
Almost Tune-Free Variance Reduction

Bingcong Li¹ Lingda Wang² Georgios B. Giannakis¹

Abstract

The variance reduction class of algorithms including the representative ones, SVRG and SARAH, have well documented merits for empirical risk minimization problems. However, they require grid search to tune parameters (step size and the number of iterations per inner loop) for optimal performance. This work introduces ‘almost tune-free’ SVRG and SARAH schemes equipped with i) Barzilai-Borwein (BB) step sizes; ii) averaging; and, iii) the inner loop length adjusted to the BB step sizes. In particular, SVRG, SARAH, and their BB variants are first reexamined through an ‘estimate sequence’ lens to enable new averaging methods that tighten their convergence rates theoretically, and improve their performance empirically when the step size or the inner loop length is chosen large. Then a simple yet effective means to adjust the number of iterations per inner loop is developed to enhance the merits of the proposed averaging schemes and BB step sizes. Numerical tests corroborate the proposed methods.

1. Introduction

In this work, we deal with the frequently encountered empirical risk minimization (ERM) problem expressed as

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) := \frac{1}{n} \sum_{i \in [n]} f_i(\mathbf{x}) \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^d$ is the parameter vector to be learned from data; the set $[n] := \{1, 2, \dots, n\}$ collects data indices; and, f_i is the loss function associated with datum i . Suppose that f is μ -strongly convex and has L -Lipchitz continuous gradient. The condition number of f is denoted by $\kappa := L/\mu$. Throughout, \mathbf{x}^* denotes the optimal solution of (1).

¹University of Minnesota, MN, USA. ²University of Illinois at Urbana-Champaign, IL, USA. Correspondence to: Bingcong Li <lix5599@umn.edu>.

The standard approach to solve (1) is *gradient descent* (GD) (Nesterov, 2004), which updates the decision variable via

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k)$$

where k is the iteration index and η the step size (or learning rate). For a strongly convex f , GD converges linearly to \mathbf{x}^* , that is, $\|\mathbf{x}_k - \mathbf{x}^*\|^2 \leq (c_\kappa)^k \|\mathbf{x}_0 - \mathbf{x}^*\|^2$ for some κ -dependent constant $c_\kappa \in (0, 1)$ (Nesterov, 2004).

In the big data regime, however, where n is large, obtaining the gradient per iteration can be computationally prohibitive. To cope with this, the *stochastic gradient descent* (SGD) reduces the computational burden by drawing uniformly at random an index $i_k \in [n]$ per iteration k , and adopting $\nabla f_{i_k}(\mathbf{x}_k)$ as an estimate of $\nabla f(\mathbf{x}_k)$. Albeit computationally lightweight with the simple update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \nabla f_{i_k}(\mathbf{x}_k)$$

the price paid is that SGD comes with sublinear convergence, hence slower than GD (Robbins and Monro, 1951; Bottou et al., 2016). It has been long recognized that the variance $\mathbb{E}[\|\nabla f_{i_t}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)\|^2]$ of the gradient estimate affects critically SGD’s convergence slowdown.

This naturally motivated gradient estimates with *reduced variance* compared with SGD’s simple $\nabla f_{i_k}(\mathbf{x}_k)$. A gradient estimate with reduced variance can be obtained by capitalizing on the finite sum structure of (1). One idea is to judiciously evaluate a so-termed *snapshot gradient* $\nabla f(\mathbf{x}_s)$, and use it as an anchor of the stochastic draws in subsequent iterations. Members of the variance reduction family include SVRG (Johnson and Zhang, 2013), SAG (Roux et al., 2012), SAGA (Defazio et al., 2014), MISO (Mairal, 2013), SARAH (Nguyen et al., 2017), and their variants (Konecny and Richtárik, 2013; Lei et al., 2017; Kovalev et al., 2019; Li et al., 2020). Most of these algorithms rely on the update $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \mathbf{v}_k$, where η is a constant step size and \mathbf{v}_k is an algorithm-specific gradient estimate that takes advantage of the snapshot gradient. In this work, SVRG and SARAH are of central interest because they are memory efficient compared with SAGA, and have no requirement for the duality arguments that SDCA (Shalev-Shwartz and Zhang, 2013) entails. Variance reduction methods converge linearly when f is strongly convex. To fairly compare the complexity of (S)GD with that of variance reduction algorithms which combine snapshot gradients with the stochastic

ones, we will rely on the incremental first-order oracle (IFO) (Agarwal and Bottou, 2015).

Definition 1. An IFO takes f_i and $\mathbf{x} \in \mathbb{R}^d$ as input, and returns the (incremental) gradient $\nabla f_i(\mathbf{x})$.

For convenience, IFO complexity is abbreviated as complexity in this work. A desirable algorithm obtains an ϵ -accurate solution satisfying $\mathbb{E}[\|\nabla f(\mathbf{x})\|^2] \leq \epsilon$ or $\mathbb{E}[f(\mathbf{x}) - f(\mathbf{x}^*)] \leq \epsilon$ with minimal complexity for a prescribed ϵ . Complexity for variance reduction alternatives such as SVRG and SARAH is $\mathcal{O}\left((n + \kappa) \ln \frac{1}{\epsilon}\right)$, a clear improvement over GD’s complexity $\mathcal{O}\left(n\kappa \ln \frac{1}{\epsilon}\right)$. And when high accuracy (small ϵ) is desired, the complexity of variance reduction algorithms is also lower than SGD’s complexity of $\mathcal{O}\left(\frac{1}{\epsilon}\right)$. The merits of gradient estimates with reduced variance go beyond convexity; see e.g., (Reddi et al., 2016; Fang et al., 2018; Cutkosky and Orabona, 2019), but nonconvex ERM are out of the present work’s scope.

Though theoretically appealing, SVRG and SARAH entail grid search to tune the step size, which is often painstakingly hard and time consuming. An automatically tuned step size for SVRG was introduced by (Barzilai and Borwein, 1988) (BB) and (Tan et al., 2016). However, since both SVRG and SARAH have a double-loop structure, the inner loop length also requires tuning in addition to the step size. Other works relying on BB step sizes introduce additional tunable parameters on top of the inner loop length (Liu et al.; Yang et al., 2019). In a nutshell, ‘tune-free’ variance reduction algorithms still have desired aspects to investigate and fulfill.

Along with the BB step sizes, this paper establishes that in order to obtain ‘tune-free’ SVRG and SARAH schemes, one must: i) develop novel types of averaging; and, ii) adjust the inner loop length along with step size as well. Averaging in double-loop algorithms reflects the means of choosing the starting point of the next outer loop (Johnson and Zhang, 2013; Tan et al., 2016; Nguyen et al., 2017). The types of averaging considered so far have been employed as tricks to simplify proofs, while in the algorithm itself the last iteration is the most prevalent choice for the starting point of the ensuing outer loop. However, we contend that different averaging methods result in different performance. And the best averaging depends on the choice of other parameters. In addition to averaging, we argue that the choice of the inner loop length for BB-SVRG in (Tan et al., 2016) is too pessimistic. Addressing this with a simple modification leads to the desired ‘almost tune-free’ SVRG and SARAH.

Our detailed contributions can be summarized as follows.

- We empirically argue that averaging is not merely a proof trick. It is prudent to adjust averaging in accordance with the step size and the inner loop length.
- SVRG and SARAH are analyzed using the notion of

estimate sequence (ES). This prompts a novel averaging that tightens up convergence rate for SVRG, and further improves SARAH’s convergence over existing works under certain conditions. Besides tighter rates, our analysis broadens the analytical tool, ES, by endowing it with the ability to deal with SARAH’s biased gradient estimate.

- The theoretical guarantees for BB-SVRG and BB-SARAH with different types of averaging are established and leveraged for performance improvement.
- Finally, we offer a principled design of the inner loop length to obtain almost tune-free BB-SVRG and BB-SARAH. The choice for the inner loop length is guided by the regime that the proposed averaging schemes favor. Numerical tests further corroborate the efficiency of the proposed algorithms.

Notation. Bold lowercase letters denote column vectors; \mathbb{E} represents expectation; $\|\mathbf{x}\|$ stands for the ℓ_2 -norm of \mathbf{x} ; and $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes the inner product of vectors \mathbf{x} and \mathbf{y} .

2. Preliminaries

We will first focus on the averaging techniques, whose generality goes beyond BB step sizes. To start with, this section briefly reviews the vanilla SVRG and SARAH, while their BB variants are postponed slightly.

2.1. Basic Assumptions

Assumption 1. Each $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ has L -Lipchitz gradient, that is, $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

Assumption 2. Each $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex.

Assumption 3. Function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ -strongly convex, that is, there exists $\mu > 0$, such that $f(\mathbf{x}) - f(\mathbf{y}) \geq \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

Assumption 4. Each $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ -strongly convex, meaning there exists $\mu > 0$, so that $f_i(\mathbf{x}) - f_i(\mathbf{y}) \geq \langle \nabla f_i(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

Assumption 1 requires each loss function to be sufficiently smooth. One can certainly require smoothness of each individual loss function and refine Assumption 1 as f_i has L_i -Lipchitz gradient. Clearly $L = \max_i L_i$. By combining with importance sampling (Xiao and Zhang, 2014; Kulunchakov and Mairal, 2019), such a refined assumption can slightly tighten the κ dependence in the complexity bound. However, since the extension is straightforward, we will keep using the simpler Assumption 1 for clarity. Assumption 3 only requires f to be strongly convex, which is weaker than Assumption 4. Assumptions 1 – 4 are all standard in variance reduction algorithms.

Algorithm 1 SVRG

```

1: Initialize:  $\tilde{\mathbf{x}}^0, \eta, m, S$ 
2: for  $s = 1, 2, \dots, S$  do
3:    $\mathbf{x}_0^s = \tilde{\mathbf{x}}^{s-1}$ 
4:    $\mathbf{g}^s = \nabla f(\mathbf{x}_0^s)$ 
5:   for  $k = 0, 1, \dots, m-1$  do
6:     uniformly draw  $i_k \in [n]$ 
7:      $\mathbf{v}_k^s = \nabla f_{i_k}(\mathbf{x}_k^s) - \nabla f_{i_k}(\mathbf{x}_0^s) + \mathbf{g}^s$ 
8:      $\mathbf{x}_{k+1}^s = \mathbf{x}_k^s - \eta \mathbf{v}_k^s$ 
9:   end for
10:  draw  $\tilde{\mathbf{x}}^s$  randomly from  $\{\mathbf{x}_k^s\}_{k=0}^m$  according to  $\mathbf{p}^s$ 
11: end for
12: Output:  $\tilde{\mathbf{x}}^S$ 

```

Algorithm 2 SARAH

```

1: Initialize:  $\tilde{\mathbf{x}}^0, \eta, m, S$ 
2: for  $s = 1, 2, \dots, S$  do
3:    $\mathbf{x}_0^s = \tilde{\mathbf{x}}^{s-1}$ , and  $\mathbf{v}_0^s = \nabla f(\mathbf{x}_0^s)$ 
4:    $\mathbf{x}_1^s = \mathbf{x}_0^s - \eta \mathbf{v}_0^s$ 
5:   for  $k = 1, 2, \dots, m-1$  do
6:     uniformly draw  $i_k \in [n]$ 
7:      $\mathbf{v}_k^s = \nabla f_{i_k}(\mathbf{x}_k^s) - \nabla f_{i_k}(\mathbf{x}_{k-1}^s) + \mathbf{v}_{k-1}^s$ 
8:      $\mathbf{x}_{k+1}^s = \mathbf{x}_k^s - \eta \mathbf{v}_k^s$ 
9:   end for
10:  draw  $\tilde{\mathbf{x}}^s$  randomly from  $\{\mathbf{x}_k^s\}_{k=0}^m$  according to  $\mathbf{p}^s$ 
11: end for
12: Output:  $\tilde{\mathbf{x}}^S$ 

```

2.2. Recap of SVRG and SARAH

The steps of SVRG and SARAH are listed in Algs. 1 and 2, respectively. Each employs a fine-grained reduced-variance gradient estimate per iteration. For SVRG, \mathbf{v}_k^s is an unbiased estimate since $\mathbb{E}[\mathbf{v}_k^s | \mathcal{F}_{k-1}^s] = \nabla f(\mathbf{x}_k^s)$, where $\mathcal{F}_{k-1}^s := \sigma(\tilde{\mathbf{x}}^{s-1}, i_0, i_1, \dots, i_{k-1})$ is the σ -algebra generated by $\tilde{\mathbf{x}}^{s-1}, i_1, i_2, \dots, i_{k-1}$; while SARAH adopts a biased \mathbf{v}_k^s , that is, $\mathbb{E}[\mathbf{v}_k^s | \mathcal{F}_{k-1}^s] = \nabla f(\mathbf{x}_k^s) - \nabla f(\mathbf{x}_{k-1}^s) + \mathbf{v}_{k-1}^s \neq \nabla f(\mathbf{x}_k^s)$. The variance (mean-square error (MSE)) of \mathbf{v}_k^s in SVRG (SARAH) can be upper bounded by quantities that dictate the optimality gap (gradient norm square).

Lemma 1. (Johnson and Zhang, 2013; Nguyen et al., 2017) *The MSE of \mathbf{v}_k^s in SVRG is bounded as follows*

$$\begin{aligned} \text{SVRG} : \mathbb{E}[\|\nabla f(\mathbf{x}_k^s) - \mathbf{v}_k^s\|^2] &\leq \mathbb{E}[\|\mathbf{v}_k^s\|^2] \\ &\leq 4L\mathbb{E}[f(\mathbf{x}_k^s) - f(\mathbf{x}^*)] + 4L\mathbb{E}[f(\mathbf{x}_0^s) - f(\mathbf{x}^*)]. \end{aligned} \quad (2a)$$

The MSE of \mathbf{v}_k^s in SARAH is also bounded as

$$\begin{aligned} \text{SARAH} : \mathbb{E}[\|\nabla f(\mathbf{x}_k^s) - \mathbf{v}_k^s\|^2] & \\ &\leq \frac{\eta L}{2 - \eta L} \left(\mathbb{E}[\|\nabla f(\mathbf{x}_0^s)\|^2] - \mathbb{E}[\|\mathbf{v}_k^s\|^2] \right). \end{aligned} \quad (2b)$$

Another upper bound on SVRG's gradient estimate is available; see e.g., (Kulunchakov and Mairal, 2019), but it is not

suitable for our analysis. Intuitively, Lemma 1 suggests that if SVRG or SARAH converges, the MSE of their gradient estimates also approaches to zero.

At the end of each inner loop, the starting point of the next outer loop is randomly selected among $\{\mathbf{x}_k^s\}_{k=0}^m$ according to a pmf vector $\mathbf{p}^s \in \Delta_{m+1}$, where $\Delta_{m+1} := \{\mathbf{p} \in \mathbb{R}_+^{m+1} | \langle \mathbf{1}, \mathbf{p} \rangle = 1\}$. We term \mathbf{p}^s the *averaging weight vector*, and let p_j^s denote the j th entry of \mathbf{p}^s . Leveraging the MSE bounds in Lemma 1 and choosing a proper averaging vector, SVRG and SARAH iterates for strongly convex problems can be proved to converge linearly.

For SVRG, two types of averaging exist.

- **U-Avg (SVRG)** (Johnson and Zhang, 2013): vector \mathbf{p}^s is chosen as the pmf of an (almost) uniform distribution, that is, $p_m^s = 0$, and $p_k^s = 1/m$ for $k = \{0, 1, \dots, m-1\}$. Under Assumptions 1–3, the choice of $\eta = \mathcal{O}(1/L)$ and $m = \mathcal{O}(\kappa)$ ensures that SVRG iterates converge linearly.¹
- **L-Avg (SVRG)** (Tan et al., 2016; Hu et al., 2018): Only the last iteration is used for averaging by setting $\tilde{\mathbf{x}}^s = \mathbf{x}_m^s$; or equivalently, by setting $p_m^s = 1$, and $p_k^s = 0, \forall k \neq m$. Under Assumptions 1–3, linear convergence is ensured by choosing $\eta = \mathcal{O}(1/(L\kappa))$ and $m = \mathcal{O}(\kappa^2)$.

To guarantee linear convergence, SVRG with L-Avg must adopt a much smaller η and larger m compared with U-Avg. L-Avg with such a small step size leads to complexity $\mathcal{O}\left((n + \kappa^2) \ln \frac{1}{\epsilon}\right)$ that has worse dependence on κ .

For SARAH, there are also two averaging options.

- **U-Avg (SARAH)** (Nguyen et al., 2017): here \mathbf{p}^s is selected to have entries $p_m^s = 0$, and $p_k^s = 1/m$, for $k = \{0, 1, \dots, m-1\}$. Linear convergence is guaranteed with complexity $\mathcal{O}\left((n + \kappa) \ln \frac{1}{\epsilon}\right)$ under Assumptions 1–3 so long as one selects $\eta = \mathcal{O}(1/L)$ and $m = \mathcal{O}(\kappa)$.
- **L-Avg (SARAH)** (Li et al., 2020)²: here \mathbf{p}^s is chosen with entries $p_{m-1}^s = 1$ and $p_k^s = 0, \forall k \neq m-1$. Under Assumptions 1–3 and with $\eta = \mathcal{O}(1/L)$ as well as $m = \mathcal{O}(\kappa^2)$, linear convergence is guaranteed at complexity of $\mathcal{O}\left((n + \kappa^2) \ln \frac{1}{\epsilon}\right)$. When both Assumptions 1 and 4 hold, setting $\eta = \mathcal{O}(1/L)$ and $m = \mathcal{O}(\kappa)$ results in linear convergence along with a reduced complexity of order $\mathcal{O}\left((n + \kappa) \ln \frac{1}{\epsilon}\right)$.

¹For simplicity and clarity of exposition we only highlight the order of η and m , and hide other constants in big- \mathcal{O} notation. Detailed choices can be found in the corresponding references.

²There is another version of L-Avg for SARAH (Liu et al.), but convergence claims require undesirably small step sizes $\eta = \mathcal{O}(\mu/L^2)$. This is why we focus on the L-Avg in (Li et al., 2020).

U-Avg (for both SVRG and SARAH) is usually employed as a ‘proof-trick’ to carry out convergence analysis, while L-Avg is implemented most of the times. However, we will argue in the next section that with U-Avg adapted to the step size choice, it is possible to improve empirical performance. Although U-Avg appears at first glance to waste updates, a simple trick in the implementation can fix this issue.

Implementation of Averaging. Rather than updating m times and then choosing $\tilde{\mathbf{x}}^s$ according to Line 10 of SVRG or SARAH, one can generate a random integer $M^s \in \{0, 1, \dots, m\}$ according to the averaging weight vector \mathbf{p}^s . Having available $\mathbf{x}_{M^s}^s$, it is possible to start the next outer loop immediately.

3. Weighted Averaging for SVRG and SARAH

This section introduces weighted averaging for SVRG and SARAH, which serves as an intermediate step for the ultimate ‘tune-free variance reduction.’ Such an averaging for SVRG will considerably tighten its analytical convergence rate; while for SARAH it will improve its convergence rate when m or η is chosen sufficiently large. These analytical results are obtained by reexamining SVRG and SARAH through the ‘estimate sequence’ (ES), a tool that has been used for analyzing momentum schemes (Nesterov, 2004); see also (Nitanda, 2014; Lin et al., 2015; Kulunchakov and Mairal, 2019). Different from existing ES analysis that relies heavily on the unbiasedness of \mathbf{v}_k^s , our advances here will endow ES with the ability to deal with the biased gradient estimate of SARAH.

3.1. Estimate Sequence

Since in this section we will focus on a specific inner loop indexed by s , the superscript s is dropped for brevity. For example, \mathbf{x}_k^s and \mathbf{v}_k^s are written as \mathbf{x}_k and \mathbf{v}_k , respectively.

Associated with the ERM objective f and a particular point \mathbf{x}_0 , consider a series of quadratic functions $\{\Phi_k(\mathbf{x})\}_{k=0}^m$ that comprise what is termed ES, with the first one given by

$$\Phi_0(\mathbf{x}) = \Phi_0^* + \frac{\mu_0}{2} \|\mathbf{x} - \mathbf{x}_0\|^2 \quad (3a)$$

and the rest defined recursively as

$$\begin{aligned} \Phi_k(\mathbf{x}) = & (1 - \delta_k)\Phi_{k-1}(\mathbf{x}) + \delta_k \left[f(\mathbf{x}_{k-1}) \right. \\ & \left. + \langle \mathbf{v}_{k-1}, \mathbf{x} - \mathbf{x}_{k-1} \rangle + \frac{\mu}{2} \|\mathbf{x} - \mathbf{x}_{k-1}\|^2 \right] \end{aligned} \quad (3b)$$

where \mathbf{v}_{k-1} is the gradient estimate in SVRG or SARAH; while Φ_0^* , μ_0 , and δ_k are some constants to be specified later. The design is similar to that of (Kulunchakov and Mairal, 2019), but the ES here is constructed per inner loop. In

addition, here we will overcome the challenge of analyzing SARAH’s biased gradient estimate \mathbf{v}_k .

Upon defining $\Phi_k^* := \min_{\mathbf{x}} \Phi_k(\mathbf{x})$, the key properties of the sequence $\{\Phi_k(\mathbf{x})\}_{k=0}^m$ are collected in the next lemma.

Lemma 2. For $\{\Phi_k(\mathbf{x})\}_{k=0}^m$ as in (3), it holds that:

- i) $\Phi_0(\mathbf{x})$ is μ_0 -strongly convex, and $\Phi_k(\mathbf{x})$ is μ_k -strongly convex with $\mu_k = (1 - \delta_k)\mu_{k-1} + \delta_k\mu$;
- ii) \mathbf{x}_k minimizes $\Phi_k(\mathbf{x})$ if $\delta_k = \eta\mu_k$; and
- iii) $\Phi_k^* = (1 - \delta_k)\Phi_{k-1}^* + \delta_k f(\mathbf{x}_{k-1}) - \frac{\mu_k\eta^2}{2} \|\mathbf{v}_{k-1}\|^2$.

Lemma 2 holds for both SVRG and SARAH. To better understand the role of ES, it is instructive to use an example.

Example. With $\Phi_0^* = f(\mathbf{x}_0)$, $\mu_0 = \mu$, and $\delta_k = \mu_k\eta$ for SVRG, it holds that $\mu_k = \mu, \forall k$, and $\delta_k = \mu\eta, \forall k$. If for convenience we let $\delta := \mu\eta$, we show in Appendix A.2 that

$$\mathbb{E}[\Phi_k(\mathbf{x})] \leq (1 - \delta)^k [\Phi_0(\mathbf{x}) - f(\mathbf{x}^*)] + f(\mathbf{x}). \quad (4)$$

As $k \rightarrow \infty$, one has $(1 - \delta)^k \rightarrow 0$, and hence $\Phi_k(\mathbf{x})$ approaches in expectation a lower bound of $f(\mathbf{x})$.

Now, we are ready to view SVRG and SARAH through the lens of $\{\Phi_k(\mathbf{x})\}_{k=0}^m$ to obtain new averaging schemes.

3.2. Weighted Averaging for SVRG

The new averaging vector \mathbf{p}^s for SVRG together with the improved convergence rate is summarized in the following theorem.

Theorem 1. (SVRG with W-Avg.) Under Assumptions 1–3, construct the ES as in (3) with $\mu_0 = \mu$, $\delta_k = \mu_k\eta$, and $\Phi_0^* = f(\mathbf{x}_0)$. Choose $\eta < 1/(4L)$, and m large enough such that

$$\begin{aligned} \lambda^{SVRG} := & \frac{1}{1 - (1 - \mu\eta)^{m-1}} \left[\frac{(1 - \mu\eta)^m}{1 - 2\eta L} \right. \\ & \left. + \frac{2\mu L\eta^2(1 - \mu\eta)^{m-1}}{1 - 2L\eta} + \frac{2L\eta}{1 - 2L\eta} \right] < 1. \end{aligned}$$

Let $p_0^s = p_m^s = 0$, and $p_k^s = (1 - \mu\eta)^{m-k-1}/q$ for $k = 1, 2, \dots, m-1$, where $q = [1 - (1 - \mu\eta)^{m-1}]/(\mu\eta)$. It then holds for SVRG with this weighted averaging (W-Avg) that

$$\mathbb{E}[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)] \leq \lambda^{SVRG} \mathbb{E}[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^*)].$$

Comparing the W-Avg in Theorem 1 against U-Avg and L-Avg we saw in Section 2.2, the upshot of W-Avg is a much tighter convergence rate. When choosing $\eta = \mathcal{O}(1/L)$, the dominating terms of the convergence rate for W-Avg are $\mathcal{O}\left(\frac{(1-1/\kappa)^m}{1-2L\eta} + \frac{2L\eta}{1-2L\eta}\right)$, and $\mathcal{O}\left(\frac{\kappa}{m(1-2L\eta)} + \frac{2L\eta}{1-2L\eta}\right)$ for U-Avg (Johnson and Zhang, 2013). Clearly, the factor $(1 - 1/\kappa)^m$ in W-Avg can be much smaller than κ/m in U-Avg; see Fig. 1(a) for comparison of convergence rates of

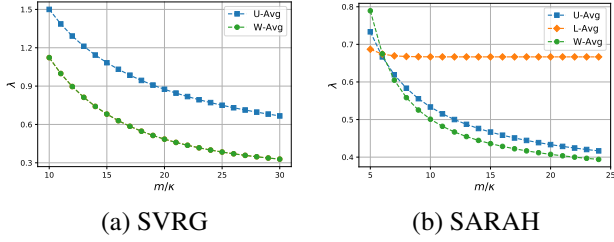


Figure 1. A comparison of the analytical convergence rate for SVRG and SARAH. In both figures we set $\kappa = 10^5$ with $L = 1$, $\mu = 10^{-5}$, and the step sizes are selected as: (a) SVRG with $\eta = 0.1/L$; and (b) SARAH with $\eta = 0.5/L$.

different averaging types. Since convergence of SVRG with L-Avg requires η and m to be chosen differently from those in U-Avg and W-Avg, L-Avg is not plotted in Fig. 1(a).

Next, we assess the complexity of SVRG with W-Avg.

Corollary 1. *Choosing $m = \mathcal{O}(\kappa)$ and other parameters as in Theorem 1, the complexity of SVRG with W-Avg to find $\tilde{\mathbf{x}}^s$ satisfying $\mathbb{E}[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)] \leq \epsilon$ is $\mathcal{O}((n + \kappa) \ln \frac{1}{\epsilon})$.*

Note that similar to U-Avg, W-Avg incurs lower complexity compared with L-Avg in (Tan et al., 2016; Hu et al., 2018).

3.3. Weighted Averaging for SARAH

SARAH is challenging to analyze due to the bias present in the estimate \mathbf{v}_k , which makes the ES-based treatment of SARAH fundamentally different from that of SVRG. To see this, it is useful to start with the following lemma.

Lemma 3. *For any deterministic \mathbf{x} , it holds in SARAH that*

$$\begin{aligned} & \mathbb{E}[\langle \mathbf{v}_k - \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle] \\ &= \frac{\eta}{2} \sum_{\tau=0}^{k-1} \mathbb{E}[\|\mathbf{v}_\tau - \nabla f(\mathbf{x}_\tau)\|^2 + \|\mathbf{v}_\tau\|^2 - \|\nabla f(\mathbf{x}_\tau)\|^2]. \end{aligned}$$

Lemma 3 reveals the main difference in the ES-based argument for SARAH, namely that $\mathbb{E}[\langle \mathbf{v}_k - \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle] \neq 0$, while the same inner product for SVRG equals to 0 in expectation. Reflecting back to (4), the consequence of having a non-zero $\mathbb{E}[\langle \mathbf{v}_k - \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle]$ is that $\mathbb{E}[\Phi_k(\mathbf{x})]$ is not necessarily approaching a lower bound of $f(\mathbf{x})$ as $k \rightarrow \infty$; thus,

$$\mathbb{E}[\Phi_k(\mathbf{x})] \leq (1 - \delta)^k [\Phi_0(\mathbf{x}) - f(\mathbf{x})] + f(\mathbf{x}) + C \quad (5)$$

where C is a non-zero term that is not present in (4) when applied to SVRG; see detailed derivations in Appendix A.2.

Interestingly, upon capitalizing on the properties of \mathbf{v}_k , the ensuing theorem establishes linear convergence for SARAH with a proper W-Avg vector \mathbf{p}^s .

Theorem 2. (SARAH with W-Avg.) *Under Assumptions 1 and 4, define the ES as in (3) with $\mu_0 = \mu$, $\delta_k = \mu_k \eta$, $\forall k$, and $\Phi_0^* = f(\mathbf{x}_0)$. With $\delta := \mu \eta$, select $\eta < 1/L$ and m large enough, so that*

$$\begin{aligned} \lambda^{\text{SARAH}} := & \left[(1 - \delta)^m - \left(1 - \frac{2\eta L}{1 + \kappa}\right)^m \right] \frac{L + \mu}{c(L - \mu)} \\ & + \frac{(1 - \delta)^m}{c\delta} + \frac{\eta L(m - 1)}{c(2 - \eta L)} + \frac{2 - 2\eta L}{2 - \eta L} \frac{1 + \kappa}{2c\eta L} < 1 \end{aligned}$$

where $c = m - \frac{1}{\delta} + \frac{(1 - \delta)^m}{\delta}$. Setting $p_k = (1 - (1 - \delta)^{m - k - 1})/c$, $\forall k = 0, 1, \dots, m - 2$, and $p_{m-1} = p_m = 0$, SARAH with this W-Avg satisfy

$$\mathbb{E}[\|\nabla f(\tilde{\mathbf{x}}^s)\|^2] \leq \lambda^{\text{SARAH}} \mathbb{E}[\|\nabla f(\tilde{\mathbf{x}}^{s-1})\|^2].$$

The expression of λ^{SARAH} is complicated because we want the upper bound of the convergence rate to be as tight as possible. To demonstrate this with an example, choosing $\eta = 1/(2L)$ and $m = 5\kappa$, we have $\lambda^{\text{SARAH}} \approx 0.8$. Fig. 1(b) compares SARAH with W-Avg versus SARAH with U-Avg and L-Avg. The advantage of W-Avg is more pronounced as m is chosen larger.

As far as complexity of SARAH with W-Avg, it is comparable with that of L-Avg or U-Avg, as asserted next.

Corollary 2. *Choosing $m = \mathcal{O}(\kappa)$ and other parameters as in Theorem 2, the complexity of SARAH with W-Avg to find $\tilde{\mathbf{x}}^s$ satisfying $\mathbb{E}[\|\nabla f(\tilde{\mathbf{x}}^s)\|^2] \leq \epsilon$, is $\mathcal{O}((n + \kappa) \ln \frac{1}{\epsilon})$.*

A few remarks are now in order on our analytical findings: i) most existing ES-based proofs use $\mathbb{E}[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)]$ as optimality metric, while Theorem 2 and Corollary 2 rely on $\mathbb{E}[\|\nabla f(\tilde{\mathbf{x}}^s)\|^2]$; ii) the analysis method still holds when Assumption 4 is weakened to Assumption 3, at the price of having worse κ -dependence of the complexity, that is, $\mathcal{O}((n + \kappa^2) \ln \frac{1}{\epsilon})$, which is of the same order as L-Avg under Assumptions 1 – 3 (Li et al., 2020; Liu et al.).

3.4. Averaging Is More Than A ‘Proof Trick’

Existing forms of averaging such as U-Avg and W-Avg, are typically considered as ‘proof tricks’ for simplifying the theoretical analysis (Johnson and Zhang, 2013; Tan et al., 2016; Nguyen et al., 2017; Li et al., 2020). In this subsection, we contend that averaging can distinctly affect performance, and should be adapted to other parameters. We will take SARAH with $\eta = \mathcal{O}(1/L)$ and $m = \mathcal{O}(\kappa)$ as an example rather than SVRG since such parameter choices guarantee convergence regardless of the averaging employed. (For SVRG with L-Avg on the other hand, the step size has to be chosen differently with W-Avg or U-Avg.)

We will first look at the convergence rate of SARAH across different averaging options. Fixing $m = \mathcal{O}(\kappa)$ and changing η , the theoretical convergence rate is plotted in Fig. 2. It

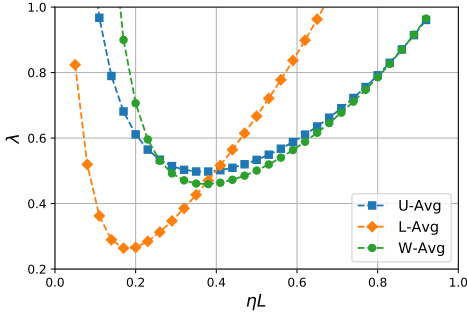


Figure 2. SARAH’s analytical convergence rate with different averaging options ($\kappa = 10^5$, $L = 1$, $\mu = 10^{-5}$, and fixed $m = 10\kappa$).

is observed that with smaller step sizes, L-Avg enjoys faster convergence, while larger step sizes tend to favor W-Avg and U-Avg instead.

Next, we will demonstrate empirically that the type of averaging indeed matters. Consider binary classification using the regularized logistic loss function

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i \in [n]} \ln [1 + \exp(-b_i \langle \mathbf{a}_i, \mathbf{x} \rangle)] + \frac{\mu}{2} \|\mathbf{x}\|^2 \quad (6)$$

where (\mathbf{a}_i, b_i) is the (feature, label) pair of datum i . Clearly, (6) is an instance of the cost in (1) with $f_i(\mathbf{x}) = \ln [1 + \exp(-b_i \langle \mathbf{a}_i, \mathbf{x} \rangle)] + \frac{\mu}{2} \|\mathbf{x}\|^2$; and it can be readily verified that Assumptions 1 and 4 are satisfied in this case.

SARAH with L-Avg, U-Avg and W-Avg are tested with fixed (moderate) $m = \mathcal{O}(\kappa)$ but different step size choices on the dataset *w7a*; see also Appendix D.1 for additional tests with datasets *a9a* and *diabetes*. Fig. 3(a) shows that for a large step size $\eta = 0.9/L$, W-Avg outperforms U-Avg as well as L-Avg by almost two orders at the 30th sample pass. For a medium step size $\eta = 0.6/L$, W-Avg and L-Avg perform comparably, while both are outperformed by U-Avg. When η is chosen small, L-Avg is clearly the winner. In short, the performance of averaging options varies with the step sizes. This is intuitively reasonable because the MSE of \mathbf{v}_k : i) scales with η (cf. Lemma 1); and ii) tends to increase with k as $\mathbb{E}[\|\mathbf{v}_k\|^2]$ decreases linearly (see Lemma 5 in Appendix B.2, and the MSE bound in Lemma 1). As a result, when both η and k are large, the MSE of \mathbf{v}_k tends to be large too. Iterates with gradient estimates having high MSE can jeopardize the convergence. This explains the inferior performance of L-Avg in Figs. 3(a) and 3(b). On the other hand, when η is chosen small, the MSE tends to be small as well; hence, working with L-Avg does not compromise convergence, while in expectation W-Avg and U-Avg compute full gradient more frequently than L-Avg. These two reasons explain the improved performance of L-Avg in Fig. 3(c).

When we fix η and change m , as depicted in Fig. 1(b), the analytical convergence rate of W-Avg improves over that of U-Avg and L-Avg when m is large. This is because the MSE of \mathbf{v}_k increases with k . W-Avg and U-Avg ensure better performance through “early ending,” by reducing the number of updates that utilize \mathbf{v}_k with large MSE.

In sum, the choice of averaging scheme should be adapted with η and m to optimize performance. For example, the proposed W-Avg for SARAH favors the regime where either η or m is chosen large, as dictated by the convergence rates and corroborated by numerical tests.

4. Tune-Free Variance Reduction

This section copes with variance reduction without tuning. In particular, i) Barzilai-Borwein (BB) step size, ii) averaging schemes, and iii) a time varying inner loop length are adopted for the best empirical performance.

4.1. Recap of BB Step Sizes

Aiming to develop ‘tune-free’ SVRG and SARAH, we will first adopt the BB scheme to obtain suitable step sizes automatically (Tan et al., 2016). In a nutshell, BB monitors progress of previous outer loops, and chooses the step size of outer loop s accordingly via

$$\eta^s = \frac{1}{\theta_\kappa} \frac{\|\tilde{\mathbf{x}}^{s-1} - \tilde{\mathbf{x}}^{s-2}\|^2}{\langle \tilde{\mathbf{x}}^{s-1} - \tilde{\mathbf{x}}^{s-2}, \nabla f(\tilde{\mathbf{x}}^{s-1}) - \nabla f(\tilde{\mathbf{x}}^{s-2}) \rangle} \quad (7)$$

where θ_κ is a κ -dependent parameter to be specified later. Note that $\nabla f(\tilde{\mathbf{x}}^{s-1})$ and $\nabla f(\tilde{\mathbf{x}}^{s-2})$ are computed at the outer loops s and $s - 1$, respectively; hence, the implementation overhead of BB step sizes only includes almost negligible memory to store $\tilde{\mathbf{x}}^{s-2}$ and $\nabla f(\tilde{\mathbf{x}}^{s-2})$.

BB step sizes for SVRG with L-Avg have relied on $\theta_\kappa = m = \mathcal{O}(\kappa^2)$ (Tan et al., 2016). Such a choice of parameters offers provable convergence at complexity $\mathcal{O}((n + \kappa^2) \ln \frac{1}{\epsilon})$, but has not been effective in our simulations for two reasons: i) step size η^s depends on m , which means that tuning is still required for step sizes; and, ii) the optimal m of $\mathcal{O}(\kappa)$ with best empirical performance significantly deviates from the theoretically suggested $\mathcal{O}(\kappa^2)$; see also Fig. 4(a). Other BB based variance reduction methods introduce extra parameters to be tuned in addition to m ; e.g., some scalars in (Liu et al.) and the mini-batch sizes in (Yang et al., 2019). This prompts us to design more practical BB methods – how to choose m with minimal tuning is also of major practical value.

4.2. Averaging for BB Step Sizes

We start with a fixed choice of m to theoretically investigate different types of averaging for the BB step sizes. The final

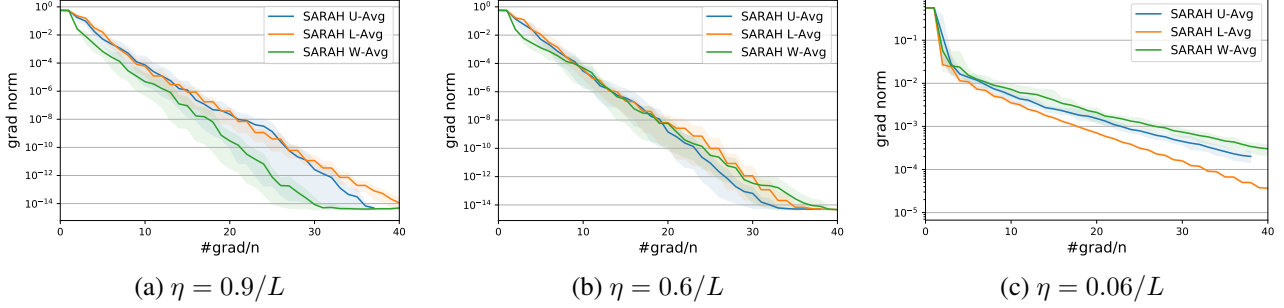


Figure 3. Comparing SARAH with different types of averaging on dataset $w7a$ ($\mu = 0.005$ and $m = 5\kappa$ in all tests).

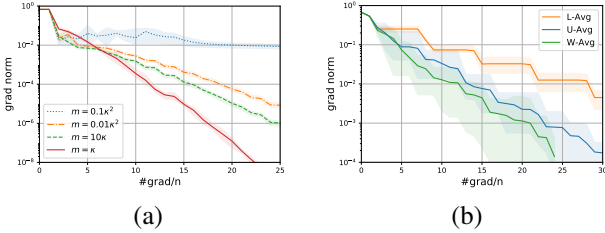


Figure 4. (a) Performance of BB-SVRG (Tan et al., 2016) under different choices of m . (b) BB-SARAH with different averaging schemes. Both tests use dataset $a9a$ with $\kappa = 1,388$.

‘tune-free’ implementation of SVRG and SARAH will rely on the analysis of this subsection.

Proposition 1. (BB-SVRG) Under Assumptions 1 – 3, if we choose $m = \mathcal{O}(\kappa^2)$ and $\theta_\kappa = \mathcal{O}(\kappa)$ (but with $\theta_\kappa > 4\kappa$), then BB-SVRG with U-Avg and W-avg can find $\tilde{\mathbf{x}}^s$ with $\mathbb{E}[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^*)] \leq \epsilon$ using $\mathcal{O}((n + \kappa^2) \ln \frac{1}{\epsilon})$ IFO calls.

Similar to BB-SVRG, the ensuing result asserts that for BB-SARAH, W-Avg, U-Avg and L-Avg have identical order of complexity.

Proposition 2. (BB-SARAH) Under Assumptions 1 and 4, if we choose $m = \mathcal{O}(\kappa^2)$ and $\theta_\kappa = \mathcal{O}(\kappa)$, then BB-SARAH finds a solution with $\mathbb{E}[\|\nabla f(\tilde{\mathbf{x}}^s)\|^2] \leq \epsilon$ using $\mathcal{O}((n + \kappa^2) \ln \frac{1}{\epsilon})$ IFO calls, when one of these conditions holds: i) either U-Avg with $\theta_\kappa > \kappa$; or ii) L-Avg with $\theta_\kappa > 3/2\kappa$; or; iii) W-Avg with $\theta_\kappa > \kappa$.

The price paid for having automatically tuned step sizes is a worse dependence of the complexity on κ , compared with the bounds in Corollaries 1 and 2. The cause of the worse dependence on κ is that one has to choose a large m at the order of $\mathcal{O}(\kappa^2)$. However, such an automatic tuning of the step size comes almost as a ‘free lunch’ when problem (1) is well conditioned, or, in the big data regime, e.g., $\kappa^2 \approx n$ or $\kappa^2 \ll n$, since the dominant term in complexity is $\mathcal{O}(n \ln \frac{1}{\epsilon})$ for both SVRG and BB-SVRG. On the other hand, it is prudent to stress that with $\kappa^2 \gg n$, the BB step sizes slow down convergence.

Given the same order of complexity, the empirical performance of BB-SARAH with different averaging types is showcased in Fig. 4(b) with the parameters chosen as in Proposition 2. It is observed that W-Avg converges most rapidly, while U-Avg outperforms L-Avg. This confirms our theoretical insight, that is, W-Avg and U-Avg are more suitable when m is chosen large enough.

4.3. Tune-Free Variance Reduction

Next, the ultimate format of the almost tune-free variance reduction is presented using SARAH as an example. We will discuss how to choose the iteration number of inner loops and averaging schemes for BB step sizes.

Adaptive inner loop length. It is observed that the BB step size can change over a wide range of values (see Appendix C for derivations),

$$\frac{1}{\theta_\kappa L} \leq \eta^s \leq \frac{1}{\theta_\kappa \mu}. \quad (8)$$

Given $\theta_\kappa = \mathcal{O}(\kappa)$, η^s can vary from $\mathcal{O}(\mu/L^2)$ to $\mathcal{O}(1/L)$. Such a wide range of η^s blocks the possibility to find a single m suitable for both small and large η^s at the same time. From a theoretical perspective, choosing $m = \mathcal{O}(\kappa^2)$ in both Propositions 1 and 2 is mainly for coping with the small step sizes $\eta^s = \mathcal{O}(1/(L\theta_\kappa))$. But such a choice is too pessimistic for large ones $\eta^s = \mathcal{O}(1/(\mu\theta_\kappa))$. In fact, choosing $m = \mathcal{O}(\kappa)$ for $\eta^s = \mathcal{O}(1/L)$ is good enough, as suggested by Corollaries 1 and 2. These observations motivate us to design an m^s that changes dynamically per outer loop s .

Reflecting on the convergence of SARAH, it is sufficient to set the inner loop length m^s according to the η^s used. To highlight the rationale behind our choice of m^s , let us consider BB-SARAH with U-Avg as an example that features convergence rate $\lambda^s = \frac{1}{\mu\eta^s m^s} + \frac{\eta^s L}{2 - \eta^s L}$ (Nguyen et al., 2017). Set $\theta_\kappa > \kappa$ as in Proposition 2 so that the second term of λ^s is always less than 1. With a large step size $\eta^s = \mathcal{O}(1/L)$, and by simply choosing $m^s = \mathcal{O}(1/(\mu\eta^s))$, one can ensure a convergent iteration having e.g., $\lambda^s < 1$.

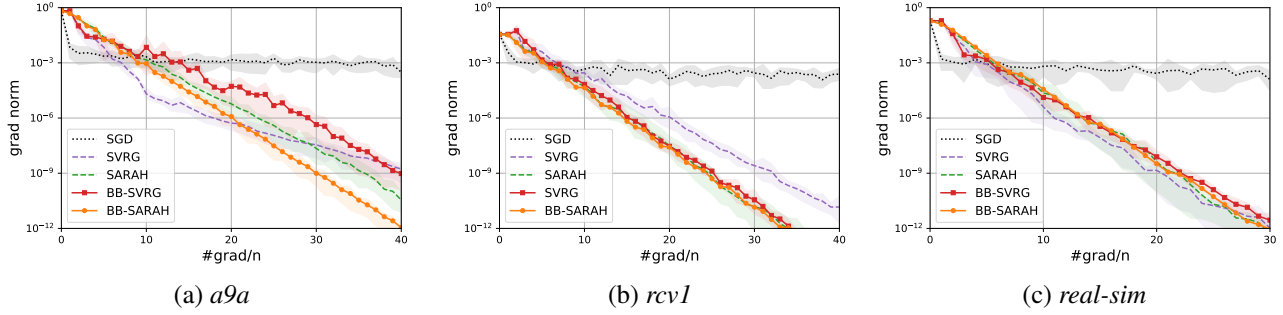


Figure 5. Tests of BB-SVRG and BB-SARAH on different datasets.

With a small step size $\eta^s = \mathcal{O}(1/(\kappa L))$ though, choosing $m^s = \mathcal{O}(1/(\mu\eta^s))$ also leads to $\lambda^s < 1$. These considerations prompt us to adopt a time-varying inner loop length adjusted by η^s in (7) as

$$m^s = \frac{c}{\mu\eta^s}. \quad (9)$$

Such choices of η^s and m^s at first glance do not lead to a tune-free algorithm directly, because one has to find an optimal θ_κ and c through tuning. Fortunately, there are simple choices for both c and θ_κ . In Propositions 1 and 2, the smallest selected θ_κ for SVRG and SARAH with different types of averaging turns out to be a reliable choice; while choosing $c = 1$ has been good enough throughout our numerical experiments. Although the selection of these parameters violates slightly the theoretical guarantee, its merits lie in the simplicity. And in our experiments, no divergence has been observed by these parameter selections.

Averaging schemes. As discussed in subsection 3.4, W-Avg gains in performance when either m^s or η^s is large. Since m^s and η^s are inversely proportional (cf. (9)), it is clear that one of the two suffices to be large; and for this reason, we will rely on W-Avg for BB-SARAH.

Extensions regarding almost tune-free variance reduction for (non)convex problems can be found in our technical note (Li and Giannakis, 2019).

5. Numerical Tests

To assess performance, the proposed tune-free BB-SVRG and BB-SARAH are applied to binary classification via regularized logistic regression (cf. (6)) using the datasets *a9a*, *rcv1.binary*, and *real-sim* from LIBSVM³. Details regarding the datasets, the μ values used, and implementation details are deferred to Appendix D.2.

For comparison, the selected benchmarks are SGD, SVRG with U-Avg, and SARAH with U-Avg. The step size for

³Online available at <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html>.

SGD is $\eta = 0.05/(L(n_e + 1))$, where n_e is the index of epochs. For SVRG and SARAH, we fix $m = 5\kappa$, and tune for the best step sizes. For BB-SVRG, we choose η^s and m^s as (9) with $\theta_\kappa = 4\kappa$ (as in Proposition 1) and $c = 1$. While we choose $\theta_\kappa = \kappa$ (as in Proposition 2) and $c = 1$ for BB-SARAH. W-Avg is adopted for both BB-SVRG and BB-SARAH.

The results are showcased in Fig. 5. We also tested BB-SVRG with parameters chosen as (Tan et al., 2016, Thm. 3.8). However it only slightly outperforms SGD and hence is not plotted here (see the blue line in Fig. 4(a) as a reference). On dataset *a9a*, BB-SARAH outperforms tuned SARAH. BB-SVRG is worse than SVRG initially, but has similar performance around the 40th sample pass on the x-axis. On dataset *rcv1* however, BB-SARAH, BB-SVRG and SARAH have similar performance, improving over SVRG. On dataset *real-sim*, BB-SARAH performs almost identical to SARAH. BB-SVRG exhibits comparable performance with SVRG.

6. Conclusions

Almost tune-free SVRG and SARAH were developed in this work. Besides the BB step size for eliminating the tuning for step size, the key insights are that both i) averaging, as well as ii) the number of inner loop iterations should be adjusted according to the BB step size. Specific major findings include: i) estimate sequence based provably linear convergence of SVRG and SARAH, which enabled new types of averaging for efficient variance reduction; ii) theoretical guarantees of BB-SVRG and BB-SARAH with different types of averaging; and, iii) implementable tune-free variance reduction algorithms. The efficacy of the tune-free BB-SVRG and BB-SARAH were corroborated numerically.

Acknowledgements

The authors would like to thank anonymous reviewers for their constructive feedback. We also gratefully acknowledge the support from NSF grants 1711471, and 1901134.

References

- Alekh Agarwal and Leon Bottou. A lower bound for the optimization of finite sums. In *Proc. Intl. Conf. on Machine Learning*, pages 78–86, Lille, France, 2015.
- Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.
- Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.
- Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. In *Proc. Advances in Neural Info. Process. Syst.*, pages 15210–15219, Vancouver, Canada, 2019.
- Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Proc. Advances in Neural Info. Process. Syst.*, pages 1646–1654, Montreal, Canada, 2014.
- Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Proc. Advances in Neural Info. Process. Syst.*, pages 687–697, Montreal, Canada, 2018.
- Bin Hu, Stephen Wright, and Laurent Lessard. Dissipativity theory for accelerating stochastic variance reduction: A unified analysis of SVRG and Katyusha using semidefinite programs. *arXiv preprint arXiv:1806.03677*, 2018.
- Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. Advances in Neural Info. Process. Syst.*, pages 315–323, Lake Tahoe, Nevada, 2013.
- Jakub Konecny and Peter Richtárik. Semi-stochastic gradient descent methods. *arXiv preprint arXiv:1312.1666*, 2013.
- Dmitry Kovalev, Samuel Horvath, and Peter Richtárik. Don’t jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop. *arXiv preprint arXiv:1901.08689*, 2019.
- Andrei Kulunchakov and Julien Mairal. Estimate sequences for variance-reduced stochastic composite optimization. In *Proc. Intl. Conf. Machine Learning*, pages 3541–3550, Long Beach, CA, 2019.
- Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via SCSG methods. In *Proc. Advances in Neural Info. Process. Syst.*, pages 2348–2358, Long Beach, CA, 2017.
- Bingcong Li and Georgios B Giannakis. Adaptive step sizes in variance reduction via regularization. *arXiv preprint arXiv:1910.06532*, 2019.
- Bingcong Li, Meng Ma, and Georgios B Giannakis. On the convergence of SARAH and beyond. In *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, Palermo, Italy, 2020.
- Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Proc. Advances in Neural Info. Process. Syst.*, pages 3384–3392, Montreal, Canada, 2015.
- Yan Liu, Congying Han, and Tiande Guo. A class of stochastic variance reduced methods with an adaptive step-size. URL http://www.optimization-online.org/DB_FILE/2019/04/7170.pdf.
- Julien Mairal. Optimization with first-order surrogate functions. In *Proc. Intl. Conf. on Machine Learning*, pages 783–791, Atlanta, 2013.
- Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2004.
- Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proc. Intl. Conf. Machine Learning*, Sydney, Australia, 2017.
- Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *Proc. Advances in Neural Info. Process. Syst.*, pages 1574–1582, Montreal, Canada, 2014.
- Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *Proc. Intl. Conf. on Machine Learning*, pages 314–323, New York City, NY, 2016.
- Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.
- Nicolas L Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Proc. Advances in Neural Info. Process. Syst.*, pages 2663–2671, Lake Tahoe, Nevada, 2012.
- Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, vol. 14:pp. 567–599, Feb. 2013.

Conghui Tan, Shiqian Ma, Yu-Hong Dai, and Yuqiu Qian. Barzilai-Borwein step size for stochastic gradient descent. In *Proc. Advances in Neural Info. Process. Syst.*, pages 685–693, Barcelona, Spain, 2016.

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

Zhuang Yang, Zengping Chen, and Cheng Wang. Accelerating mini-batch sara by step size rules. *arXiv preprint arXiv:1906.08496*, 2019.