods fare far better than SGLD (see Sec. 5).

2. Preliminaries

Throughout the paper we aim to make inference on a vector of parameters $\theta \in \mathbb{R}^d$. The resulting posterior density is $p(\theta|\mathbf{z}) \propto p(\theta)\prod_{i=1}^N p(z_i|\theta)$. For brevity we write $f_i(\theta) = -\log(p(z_i|\theta))$, for $i \in \{1,\dots,N\}$, $f_0(x) = -\log(p(\theta))$ and $f(\theta) = -\log(p(\theta|\mathbf{z}))$. Moving forward we state all results in terms of general sum-decomposable functions f (see Assumption (A1)), however it is useful to keep the above example in mind as the main motivating example. We let $\|v\|_2$ denote the Euclidean norm, for a vector $v \in \mathbb{R}^d$. For a matrix A we let $\|A\|$ denote its spectral norm and let $\|A\|_F$ denote its Frobenius norm.

Assumptions on f: We make the following assumptions about the potential function $f : \mathbb{R}^d \mapsto \mathbb{R}$.

(A1) **Sum-decomposable:** The function f is decomposable, $f(x) = \sum_{i=1}^N f_i(x)$.

(A2) **Smoothness:** The functions f_i are twice continuously-differentiable on $\mathbb{R}^d$ and have Lipschitz-continuous gradients; that is, there exist positive constants $\tilde{M} > 0$ such that for all $x, y \in \mathbb{R}^d$ and for all $i \in \{1,\dots,N\}$ we have, $\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq \tilde{M}\|x-y\|_2$. We accordingly characterize the smoothness of f with the parameter $M := N\tilde{M}$.

(A3) **Strong Convexity:** f is m-strongly convex; that is, there exists a constant $m > 0$ such that for all $x, y \in \mathbb{R}^d$, $f(y) \geq f(x) + \langle \nabla f(x), y-x\rangle + \frac{m}{2}\|x-y\|_2^2$. We also define the condition number $\kappa := M/m$.

(A4) **Hessian Lipschitz:** The function f is Hessian Lipschitz, i.e., there exists a constant $L > 0$ such that, $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x-y\|_2$ for every $x, y \in \mathbb{R}^d$.

It is worth noting that M, m and L can all scale with N.

Wasserstein Distance: We define the Wasserstein distance between a pair of probability measures (μ, ν) as follows:

$$W_2^2(\mu,\nu) := \inf_{\zeta \in \Gamma(\mu,\nu)} \int \|x-y\|_2^2 d\zeta(x,y),$$

where $\Gamma(\mu,\nu)$ denotes the set of joint distributions such that the first set of coordinates has marginal μ and the second set has marginal ν. (See Appendix **??** for more details).

Langevin Diffusion: The classical overdamped Langevin diffusion is based on the following Itô Stochastic Differential Equation (SDE):

$$dx_t = -\nabla f(x_t)dt + \sqrt{2}dB_t, \tag{1}$$

where $x_t \in \mathbb{R}^d$ and B_t represents standard Brownian motion (see, e.g., Mörters & Peres, 2010). It can be shown that under mild conditions like $\exp(-f(x)) \in L^1$ (absolutely integrable) the invariant distribution of Eq. (1) is given by $p^*(x) \propto \exp(-f(x))$. This fact motivates the Langevin MCMC algorithm where given access to full gradients it is possible to efficiently simulate the discretization,

$$d\tilde{x}_t = -\nabla f(x_k)dt + \sqrt{2}dB_t, \tag{2}$$

where the gradient is evaluated at a fixed point x_k (the previous iterate in the chain) and the SDE (2) is integrated up to time δ (the step size) to obtain

$$x_{k+1} = x_k - \delta\nabla f(x_k) + \sqrt{2\delta}\xi_k,$$

with $\xi_k \sim N(0, I_{d\times d})$. (Welling & Teh, 2011) proposed an alternative algorithm—Stochastic Gradient Langevin Diffusion (SGLD)—for sampling from sum-decomposable functions where the chain is updated by integrating the SDE:

$$d\tilde{x}_t = -g_k dt + \sqrt{2} dB_t, \tag{3}$$

and where $g_k = \frac{N}{n}\sum_{i\in S}\nabla f_i(x_k)$ is an unbiased estimate of the gradient at x_k. The attractive property of this algorithm is that it is computationally tractable for large datasets (when N is large). At a high level the variance reduction schemes studied in this paper replace the simple gradient estimate in Eq. (3) (and other variants of Langevin MC) with more sophisticated unbiased estimates with lower variance.

3. Variance Reduction Techniques

In the seminal work of (Schmidt et al., 2017) and (Johnson & Zhang, 2013), it was observed that the variance of the unbiased estimate of the gradient used in Stochastic Gradient Descent (SGD) when applied to optimizing sum-decomposable strongly convex functions decreases to zero only if the step-size also decays at a suitable rate. This prevents the algorithm from converging at a linear rate, as opposed to methods like batch gradient descent that use the entire gradient at each step. They introduced and analyzed different gradient estimates with lower variance. Subsequently these methods were also adapted to Monte Carlo sampling by (Dubey et al., 2016; Nagapetyan et al., 2017; Baker et al., 2017). These methods use information from previous iterates and are no longer Markovian. In this section we describe several variants of these methods.

3.1. SAGA Langevin MC

We present a sampling algorithm based on SAGA of (Defazio et al., 2014) which was developed as a modification of SAG by (Schmidt et al., 2017). In SAGA, presented as Algorithm 1, an approximation of the gradient of each function f_i is stored as $\{g_k^i\}_{i=1}^N$ and is iteratively updated in order to build an estimate with reduced variance. At each step of the algorithm, if the function f_i is selected in the mini-batch S, then the value of the gradient approximation is updated by setting $g_{k+1}^i = \nabla f_i(x_k)$. Otherwise the gradient of f_i is approximated by the previous value g_k^i. Overall we obtain the following unbiased estimate of the gradient:

$$g_k = \sum_{i=1}^{n} g_k^i + \frac{N}{n}\sum_{i\in S}(\nabla f_i(x_k) - g_k^i). \tag{4}$$

In Algorithm 1 we form this gradient estimate and plug it into the classic Langevin MCMC method driven by the SDE (3). Computationally this algorithm is efficient; essentially it enjoys the oracle query complexity (number of calls to the gradient oracle per iteration) of methods like SGLD but due to the reduced variance of the gradient estimator it converges almost as quickly (in terms of number of iterations) to the posterior distribution as methods such as Langevin MCMC that use the complete gradient at every step. We prove a novel non-asymptotic convergence result in Wasserstein distance for Algorithm 1 in the next section that formalizes this intuition.

Algorithm 1 SAGA Langevin MCMC

Input: Gradient oracles $\{\nabla f_i(\cdot)\}_{i=0}^N$, step size δ, batch size n, initial point $x_0 \in \mathbb{R}^d$.
Initialize $\{g_0^i = \nabla f_i(x_0)\}_{i=1}^N$.
for $k = 1, \dots, T$ **do**
  Draw $S \subset \{0, \dots, N\} : |S| = n$ uniformly with replacement
  Sample $\xi_k \sim N(0, I_{d\times d})$
  Update g_k using (4)
  Update $x_{k+1} \leftarrow x_k - \delta g_k + \sqrt{2\delta}\xi_k$.
  Update $\{g_k^i\}_{i=1}^N$: for $i \in S$ set $g_{k+1}^i = \nabla f_i(x_k)$, for $i \in S^c$, set $g_{k+1}^i = g_k^i$
end for
Output: Iterates $\{x_k\}_{k=1}^T$.

The principal downside of this method is its memory requirement. It is necessary to store the gradient estimator for each individual f_i, which essentially means that in the worst case the memory complexity scales as $\mathcal{O}(Nd)$. However in many interesting applications, including some of those considered in the experiments in Sec. 5, the memory costs scale only as $\mathcal{O}(N)$ since each function f_i depends on a linear function in x and therefore the gradient ∇f_i is just a re-weighting of the single data point z_i.

3.2. SVRG Langevin MC

Next we explore an algorithm based on the SVRG method of Johnson & Zhang (2013) which takes its roots in work of Greensmith et al. (2004). The main idea behind SVRG is to build an auxiliary sequence $\tilde{x}$ where the full gradient is calculated and used as a reference in building a gradient estimate: $\nabla f_i(x) - \nabla f_i(\tilde{x}) + \nabla f(\tilde{x})$. This estimate is unbiased under the uniform choice of i. While using this gradient estimate to optimize sum-decomposable functions, the variance is small when x and $\tilde{x}$ are close to the optimum as $\nabla f(\tilde{x})$ is *small* and $\|\nabla f_i(x) - \nabla f_i(\tilde{x})\|$ is of the order $\|x - \tilde{x}\|_2$. We expect a similar behavior in the case of Monte Carlo sampling and we thus use this gradient estimate in Algorithm 2. Observe that crucially—unlike SAGA-based algorithms—this method does not require a gradient estimate of all of the individual f_i, so its memory cost scales as $\mathcal{O}(d)$. Algorithm 2 uses the unbiased gradient estimate

$$g_k = \tilde{g} + \frac{N}{n}\sum_{i\in S}\left[\nabla f_i(x_k) - \nabla f_i(\tilde{x})\right], \tag{5}$$

Algorithm 2 SVRG Langevin MCMC

Input: Gradient oracles $\{\nabla f_i(\cdot)\}_{i=0}^N$, step size δ, epoch length τ, batch size n, initial point $x_0 \in \mathbb{R}^d$.
Initialize $\tilde{x} \leftarrow x_0$, $\tilde{g} \leftarrow \sum_{i=1}^N \nabla f_i(x_0)$
for $k = 1, \dots, T$ **do**
  if $k \bmod \tau = 0$ **then**
    Option I: Sample $\ell \sim unif(0, 1, \dots, \tau - 1)$ and
    Update $\tilde{x} \leftarrow x_{k-\ell}$
    Update $x_k \leftarrow \tilde{x}$
    Option II: Update $\tilde{x} \leftarrow x_k$
    $\tilde{g} \leftarrow \sum_{i=1}^N \nabla f_i(x_k)$
  end if
  Draw $S \subset \{0, \dots, N\} : |S| = n$ uniformly with replacement
  Sample $\xi_k \sim N(0, I_{d\times d})$
  Update g_k using (5)
  Update $x_{k+1} \leftarrow x_k - \delta g + \sqrt{2\delta}\xi_k$.
end for
Output: Iterates $\{x_k\}_{k=1}^T$.

which uses a mini-batch of size n. The downside of this algorithm compared to SAGA however is that every few steps (an epoch) the full gradient, $\nabla f(\tilde{x})$, needs to be calculated at $\tilde{x}$. This results in the query complexity of each epoch being $\mathcal{O}(N)$. Also SVRG has an extra parameter that needs to be set—its hyperparameters are the epoch length (τ), the step size (δ) and the batch size (n), as opposed to just the step size and batch size for Algorithm 1 which makes it harder to tune. It also turns out that in practice, SVRG seems to be consistently outperformed by SAGA and control-variate techniques for sampling which is observed both in previous work and in our experiments.

3.3. Control Variates with Underdamped Langevin MC

Another approach is to use control variates (Ripley, 2009) to reduce the variance of stochastic gradients. This technique has also been previously explored both theoretically and experimentally by (Baker et al., 2017) and (Nagapetyan et al., 2017). Similar to SAGA and SVRG the idea is to build an unbiased estimate of the gradient $g(x)$ at a point x:

$$g(x) = \nabla f(\hat{x}) + \sum_{i \in S} \left[\nabla f_i(x) - \nabla f_i(\hat{x})\right],$$

where the set S is the mini-batch and $\hat{x}$ is a fixed point that is called the *centering value*. Observe that taking an expectation over the choice of the set S yields $\nabla f(x)$. A good centering value $\hat{x}$ would ensure that this estimate also has low variance; a natural choice in this regard is the *global minima* of f, x^*. A motivating example is the case of a Gaussian random variable where the mean of the distribution and x^* coincide.

A conclusion of previous work that applies control variate techniques to stochastic gradient Langevin MCMC is the following—the variance of the gradient estimates can be lowered to be of the order of the discretization error. Motivated by this, we apply these techniques to *underdamped* Langevin MCMC where the underlying continuous time diffusion process is given by the following second-order SDE:

$$\begin{aligned} dv_t &= -\gamma v_t dt - u\nabla f(x_t)dt + \sqrt{2}dB_t, \\ dx_t &= v_t dt, \end{aligned} \tag{6}$$

where $(x_t, v_t) \in \mathbb{R}^d$, B_t represents the standard Brownian motion and γ and u are constants. At a high level the advantage of using a second-order MCMC method like underdamped Langevin MCMC (Chen et al., 2014), or related methods like Hamiltonian Monte Carlo (see, e.g, Neal et al., 2011; Girolami & Calderhead, 2011), is that the discretization error is lower compared to overdamped Langevin MCMC. However when stochastic gradients are used (see (Chen et al., 2014; Ma et al., 2015) for implementation), this advantage can be lost as the variance of the gradient estimates dominates the total error. We thus apply control variate techniques to this second-order method. This reduces the variance of the gradient estimates to be of the order of the discretization error and enables us to recover faster rates of convergence. The discretization of SDE (6) (which we can simulate efficiently) is

$$\begin{aligned} d\tilde{v}_t &= -\gamma \tilde{v}_t dt - u\nabla \tilde{f}(x_k)dt + \sqrt{2}dB_t, \\ d\tilde{x}_t &= \tilde{v}_t dt, \end{aligned} \tag{7}$$

with initial conditions x_k, v_k (the previous iterate of the Markov Chain) and $\nabla \tilde{f}(x_k)$ is the estimate of the gradient at x_k, defined in (8). We integrate (7) for time δ (the step size) to get our next iterate of the chain—x_{k+1}, v_{k+1} for some $k \in \{1, \dots, T\}$. This MCMC procedure was introduced and analyzed by (Cheng et al., 2017) where they obtain that given access to full gradient oracles the chain converges in $T = \tilde{\mathcal{O}}(\sqrt{d}/\epsilon)$ steps (without Assumption (A4)) as opposed to standard Langevin diffusion which takes $T = \tilde{\mathcal{O}}(d/\epsilon)$ steps (with Assumption (A4)). With noisy gradients (variance $\sigma^2 d$), however, the mixing time of underdamped Langevin MCMC again degrades to $\tilde{\mathcal{O}}(\sigma^2 d/\epsilon^2)$.

Algorithm 3 CV Underdamped Langevin MCMC

Input: Gradient oracles $\{\nabla f_i(\cdot)\}_{i=0}^N$, step size δ, smoothness M, batch size n.
Set $x^* \in \operatorname{argmin}_{x\in\mathbb{R}^d} f(x)$.
Set $(x_0, v_0) \leftarrow (x^*, 0)$
for $k = 1, \dots, T$ **do**
  Draw a set $S \subset \{0, \dots N\}$ of size n u.a.r.
  Update $\nabla \tilde{f}(x_k)$ using (8)
  Sample $(x_{k+1}, v_{k+1}) \sim Z^{k+1}(x_k, v_k)$ defined in (**??**)
end for
Output: Iterates $\{x_k\}_{k=1}^T$.

In Algorithm 3 we use control variates to reduce variance and are able to provably recover the fast mixing time guarantee ($T = \tilde{\mathcal{O}}(\sqrt{d}/\epsilon)$) in Theorem 4.3. Algorithm 3 requires a pre-processing step of calculating the (approximate) minimum of f as opposed to Algorithm 1,2; however since f is strongly convex this pre-processing cost (using say SAGA for optimizing f with stochastic gradients) is small compared to the computational cost of the other steps.

In Algorithm 3 the updates of the gradients are dictated by,

$$\nabla \tilde{f}(x_k) = \nabla f(x^*) + \frac{N}{n} \sum_{i \in S} \left[\nabla f_i(x_k) - \nabla f_i(x^*) \right]. \quad (8)$$

The random vector that we draw, $Z^k(x_k, v_k) \in \mathbb{R}^{2d}$, conditioned on x_k, v_k, is a Gaussian vector with conditional mean and variance that can be *explicitly* calculated in closed form expression in terms of the algorithm parameters δ and M. Its expression is presented in Appendix **??**. Note that Z^k is a Gaussian vector and can be sampled in $\mathcal{O}(d)$ time.

4. Convergence results

In this section we provide convergence results of the algorithms presented above, which improve upon the convergence guarantees for SGLD. (Dalalyan & Karagulyan, 2017) show that for SGLD run for T iterations:

$$W_2(p^{(T)}, p^*) \leq \exp\left(-\delta m T\right) W_2(p^{(0)}, p^*) + \frac{\delta L d}{2m} + \frac{11 \delta M^{3/2} \sqrt{d}}{5m} + \frac{\sigma \sqrt{\delta d}}{2\sqrt{m}}, \quad (9)$$

under assumptions (A2)-(A4) with access to stochastic gradients with bounded variance – $\sigma^2 d$. The term involving the variance – $\sigma\sqrt{\delta d}/2\sqrt{m}$ dominates the others in many interesting regimes. For sum-decomposable functions that we are studying in this paper this is also the case as the variance of the gradient estimate usually scales linearly with N^2. Therefore the performance of SGLD sees a deterioration when compared to the convergence guarantees of Langevin Diffusion where $\sigma = 0$. To prove our convergence results we follow the general framework established by (Dalalyan & Karagulyan, 2017), with the noteworthy difference of working with more sophisticated Lyapunov functions (for Theorems 4.1 and 4.2) inspired by proof techniques in optimization theory. This contributes to strengthening the connection between optimization and sampling methods raised in previous work and may potentially be applied to other sampling algorithms (we elaborate on these connections in more detail in Appendix **??**). This comprehensive proof technique also allows us to sharpen the convergence guarantees obtained by (Dubey et al., 2016) on variance reduction methods like SAGA and SVRG by allowing us to present bounds in W_2 and to drop the assumption on requiring uniformly bounded gradients. In the theoretical analysis that follows we assume that the algorithms use a *fixed step-size* to simplify the statement of our results; similar results hold when we use a shrinking step-size. We now present convergence guarantees for Algorithm 1.

Theorem 4.1. *Let assumptions (A1)-(A4) hold. Let $p^{(T)}$ be the distribution of the iterate of Algorithm 1 after T steps. If we set the step size to be $\delta < \frac{n}{8MN}$ and the batch size $n \geq 9$ then we have the guarantee:*

$$W_2(p^{(T)}, p^*) \leq 5 \exp\left(-\frac{m\delta}{4} T\right) W_2(p^{(0)}, p^*) + \frac{2\delta L d}{m} + \frac{2\delta M^{3/2}\sqrt{d}}{m} + \frac{24\delta M \sqrt{dN}}{\sqrt{mn}}. \quad (10)$$

For the sake of clarity, only results for small step-size δ are presented however, it is worth noting that convergence guarantees hold for any $\delta \leq \frac{1}{8M}$ (see details in Appendix **??**). If we consider the regime where σ, M, L and m all scale linearly with the number of samples N, then for SGLD the dominating term is $\mathcal{O}(\sigma\sqrt{\delta d/m})$. If the target accuracy is ϵ, SGLD would require the step size to scale as $\mathcal{O}(\epsilon^2/d)$ while for SAGA a step size of $\delta = \mathcal{O}(\epsilon/d)$ is sufficient. The mixing time T for both methods is roughly proportional to the inverse step-size; thus SAGA provably takes fewer iterations while having almost the same computational complexity per step as SGLD. Similar to the optimization setting, theoretically SAGA Langevin diffusion recovers the *fast rate* of Langevin diffusion while just using cheap gradient updates. Next we present our guarantees for Algorithm 2.

Theorem 4.2. *Let assumptions (A1)-(A4) hold. Let $p^{(T)}$ be the distribution of the iterate of Algorithm 2 after T steps.*

If we set $\delta < \frac{1}{8M}$, $n \geq 2$, $\tau \geq \frac{8}{m\delta}$ and run Option I *then for all T mod $\tau = 0$ we have*

$$W_2(p^{(T)}, p^*) \leq \exp\left(-\frac{\delta m T}{56}\right) \frac{\sqrt{M}}{\sqrt{m}} W_2(p^{(0)}, p^*) + \frac{2\delta L d}{m} + \frac{2\delta M^{3/2}\sqrt{d}}{m} + \frac{64 M^{3/2}\sqrt{\delta d}}{m\sqrt{n}}. \quad (11)$$

If we set $\delta < \frac{\sqrt{n}}{4\tau M}$ and run Option II *for T iterations then,*

$$W_2(p^{(T)}, p^*) \leq \exp\left(-\frac{\delta m T}{4}\right) W_2(p^{(0)}, p^*) + \frac{\sqrt{2}\delta L d}{m} + \frac{5\delta M^{3/2}\sqrt{d}}{m} + \frac{9\delta M \tau \sqrt{d}}{\sqrt{mn}}. \quad (12)$$

For Option I, if we study the same regime as before where M, m and L are scaling linearly with N we find that the discretization error is dominated by the term which is of order $\mathcal{O}(\sqrt{\delta N d/n})$. To achieve target accuracy of ϵ we would need $\delta = \mathcal{O}(\epsilon^2 n/Nd)$. This is less impressive than the guarantees of SAGA and essentially we only gain a

constant factor as compared to the guarantees for SGLD. This behavior may be explained as follows: at each epoch, a constant decrease of the objective is needed in the classical proof of SVRG when applied to optimization. When the step-size is small, the epoch length is required to be large that washes away the advantages of variance reduction.

For Option II, similar convergence guarantees as SAGA are obtained, but worse by a factor of $\sqrt{n}$. In contrast to SAGA, this result holds only for small step-size, with the constants in Eq. (12) blowing up exponentially quickly for larger step sizes (for more details see proof in Appendix **??**). We also find that experimentally SAGA routinely outperforms SVRG both in terms of run-time and iteration complexity to achieve a desired target accuracy. However, it is not clear whether it is an artifact of our proof techniques that we could not recover matching bounds as SAGA or if SVRG is less suited to work with sampling methods. We now state our results for the convergence guarantees of Algorithm 3.

Theorem 4.3. *Let assumptions (A1)-(A3) hold. Let $p^{(T)}$ be the distribution of the iterate of Algorithm 3 after T steps starting with the initial distribution $p^{(0)}(x,v) = 1_{x=x^*} \cdot 1_{v=0}$. If we set the step size to be $\delta < 1/M$ and run Algorithm 3 then we have the guarantee that*

$$W_2(p^{(T)}, p^*) \leq 4 \exp\left(-\frac{m\delta T}{2}\right) W_2(p^{(0)}, p^*) + \frac{164\delta M^2\sqrt{d}}{m^{3/2}} + \frac{83M\sqrt{d}}{m^{3/2}\sqrt{n}}. \quad (13)$$

We initialize the chain in Algorithm 3 with x^*, the global minimizer of f as we already need to calculate it to build the gradient estimate. Observe that Theorem 4.3 does not guarantee the error drops to 0 when $\delta \to 0$ but is proportional to the standard deviation of our gradient estimate. This is in contrast to SAGA and SVRG based algorithms where a more involved gradient estimate is used. The advantage however of using this second order method is that we get to a desired error level ϵ at a faster rate as the step size can be chosen proportional to $\epsilon/\sqrt{d}$, which is $\sqrt{d}$ better than the corresponding results of Theorem 4.1 and 4.2 and without Assumption (A4) (Hessian Lipschitzness).

Note that by Lemma **??** we have the guarantee that $W_2(p^{(0)}, p^*) \leq 2d/m$; this motivates the choice of $\delta = \mathcal{O}\left(\frac{m\epsilon\sqrt{m}}{M^2\sqrt{d}}\right)$ and, $n = \mathcal{O}\left(\frac{M^2 d}{m^3\epsilon^2}\right)$ with $T = \tilde{\mathcal{O}}\left(1/(m\delta)\right)$. It is easy to check that under this choice of δ, n and T, Theorem 4.3 guarantees that $W_2(p^{(T)}, p^*) \leq \epsilon$. We note that no attempt has been made to optimize the constants. To interpret these results more carefully let us think of the case when M, m both scale linearly with the number of samples N. Here the number of steps $T = \tilde{\mathcal{O}}(\sqrt{d/(\epsilon^2 N)})$ and the batch size is $n = \mathcal{O}(d/(N\epsilon^2))$. If we compare it to previous results on control variate variance reduction techniques applied to *overdamped* Langevin MCMC by (Baker et al., 2017), the corresponding rates are $T = \tilde{\mathcal{O}}(d/(\epsilon^2 N))$ and $n = \mathcal{O}(d/(N\epsilon^2))$, essentially it is possible to get a quadratic improvement by using a second order method even in the presence of noisy gradients. Note however that these methods are not viable when the target accuracy ϵ is small as the batch size n needs to grow as $\mathcal{O}(1/\epsilon^2)$.

Table 1. Mixing time and (total) computational complexity comparison of Langevin sampling algorithms. All the entries are in Big-O notation which hides constants and poly-logarithmic factors. Note that the guarantees presented for ULD, SGULD, CV-LD and CV-ULD are without the Hessian Lipschitz assumption, (A4).

ALGORITHM	MIXING TIME	COMPUTATION
LD	$\kappa^2\sqrt{d}/(\sqrt{N}\epsilon)$	$\kappa^2\sqrt{dN}/\epsilon$
ULD	$\kappa^{\frac{5}{2}}\sqrt{d}/(\sqrt{N}\epsilon)$	$\kappa^{\frac{5}{2}}\sqrt{dN}/\epsilon$
SGLD	$\kappa^2 d/(n\epsilon^2)$	$\kappa^2 d/\epsilon^2$
SGULD	$\kappa^2 d/(n\epsilon^2)$	$\kappa^2 d/\epsilon^2$
SAGA-LD	$\kappa^{\frac{3}{2}}\sqrt{d}/(n\epsilon)$	$N+\kappa^{\frac{3}{2}}\sqrt{d}/\epsilon$
SVRG-LD (I)	$\kappa^3 d/(n\epsilon^2)$	$N+\kappa^3 d/\epsilon^2$
SVRG-LD (II)	$\kappa^{\frac{11}{6}}\sqrt{d}/(N^{\frac{2}{3}}\epsilon)$	$N+\kappa^{\frac{5}{3}}N^{\frac{1}{6}}\sqrt{d}/\epsilon$
CV-LD	$\kappa^3 d/(N\epsilon^2)$	$N+\kappa^6 d^2/(N^2\epsilon^4)$
CV-ULD	$\kappa^{\frac{5}{2}}\sqrt{d}/(\sqrt{N}\epsilon)$	$N+\kappa^{\frac{11}{2}} d^{\frac{3}{2}}/(N^{\frac{3}{2}}\epsilon^3)$

Comparison of Methods. Here we compare the theoretical guarantees of Langevin MC (LD, Durmus & Moulines, 2016), Underdamped Langevin MC (ULD, Cheng et al., 2017), SGLD (Dalalyan & Karagulyan, 2017), stochastic gradient underdamped Langevin diffusion (SGULD, Cheng et al., 2017), SAGA-LD (Algorithm 1), SVRG-LD (Algorithm 2 with Option I and II), Control Variate Langevin diffusion (CV-LD, Baker et al., 2017) and Control Variate underdamped Langevin diffusion (CV-ULD, Algorithm 3). We always consider the scenario where M, m and L are scaling linearly with N and where $N \gg d$ (*tall-data* regime). We note that the memory cost of all these algorithms except SAGA-LD is $\mathcal{O}(nd)$; for SAGA-LD the worst-case memory cost scales as $\mathcal{O}(Nd)$. Next we compare the mixing time (T), i.e., the number of steps needed to provably have error less than ϵ measured in W_2 and the computational complexity, which is the mixing time T times the query complexity per iteration. In the comparison below we focus on the dependence of the mixing time and computational complexity on the dimension d, number of samples N, condition number κ, and the target accuracy ϵ. The mini-batch size has no effect on the computational complexity of SGLD, SGULD and SAGA-LD; while for SVRG-LD, CV-LD and CV-ULD the mini-batch size is chosen to optimize the upper bound.

As illustrated in Fig. 1 we see qualitative differences in behaviors of variance reduced algorithms compared to SGLD. To calculate higher order statistics or computing confidence intervals for uncertainty quantification, it is imperative to calculate the posterior with high accuracy. In

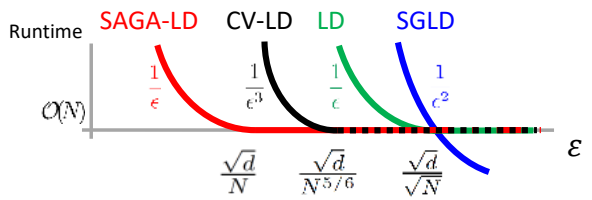

Figure 1. Different Regimes: The x-axis represents the target accuracy ϵ and the y-axis represents the predicted run-time T (number of queries to the gradient oracle) of different algorithms.

the regime where the desired accuracy $\epsilon < \mathcal{O}(\sqrt{d/N})$, total computation cost (runtime) of SGLD starts to grow larger than $\mathcal{O}(N)$ at rate $\mathcal{O}(\sqrt{d}/\epsilon^2)$ whereas runtime of variance reduced methods is lower. For SAGA-LD runtime is $\mathcal{O}(N)$ up until when $\epsilon = \mathcal{O}(\sqrt{d}/N)$ after which it grows at a rate $\mathcal{O}(\sqrt{d}/\epsilon)$. CV-ULD also has runtime of $\mathcal{O}(N)$ up until the point where $\epsilon = \mathcal{O}(\sqrt{d}/N^{5/6})$ after which it starts to grow as $\mathcal{O}(d^{3/2}/(N^{3/2}\epsilon^3))$. When $\mathcal{O}(\sqrt{d}/N^{5/6}) \leq \epsilon < \mathcal{O}(\sqrt{d}/\sqrt{N})$ our bounds predict both SAGA-LD and CV-ULD to have comparative performance ($\mathcal{O}(N)$) and in some scenarios one might outperform the other. For higher accuracy our results predict SAGA-LD performs better than CV-ULD. Note that Option II of SVRG performs also well in this regime of small ϵ but not as well as SAGA-LD or CV-ULD.

At the other end of the spectrum for most classical statistical problems accuracy of $\epsilon = \mathcal{O}(\sqrt{d/N})$ is sufficient and less than a single pass over the data is enough. In this regime when $\epsilon > \mathcal{O}(\sqrt{d/N})$ and we are looking to find a crude solution quickly, our bounds predict that SGLD is the fastest method. Other variance reduction methods need at least a single pass over the data to initialize.

Our sharp theoretical bounds allow us to classify and accurately identify regimes where the different variance reduction algorithms are efficient; bridging the gap between experimentally observed phenomenon and theoretical guarantees of previous works. Also noteworthy is that here we compare the algorithms only in the tall-data regime which grossly simplifies our results in Sec. 4, many other interesting regimes could be considered, for example the *fat-data* regime where $d \approx N$, but we omit this discussion here.

5. Experiments

In this section we explore the performance of SG-MCMC with variance reduction through experiments. As with most inference tasks on large datasets, we aim to infer model parameters for prediction. For this reason, the focus is in our experiements is the reduction of bias in our predictions through variance reduction algorithms. This is quantified by the log probability of the held-out dataset under the trained model. It is worth noting that this metric only reflects performance of the mean estimate and does not fully characterize the convergence of the distributions as our theoretical results do. We compare SAGA-LD, SVRG-LD (with Option II), CV-LD, CV-ULD with SGLD as the baseline method.

5.1. Bayesian Logistic Regression

We demonstrate results from sampling a Bayesian logistic regression model. We consider an $N \times d$ design matrix $\mathbf{X}$ comprised of N samples each with d covariates and a binary response variable $\mathbf{y} \in \{0,1\}^N$ (Gelman et al., 2004). If we denote the logistic link function by $s(\cdot)$, a Bayesian logistic regression model of the binary response with likelihood $P(\mathbf{y}_i = 1) = s(\beta^T \mathbf{X}_i)$ is obtained by introducing regression coefficients $\beta \in \mathbb{R}^d$ with a Gaussian prior $\beta \sim \mathcal{N}(0, \alpha I)$, where $\alpha = 1$ in the experiments. We make use of three datasets available at the UCI machine learning repository describing various connections between real valued attributes and categorical dependent variables. We use part of the datasets to obtain a mean estimate of the parameters and hold out the rest to test their likelihood under the estimated models. Sizes of the datasets being used in Bayesian estimation are 100, 600, and $1e5$, respectively.

Performance is measured by the log probability of the held-out dataset under the trained model. We first find the optimal log held-out probability attainable by all the currently methods being tested. We then try to obtain levels of log held-out probability increasingly closer to the optimal one with the other methods. We record number of passes through data that are required for each method to achieve the desired log held-out probability (averaged over 30 trials) for comparison in Fig. 2. The batch size is always $n = 10$, this is to explore whether the overall computational cost for SG-MCMC methods can grow sub-linearly with the overall size of the dataset N. A grid search is performed for the optimal hyperparameters in each algorithm, including an optimal scheduling plan of decreasing stepsizes. For CV-LD, we first use a stochastic gradient descent with SAGA variance reduction method to find the approximate mode x^*; we then calculate the full data gradient at x^* and initialize here.

From the experiments, we recover the three regimes displayed in Fig. 1 with different data size N and accuracy level with error ϵ. When N is large, SGLD performs best for big ϵ (c.f. left of PIMA and SUSY). When N is small, CV-LD/ULD is the fastest for relatively big ϵ (c.f. left of Heart). When N and ϵ are both small so that many passes through data are required, SAGA-LD is the most efficient method (c.f. right of Heart and PIMA). It is also clear from Fig. 2 that although CV-LD/ULD methods initially converges fast, there is a non-decreasing error (with the constant mini-batch size) even after the algorithm converges (corresponding to the last term in Eq. (13)). Because CV-LD and CV-ULD both converge fast and have the same non-decreasing error, their performance curves overlap. Con-

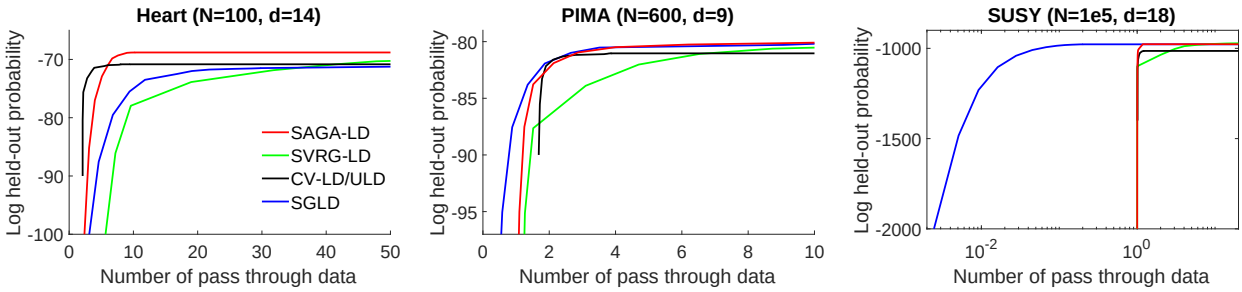

Figure 2. Number of passes through the datasets versus log held-out probability on test datasets.

vergence of SVRG-LD is slower than SAGA-LD, because the control variable for the stochastic gradient is only updated every epoch. This attribute combined with the need to compute the full gradient periodically makes it less efficient and costlier than SAGA-LD. We also see that number of passes through the dataset required for SG-MCMC methods (with and without variance reduction) is decreasing with the dataset size N. Close observation shows that although the overall computational cost is not constant with growing N, it is sublinear.

5.2. Breaking CLT: Synthetic Log Normal Data

Many works using SG-MCMC assume that the data in the mini-batches follow the central limit theorem (CLT) such that the stochastic gradient noise is Gaussian. But as explained by (Bardenet et al., 2017), if the dataset follows a long-tailed distribution, size of the mini-batch needed for CLT to take effect may exceed that of the entire dataset. We study the effects of breaking this CLT assumption on the behavior of SGLD and its variance reduction variants.

We use synthetic data generated from a log normal distribution: $f_X(x) = 1/(x\sigma\sqrt{2\pi}) \cdot \exp(-(\ln x - \mu)^2/(2\sigma^2))$ and sample the parameters μ and σ according to the likelihood $p(\mathbf{x}|\mu,\sigma) = \prod_{i=1}^{N} f_X(x_i)$. It is worth noting that this target distribution not only breaks the CLT for a wide range of mini-batch sizes, but also violates assumptions (A2)-(A4).

To see whether each method can perform well when the CLT assumption is violated, we still let mini-batch size to be 10 and grid search for the optimal hyperparameters for each method. We use mean squared error (MSE) as the convergence criteria and take LD as the baseline method.

From the experimental results, we see that SGLD does not converge to the target distribution. This is because most of the mini-batches only contain data close to the mode of the log normal distribution. Information about the tail is hard to capture with stochastic gradient. It can be seen that SAGA-LD and SVRG-LD are performing well because history information is recorded in the gradient so that data in the tail distribution is accounted for. Similar to the previous experiments, CV-LD converges fastest at first, but retains

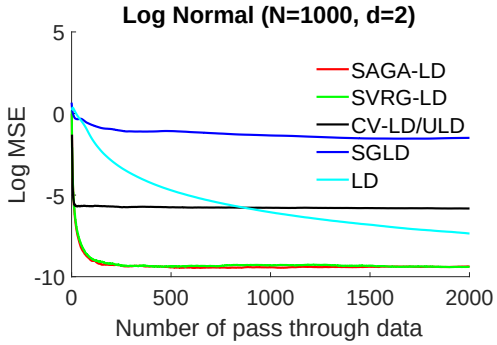

Figure 3. Number of passes through the datasets versus log mean square error (MSE).

a finite error. For LD, it converges to the same accuracy as SAGA-LD and SVRG-LD after 10^4 number of passes through data. The variance reduction methods which uses long term memory may be especially suited to this scenario, where data in the mini-batches violates the CLT assumption. It is also worth noting that the computation complexity for this problem is higher than our previous experiments. Number of passes through the entire dataset is on the order of $10^2 \sim 10^3$ to reach convergence even for SAGA-LD and SVRG-LD. It would be interesting to see whether non-uniform subsampling of the dataset (Schmidt et al., 2015) can accelerate the convergence of SG-MCMC even more.

6. Conclusions

In this paper, we derived new theoretical results for variance-reduced stochastic gradient MC. Our theory allows us to accurately classify two major regimes. When a low-accuracy solution is desired and less than one pass on the data is sufficient, SGLD should be preferred. When high accuracy is needed, variance-reduced methods are much more powerful. There are a number of further directions worth pursuing. It would be of interest to connect sampling with advances in finite-sum optimization, specifically advances in accelerated gradient descent (Lin et al., 2015) or single-pass methods (Lei & Jordan, 2017). Finally the development of a theory of lower bounds for sampling would be an essential counterpart to this work.

Acknowledgments

We gratefully acknowledge the support of the NSF through grant IIS-1619362 and the Army Research Office under grant number W911NF-17-1-0304.

References

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

Bardenet, R., Doucet, A., and Holmes, C. C. On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18(47):1–43, 2017.

Bierkens, J., Fearnhead, P., and Roberts, G. The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data. *arXiv preprint arXiv:1607.03188*, 2016.

Chen, C., Wang, W., Zhang, Y., Su, Q., and Carin, L. A convergence analysis for a class of practical variance-reduction stochastic gradient MCMC. *arXiv preprint arXiv:1709.01180*, 2017.

Chen, T., Fox, E. B., and Guestrin, C. Stochastic gradient Hamiltonian Monte Carlo. In *Proceeding of 31st International Conference on Machine Learning (ICML'14)*, 2014.

Cheng, X. and Bartlett, P. L. Convergence of Langevin MCMC in KL-divergence. *arXiv preprint arXiv:1705.09048*, 2017.

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

Clark, D. S. Short proof of a discrete Gronwall inequality. *Discrete applied mathematics*, 16(3):279–281, 1987.

Dalalyan, A. Theoretical guarantees for approximate sampling from a smooth and log-concave density. *J. R. Stat. Soc. B*, 79:651–676, 2017a.

Dalalyan, A. Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent. In *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pp. 678–689. PMLR, 07–10 Jul 2017b.

Dalalyan, A. and Karagulyan, A. User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. *arXiv preprint arXiv:1710.00095*, 2017.

Defazio, A. A simple practical accelerated method for finite sums. In *Advances in Neural Information Processing Systems 29*, pp. 676–684, 2016.

Defazio, A., Bach, F., and Lacoste-Julien, S. SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27*, pp. 1646–1654, 2014.

Dubey, K. A., Reddi, S. J., Williamson, S. A., Poczos, B., Smola, A. J., and Xing, E. P. Variance reduction in stochastic gradient Langevin dynamics. In *Advances in Neural Information Processing Systems 29*, pp. 1154–1162, 2016.

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

Durmus, A. and Moulines, E. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *Ann. Appl. Probab.*, 27(3):1551–1587, 06 2017.

Durmus, A., Simsekli, U., Moulines, E., Badeau, R., and Richard, G. Stochastic gradient Richardson-Romberg Markov chain Monte Carlo. In *Advances in Neural Information Processing Systems 29*, pp. 2047–2055, 2016.

Gelman, A., Carhn, J. B., Stern, H. S., and Rubin, D. B. *Bayesian Data Analysis*. Chapman and Hall, 2004.

Girolami, M. and Calderhead, B. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 73(2):123–214, 2011.

Greensmith, E., Bartlett, P. L, and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov): 1471–1530, 2004.

Hofmann, T., Lucchi, A., Lacoste-Julien, S., and McWilliams, B. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems 28*, pp. 2305–2313, 2015.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pp. 315–323, 2013.

Lei, L. and Jordan, M. I. Less than a single pass: stochastically controlled stochastic gradient. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 148–156. PMLR, 20–22 Apr 2017.

Lin, H., Mairal, J., and Harchaoui, Z. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems 28*, pp. 3384–3392. 2015.

Ma, Y.-A, Chen, T., and Fox, E. B. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems 28*, pp. 2899–2907. 2015.

Mörters, P. and Peres, Y. *Brownian Motion*, volume 30. Cambridge University Press, 2010.

Nagapetyan, T., Duncan, A. B, Hasenclever, L., Vollmer, S. J, Szpruch, L., and Zygalakis, K. The true cost of stochastic gradient Langevin dynamics. *arXiv preprint arXiv:1706.02692*, 2017.

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

Nesterov, Yurii. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Pavliotis, G. A. *Stochastic Processes and Applications*. Springer, 2016.

Ripley, B. D. *Stochastic Simulation*, volume 316. John Wiley & Sons, 2009.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3): 400–407, 09 1951.

Schmidt, M., Babanezhad, R., Ahmed, M. O., Defazio, A., Clifton, A., and Sarkar, A. Non-uniform stochastic average gradient method for training conditional random fields. In *18th International Conference on Artificial Intelligence and Statistics*, 2015.

Schmidt, M., Le Roux, N., and Bach, F. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Shalev-Shwartz, S. and Zhang, T. Stochastic dual coordinate ascent methods for regularized loss minimization. *J. Mach. Learn. Res.*, 14:567–599, 2013.

Villani, C. *Optimal Transport: Old and New*. Springer Science and Business Media, 2008.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 681–688, June 2011.