On the Convergence of SARAH and Beyond

Bingcong Li
University of Minnesota

Meng Ma
University of Minnesota

Georgios B. Giannakis
University of Minnesota

Abstract

The main theme of this work is a unifying algorithm, **L**oop**L**ess **S**ARAH (L2S) for problems formulated as summation of n individual loss functions. L2S broadens a recently developed variance reduction method known as SARAH. To find an ϵ-accurate solution, L2S enjoys a complexity of $\mathcal{O}\big((n+\kappa)\ln(1/\epsilon)\big)$ for strongly convex problems. For convex problems, when adopting an n-dependent step size, the complexity of L2S is $\mathcal{O}(n+\sqrt{n}/\epsilon)$; while for more frequently adopted n-independent step size, the complexity is $\mathcal{O}(n+n/\epsilon)$. Distinct from SARAH, our theoretical findings support an n-independent step size in convex problems without extra assumptions. For nonconvex problems, the complexity of L2S is $\mathcal{O}(n+\sqrt{n}/\epsilon)$. Our numerical tests on neural networks suggest that L2S can have better generalization properties than SARAH. Along with L2S, our side results include the linear convergence of the last iteration for SARAH in strongly convex problems.

1 INTRODUCTION

Consider the frequently encountered empirical risk minimization (ERM) problem

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

where $\mathbf{x} \in \mathbb{R}^d$ is the parameter to be learned from data; the set $[n] := \{1, 2, \ldots, n\}$ collects data indices; and, f_i is the loss function corresponding to datum i. Suppose that the set of minimizers is non-empty and F is bounded from below.

The standard method to solve (1) is gradient descent (GD), which per iteration t relies on the update

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\nabla F(\mathbf{x}_t)$$

where η is the step size (a.k.a learning rate). For a strongly convex F, GD convergences linearly to $\mathbf{x}^*$, meaning after T iterations it holds that $\|\mathbf{x}_T - \mathbf{x}^*\|^2 \leq c^T\|\mathbf{x}_0 - \mathbf{x}^*\|^2$ with some constant $c \in (0, 1)$; while for convex F it holds that $F(\mathbf{x}_T) - F(\mathbf{x}^*) = \mathcal{O}(1/T)$, and for nonconvex F one has $\min_t \|\nabla F(\mathbf{x}_t)\| = \mathcal{O}(1/T)$; see e.g., [Nesterov, 2004, Ghadimi and Lan, 2013]. However, finding $\nabla F(\mathbf{x}_t)$ per iteration in the big data regime, i.e., with large n, can be computationally prohibitive. To cope with this, the stochastic gradient descent (SGD) [Robbins and Monro, 1951, Bottou et al., 2016] draws uniformly at random an index $i_t \in [n]$ per iteration, and updates via

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t\nabla f_{i_t}(\mathbf{x}_t).$$

Albeit computationally light, SGD comes with a slower convergence rate than GD [Bottou et al., 2016, Ghadimi and Lan, 2013], which is mainly due to the variance of the gradient estimate given by $\mathbb{E}[\|\nabla f_{i_t}(\mathbf{x}_t) - \nabla F(\mathbf{x}_t)\|^2]$.

By capitalizing on the finite sum structure of ERM, a class of algorithms, variance reduction family, can be designed to solve (1) more efficiently. The idea is to judiciously (often periodically) evaluate a *snapshot gradient* $\nabla F(\mathbf{x}_s)$, and use it as an anchor of the stochastic draws $\{\nabla f_{i_t}(\mathbf{x}_t)\}$ in subsequent iterations. As a result, compared with the simple gradient estimate $\nabla f_{i_t}(\mathbf{x}_t)$ in SGD, the variance of estimated gradients can be reduced. Members of the variance reduction family include SDCA [Shalev-Shwartz and Zhang, 2013], SVRG [Johnson and Zhang, 2013, Reddi et al., 2016a, Allen-Zhu and Hazan, 2016], SAG [Roux et al., 2012], SAGA [Defazio et al., 2014, Reddi et al., 2016b], MISO [Mairal, 2013], SCSG [Lei and Jordan, 2017, Lei et al., 2017], SNVRG [Zhou et al., 2018] and SARAH [Nguyen et al., 2017, 2019], and their variants [Konecnỳ and Richtárik, 2013, Kovalev et al., 2019, Qian et al., 2019, Li et al., 2019]. Most of these rely on the update $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\mathbf{v}_t$, where η is a *constant* step size and $\mathbf{v}_t$ is a carefully designed gradient estimator that takes advantage of the snapshot gradient. When aiming for an accurate solution, variance reduction methods are faster than SGD for convex and nonconvex problems, and remarkably they

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

converge linearly when F is strongly convex. The *complexity* of algorithms such as GD, SGD, and variance reduction families will be quantified by the number of incremental first-order oracle (IFO) calls that counts how many (incremental) gradients are computed [Agarwal and Bottou, 2015], as specified next using our notational conventions.

Definition 1. *An IFO is a black box with inputs f_i and $\mathbf{x} \in \mathbb{R}^d$, and output the gradient $\nabla f_i(\mathbf{x})$.*

For example, the IFO complexity to compute $\nabla F(\mathbf{x})$ is n. For a prescribed ϵ, a desirable algorithm obtains an ϵ-accurate solution (defined as follows) with minimal complexity[1].

Definition 2. *Let $\mathbf{x}$ be a solution returned by certain algorithm. If $\mathbb{E}[\|\nabla F(\mathbf{x})\|^2] \leq \epsilon$ is satisfied, $\mathbf{x}$ is termed as an ϵ-accurate solution to* (1)*.*

Variance reduction algorithms outperform GD in terms of complexity. And when high accuracy (ϵ small) is desired, the complexity of variance reduction methods is also lower than that of SGD. Among variance reduction algorithms, the distinct feature of SARAH [Nguyen et al., 2017, 2019] and its variants [Fang et al., 2018, Zhang et al., 2018, Wang et al., 2018, Nguyen et al., 2018a, Pham et al., 2019] is that they rely on a *biased* gradient estimator $\mathbf{v}_t$ formed by recursively using stochastic gradients. SARAH performs comparably to SVRG/SAGA on strongly convex problems, but reduces the complexity of SVRG/SAGA for nonconvex losses. In addition, no duality (as in SDCA) or gradient table (for SAGA) is required. With SARAH's analytical and practical merits granted, there are unexplored issues. For example, guarantees on SARAH with *n-independent* step size for convex problems are missing since analysis in [Nguyen et al., 2017] requires an extra presumption. The last iteration convergence of SARAH is also not well understood yet. In this context, our contributions are summarized next.

- **Unifying algorithm and novel analysis:** A new algorithm, **L**oop**L**ess **S**ARAH (L2S) is developed. It offers a unified algorithmic framework with provable convergence properties through a novel analyzing technique. To find an ϵ-accurate solution, L2S enjoys a complexity $\mathcal{O}\big((n+\kappa)\ln(1/\epsilon)\big)$ for strongly convex problems with condition number κ. For convex problems, the complexity of L2S is $\mathcal{O}(n+\sqrt{n}/\epsilon)$ when an n-related step size is used; or $\mathcal{O}(n+n/\epsilon)$ for an n-independent step size. The complexity of L2S for nonconvex problems is $\mathcal{O}(n+\sqrt{n}/\epsilon)$.
- **Tale of generalization:** Supported by experimental evidence, we find that L2S can have generalization merits compared with SARAH for nonconvex tasks such as training neural networks.
- **Last iteration convergence of SARAH:** Linear convergence of the last iteration for SARAH on μ-strongly convex problems is established. Distinct from [Liu et al.] with step size $\mathcal{O}(\mu/L^2)$, our analysis enables a much larger step size, i.e., $\eta = \mathcal{O}(1/L)$. In addition, we find that if each f_i is strongly convex, the complexity of adopting last iteration in SARAH is lower than that of SVRG.

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

2 PRELIMINARIES

This section reviews SARAH [Nguyen et al., 2017, 2019] with emphases on the quality of gradient estimates. Before diving into SARAH, we first state the assumptions posed on F and f_i.

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

Assumptions 1 – 4 are standard in the analysis of variance reduction algorithms. Assumption 1 requires each loss function to be sufficiently smooth. In fact one can distinguish the smoothness of individual loss function and refine Assumption 1 as f_i has L_i-Lipchitz gradient. Clearly $L = \max_i L_i$. With slight modifications on SARAH, such refinement can tighten the complexity bounds slightly. The detailed discussions can be found in Appendix E. In the main text, we will keep using the simpler Assumption 1 for clarity. Assumption 2 implies that F is also convex. Assumption 3 only requires F to be strongly convex, which is slightly weaker than Assumption 4. And it is clear when Assumption 4 is true, both Assumptions 2 and 3 hold automatically. Under Assumptions 1 and 3 (or 4), the condition number of F is defined as $\kappa := L/\mu$.

2.1 Recap of SARAH

SARAH for Strongly Convex Problems: The detailed steps of SARAH are listed under Alg. 1. In a particular outer loop (lines 3 - 11) indexed by s, a snapshot gradient $\mathbf{v}_0^s = \nabla F(\mathbf{x}_0^s)$ is computed first to serve as an anchor of gradient estimates $\mathbf{v}_t^s$ in the ensuing inner loop (lines 6 - 10). Then $\mathbf{x}_0^s$ is updated $m+1$ times based on $\mathbf{v}_t^s$

$$\mathbf{v}_t^s = \nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\mathbf{x}_{t-1}^s) + \mathbf{v}_{t-1}^s. \qquad (2)$$

[1] Complexity is the abbreviation for IFO complexity throughout this work.

Algorithm 1 SARAH

```
1: **Initialize:** $\tilde{\mathbf{x}}^0$, $\eta$, $m$, $S$
2: **for** $s = 1, 2, \ldots, S$ **do**
3:     $\mathbf{x}_0^s = \tilde{\mathbf{x}}^{s-1}$
4:     $\mathbf{v}_0^s = \nabla F(\mathbf{x}_0^s)$
5:     $\mathbf{x}_1^s = \mathbf{x}_0^s - \eta \mathbf{v}_0^s$
6:     **for** $t = 1, 2, \ldots, m$ **do**
7:         uniformly sample $i_t \in [n]$
8:         $\mathbf{v}_t^s = \nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\mathbf{x}_{t-1}^s) + \mathbf{v}_{t-1}^s$
9:         $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s - \eta \mathbf{v}_t^s$
10:    **end for**
11:    choose $\tilde{\mathbf{x}}^s$ uniformly rnd. from $\{\mathbf{x}_t^s\}_{t=0}^m$
12: **end for**
13: **Output:** $\tilde{\mathbf{x}}^S$
```

SARAH's gradient estimator $\mathbf{v}_t^s$ is *biased*, since $\mathbb{E}\big[\mathbf{v}_t^s | \mathcal{F}_{t-1}\big] = \nabla F(\mathbf{x}_t^s) - \nabla F(\mathbf{x}_{t-1}^s) + \mathbf{v}_{t-1}^s \neq \nabla F(\mathbf{x}_t^s)$, where $\mathcal{F}_{t-1} := \sigma(\mathbf{x}_0^s, i_1, i_2, \ldots, i_{t-1})$ denotes the σ-algebra generated by $\mathbf{x}_0^s, i_1, i_2, \ldots, i_{t-1}$. Albeit biased, $\mathbf{v}_t^s$ is carefully designed to ensure the mean square error (MSE) relative to $\nabla F(\mathbf{x}_t^s)$ is bounded above, and stays proportional to $\mathbb{E}[\|\nabla F(\tilde{\mathbf{x}}^{s-1})\|^2]$.

Lemma 1. *[Nguyen et al., 2017, Lemma 2] If Assumptions 1 and 2 hold and $\eta < 2/L$, SARAH guarantees that*

$$\mathbb{E}\big[\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|^2\big] \leq \frac{\eta L}{2 - \eta L} \mathbb{E}\big[\|\nabla F(\tilde{\mathbf{x}}^{s-1})\|^2\big], \ \forall t.$$

This MSE bound of Lemma 1 is critical for analyzing SARAH, and instrumental in establishing its linear convergence for strongly convex F. It is worth stressing that the step size of SARAH should be chosen by $\eta < 1/L$ to ensure convergence, which can be larger than that of SVRG, whose step size should be less than $1/(4L)$.

SARAH for Convex Problems: Establishing the convergence rate of SARAH with an n-independent step size remains open for convex problems. Regarding complexity, the only analysis implicitly assumes SARAH to be *non-divergent*, as confirmed by the following claim used to derive the complexity.

Claim: [Nguyen et al., 2017, Theorem 3] If $\delta_s := \frac{2}{\eta(m+1)} \mathbb{E}\big[F(\tilde{\mathbf{x}}^s) - F(\mathbf{x}^*)\big]$, $\delta := \max_s \delta_s$, $\Delta := \delta + \frac{\delta \eta L}{2 - 2\eta L}$, and $\alpha = \frac{\eta L}{2 - \eta L}$, it holds that $\mathbb{E}[\|\nabla F(\tilde{\mathbf{x}}^s)\|^2] - \Delta \leq \alpha^s (\|F(\tilde{\mathbf{x}}^0)\|^2 - \Delta)$.

The missing piece of this claim is that for a finite δ_s or δ, $\mathbb{E}[F(\tilde{\mathbf{x}}^s) - F(\mathbf{x}^*)]$ must be bounded; or equivalently, the algorithm must be assumed *non-divergent*. Even if $\mathbb{E}[F(\tilde{\mathbf{x}}^s) - F(\mathbf{x}^*)]$ is finite, assuming it to be $\mathcal{O}(1)$ as in [Nguyen et al., 2017] is not reasonable. Another variant of SARAH in [Nguyen et al., 2018b] also relies on a similar assumption to guarantee convergence. We will show that the proposed algorithm can bypass this extra non-divergent assumption.

Algorithm 2 L2S

```
**Initialize:** $\mathbf{x}_0$, $\eta$, $m$, $T$
compute $\mathbf{v}_0 = \nabla F(\mathbf{x}_0)$
$\mathbf{x}_1 = \mathbf{x}_0 - \eta \mathbf{v}_0$
**for** $t = 1, 2, \ldots, T$ **do**
    choose $\mathbf{v}_t$ via (3).
    $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{v}_t$
**end for**
**Output:** $\mathbf{x}_a$ that is uniformly chosen from $\{\mathbf{x}_t\}_{t=1}^T$
```

SARAH for Nonconvex Problems. SARAH also works for nonconvex problems if line 11 in Alg. 1 is modified to $\tilde{\mathbf{x}}^s = \mathbf{x}_{m+1}^s$. The key to convergence again lies in the MSE of $\mathbf{v}_t^s$.

Lemma 2. *[Fang et al., 2018, Lemma 1] If Assumption 1 holds, the MSE of $\mathbf{v}_t^s$ is bounded by*

$$\mathbb{E}\big[\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|^2\big] \leq \eta^2 L^2 \sum_{\tau=0}^{t-1} \mathbb{E}\big[\|\mathbf{v}_\tau^s\|^2\big].$$

Lemma 2 states that the upper bound of MSE of $\mathbf{v}_t^s$ is i) proportional to η^2; and, ii) larger when t is larger. Leveraging the MSE bound, it was established that the complexity to find an ϵ-accurate solution is $\mathcal{O}(n + \sqrt{n}/\epsilon)$ [Nguyen et al., 2019]. Compared with SARAH, the proposed algorithm has its own merits for tasks such as training neural network, which will be clear in Section 4.

3 LOOPLESS SARAH

This section presents the **L**oop**L**ess **S**ARAH (L2S) algorithmic framework, which is capable of dealing with (strongly) convex and nonconcex ERM problems.

L2S is summarized in Alg. 2. Besides the single loop structure, the most distinct feature of L2S is that $\mathbf{v}_t$ is a probabilistically computed snapshot gradient given by

$$\mathbf{v}_t = \begin{cases} \nabla F(\mathbf{x}_t) & \text{w.p. } 1/m \\ \nabla f_{i_t}(\mathbf{x}_t) - \nabla f_{i_t}(\mathbf{x}_{t\text{-}1}) + \mathbf{v}_{t\text{-}1} & \text{w.p. } 1 - 1/m \end{cases} \tag{3}$$

where $i_t \in [n]$ is again uniformly sampled. The gradient estimator $\mathbf{v}_t$ is still biased, since $\mathbb{E}[\mathbf{v}_t | \mathcal{F}_{t-1}] = \nabla F(\mathbf{x}_t) - (1 - \frac{1}{m})\big[\nabla F(\mathbf{x}_{t-1}) - \mathbf{v}_{t-1}^s\big] \neq \nabla F(\mathbf{x}_t)$. In L2S, the snapshot gradient is computed every m iterations *in expectation*, while SARAH computes the snapshot gradient once every $m + 1$ updates. The emergent challenge is that one has to ensure a small MSE of $\mathbf{v}_t$ to guarantee convergence, where the difficulty arises from the randomness of when a snapshot gradient is computed.

An equivalent manner to describe (3) is through a sequence of i.i.d. Bernoulli random variables $\{B_t\}$ with pmf

$$\mathbb{P}(B_t = 1) = \frac{1}{m}; \ \mathbb{P}(B_t = 0) = 1 - \frac{1}{m}. \tag{4}$$

If $B_t = 1$, a snapshot gradient $\mathbf{v}_t = \nabla F(\mathbf{x}_t)$ is computed; otherwise, the estimated gradient $\mathbf{v}_t = \nabla f_{i_t}(\mathbf{x}_t) - \nabla f_{i_t}(\mathbf{x}_{t-1}) + \mathbf{v}_{t-1}$ is used for the update. Let $N_{t_1:t}$ denote the event that at iteration t the last evaluated snapshot gradient was at t_1. In other words, $N_{t_1:t}$ is equivalent to $B_{t_1} = 1, B_{t_1+1} = 0, \dots, B_t = 0$. Note that t_1 can take values from 0 (no snapshot gradient computed) to t (corresponding to $\mathbf{v}_t = \nabla F(\mathbf{x}_t)$).

The key lemma enabling our analysis is a simple probabilistic observation.

Lemma 3. *For a given t, i) events $N_{t_1:t}$ and $N_{t_2:t}$ are disjoint when $t_1 \neq t_2$; and, ii) $\sum_{t_1=0}^{t} \mathbb{P}(N_{t_1:t}) = 1$.*

The general idea is to exploit these properties of $N_{t_1:t}$ to obtain the MSE of $\mathbf{v}_t^s$, which is further leveraged to derive the convergence of L2S. Note that our idea for establishing the convergence of L2S is general enough to provide a parallel analysis for a loopless version of SVRG [Kovalev et al., 2019, Qian et al., 2019], without relying on the complicated Lyapunov function.

3.1 L2S for Convex Problems

The subject of this subsection is problems with smooth and convex losses such as those obeying Assumptions 1 and 2. We find that SARAH is challenged analytically because $\tilde{\mathbf{x}}^s \neq \mathbf{x}_{m+1}^s$ in Line 11 of Alg. 1, which necessitates SARAH's 'non-divergent' assumption. A few works have identified this issue [Nguyen et al., 2019, Wang et al., 2018, Pham et al., 2019], but require an n-dependent step size (e.g., $\eta = \mathcal{O}(\frac{1}{L\sqrt{n}})$) to address it[2]. However, n-independent step sizes are also widely adopted in practice. The key to bypassing this n-dependence in step size, is removing the inner loop of SARAH and computing snapshot gradients following a random schedule as (3).

The analysis starts with the MSE of $\mathbf{v}_t$ in L2S. All proofs are relegated to Appendix due to space limitations.

Lemma 4. *Under Assumptions 1 and 2, the following inequality holds for a given t when $\eta < 2/L$*

$$\mathbb{E}\big[\|\nabla F(\mathbf{x}_t) - \mathbf{v}_t\|^2 | N_{t_1:t}\big] \le \frac{\eta L}{2 - \eta L}\mathbb{E}\big[\|\nabla F(\mathbf{x}_{t_1})\|^2\big]. \tag{5}$$

Furthermore, we have

$$\mathbb{E}\big[\|\nabla F(\mathbf{x}_t) - \mathbf{v}_t\|^2\big] \le \frac{\eta L}{2 - \eta L}\Big(1 - \frac{1}{m}\Big)^t \|\nabla F(\mathbf{x}_0)\|^2 + \frac{\eta L}{2 - \eta L}\frac{1}{m}\sum_{\tau=1}^{t-1}\Big(1 - \frac{1}{m}\Big)^{t-\tau}\mathbb{E}\big[\|\nabla F(\mathbf{x}_\tau)\|^2\big].$$

Comparing (5) with Lemma 1 reveals that conditioning on $N_{t_1:t}$, $\mathbf{x}_{t_1}$ in L2S is similar to the starting point of an outer loop in SARAH (i.e., $\mathbf{x}_0^s$), while the following iterations $\{\mathbf{x}_\tau\}_{\tau=t_1+1}^{t}$ mimic the behavior of SARAH's inner loop. Taking expectation w.r.t. $N_{t_1:t}$ in (5), Lemma 4 further asserts that the MSE of $\mathbf{v}_t$ depends on the *exponentially moving average* of the norm square of past gradients.

Theorem 1. *If Assumptions 1 and 2 hold, and the step size is chosen such that $\eta < 1/L$ and $1 - \frac{\eta L}{2-\eta L} \ge C_\eta$, where C_η is a positive constant, the output of L2S, $\mathbf{x}_a$, is guaranteed to satisfy*

$$\mathbb{E}\big[\|\nabla F(\mathbf{x}_a)\|^2\big] = \mathcal{O}\bigg(\frac{F(\mathbf{x}_0) - F(\mathbf{x}^*)}{\eta T C_\eta} + \frac{m\eta L\|\nabla F(\mathbf{x}_0)\|^2}{T C_\eta}\bigg).$$

The constant C_η depends on the choice of η, e.g., $C_\eta = 2/3$ for $\eta = 0.5/L$. Based on Theorem 1, the convergence rates as well as the complexities under different choices of η and m are specified in the following corollaries. Let us start with a constant step size that is irrelevant with n.

Corollary 1. *Choose a constant $\eta < 1/L$. If $m = \Theta(\sqrt{n})$, then L2S has convergence rate $\mathcal{O}(\sqrt{n}/T)$ and requires $\mathcal{O}(n + n/\epsilon)$ IFO calls to find an ϵ-accurate solution.*

Corollary 2. *Choose a constant $\eta < 1/L$. If $m = \Theta(n)$, the convergence rate of L2S is $\mathcal{O}(n/T)$. The complexity to ensure an ϵ-accurate solution is $\mathcal{O}(n + n/\epsilon)$.*

In Corollaries 1 and 2, the choice of η does not depend on n. Thus, relative to SARAH, L2S eliminates the non-divergence assumption and establishes the convergence rate as well. On the other hand, an n-dependent step size is also supported by L2S, whose complexity is specified in the following corollary.

Corollary 3. *If we select $\eta = \mathcal{O}\big(\frac{1}{L\sqrt{m}}\big)$, and $m = \Theta(n)$, then L2S has convergence rate $\mathcal{O}(\sqrt{n}/T)$, and the complexity to find an ϵ-accurate solution is $\mathcal{O}(n + \sqrt{n}/\epsilon)$.*

When to Adopt n-dependent Step Sizes? An interesting observation is that though the complexity of using an n-dependent step size is lower than those of an n-independent step size in both L2S and SVRG [Reddi et al., 2016a], the numerical performances on modern datasets such as *rcv1* and *a9a* suggest that n-independent step sizes boost the convergence speed. We argue that an n-dependent step size only reveals its numerical merits when n is *extremely large*. Intuitively, a large n positively correlates with the larger MSE of the gradient estimate, which in turn calls for a smaller (n-dependent) step size. Our numerical results in Appendix D.3 also support this argument. We subsample aforementioned datasets with different values of n. SVRG and L2S are tested on these subsampled datasets. Besides the faster convergence when using an n-independent step size, it is also observed that as n increases, i) the gradient

[2] These algorithms are designed for nonconvex problems, however, even assuming convexity we are unable to show the convergence with a step size independent with n.

norm of solutions obtained by n-dependent step sizes becomes smaller; and ii) the difference on the performance gap between n-dependent and n-independent step sizes reduces.

3.2 L2S for Nonconvex Problems

The scope of L2S can also be broadened to nonconvex problems under Assumption 1, that is, L2S with a proper step size is guaranteed to use $\mathcal{O}(n+\sqrt{n}/\epsilon)$ IFO calls to find an ϵ-accurate solution. Compared with SARAH, the merit of L2S is that the extra MSE introduced by the randomized scheduling of snapshot gradient computation can be helpful for exploring the landscape of the loss function, which will be discussed in detail in Section 4. Here we focus on the convergence properties only, starting with the MSE in nonconvex settings.

Lemma 5. *If Assumption 1 holds, L2S guarantees that for a given $N_{t_1:t}$*

$$\mathbb{E}\big[\|\nabla F(\mathbf{x}_t)-\mathbf{v}_t\|^2|N_{t_1:t}\big] \le \eta^2 L^2 \sum_{\tau=t_1+1}^{t} \mathbb{E}\big[\|\mathbf{v}_{\tau-1}\|^2|N_{t_1:t}\big]. \tag{6}$$

In addition, the following inequality is true

$$\mathbb{E}\big[\|\nabla F(\mathbf{x}_t)-\mathbf{v}_t\|^2\big] \le \eta^2 L^2 \sum_{\tau=0}^{t-1}\Big(1-\frac{1}{m}\Big)^{t-\tau}\mathbb{E}\big[\|\mathbf{v}_\tau\|^2\big].$$

Conditioning on $N_{t_1:t}$, iterations $\{\mathbf{x}_\tau\}_{\tau=t_1}^{t}$ are comparable to an outer loop of SARAH. Similar to Lemma 2, the MSE upper bound of $\mathbf{v}_t$ in (6) is large when $t-t_1$ is large. If we take expectation w.r.t. the randomness of $N_{t_1:t}$, the MSE of $\mathbf{v}_t$ then depends on the exponentially moving average of the norm square of all past gradient estimates $\{\mathbf{v}_\tau\}_{\tau=0}^{t-1}$, which is different from Lemma 4 (for convex problems) where the MSE involves the past gradients $\{\nabla F(\mathbf{x}_\tau)\}_{\tau=0}^{t-1}$. It turns out that such a past-estimate-based MSE is difficult to cope with using only the exponentially deceasing sequence $\{(1-1/m)^{t-\tau}\}_{\tau=0}^{t-1}$, prompting a cautiously designed (m-dependent) η.

Theorem 2. *With Assumption 1 holding, and choosing $\eta\in(0,\frac{\sqrt{4m+1}-1}{2mL}]=\mathcal{O}\big(\frac{1}{L\sqrt{m}}\big)$, the final L2S output $\mathbf{x}_a$ satisfies*

$$\mathbb{E}\Big[\|\nabla F(\mathbf{x}_a)\|^2\Big] = \mathcal{O}\bigg(\frac{L\sqrt{m}\big[F(\mathbf{x}_0)-F(\mathbf{x}^*)\big]}{T}+\frac{\|\nabla F(\mathbf{x}_0)\|^2}{T}\bigg).$$

An intuitive explanation of the m-dependent η is that with a small m, L2S evaluates a snapshot gradient more frequently [cf. (3)], which translates to a relatively small MSE bound in Lemma 5. Given an accurate gradient estimate, it is thus reasonable to adopt a larger step size.

Algorithm 3 L2S-SC

```
Initialize: $\mathbf{x}_0$, $\eta$, $m$, $S$, and $s=0$
compute $\mathbf{v}_0=\nabla F(\mathbf{x}_0)$
$\mathbf{x}_1=\mathbf{x}_0-\eta\mathbf{v}_0$
while $s\neq S$ do
    randomly generate $B_t$ as (4)
    if $B_t=1$ then
        $\mathbf{x}_t=\mathbf{x}_{t-1}$
        $\mathbf{v}_t=\nabla F(\mathbf{x}_t),\ s=s+1$
    else
        $\mathbf{v}_t=\nabla f_{i_t}(\mathbf{x}_t)-\nabla f_{i_t}(\mathbf{x}_{t-1})+\mathbf{v}_{t-1}$
    end if
    $\mathbf{x}_{t+1}=\mathbf{x}_t-\eta\mathbf{v}_t,\ t=t+1$
end while
$T=t$
Output: $\mathbf{x}_T$
```

Corollary 4. *Selecting $\eta=\mathcal{O}\big(\frac{1}{L\sqrt{m}}\big)$ and $m=\Theta(n)$, L2S converges with rate $\mathcal{O}(\sqrt{n}/T)$, and the complexity to find an ϵ-accurate solution is $\mathcal{O}(n+\sqrt{n}/\epsilon)$.*

Almost matching the lower bound $\Omega(\sqrt{n}/\epsilon)$ of nonconvex ERM problems [Fang et al., 2018], the complexity of L2S is similar to other SARAH type algorithms [Fang et al., 2018, Wang et al., 2018, Nguyen et al., 2019]. The slight suboptimality is due to the n extra IFO calls involved in computing $\mathbf{v}_0$.

3.3 L2S for Strongly Convex Problems

In addition to convex and nonconvex problems, a modified version of L2S that we term **L2S** for **S**trongly **C**onvex problems (L2S-SC), converges linearly under Assumptions 1 – 3. As we have seen previously, L2S is closely related to SARAH, especially when conditioned on a given $N_{t_1:t}$. Hence, we will first state a useful property of SARAH that will guide the design and analysis of L2S-SC.

Lemma 6. *Consider SARAH (Alg. 1) with Line 11 replaced by $\tilde{\mathbf{x}}^s=\mathbf{x}_m^s$. Choosing $\eta<2/(3L)$ and m large enough such that*

$$\lambda_m := \frac{2\eta L}{2-\eta L}+\big(2+2\eta L\big)(\theta)^m<1,$$

where θ is defined as

$$\theta=\begin{cases} 1-\Big(\frac{2}{\eta L}-1\Big)\mu^2\eta^2 & \text{with As. 1 – 3} \\ 1-\frac{2\eta L}{1+\kappa} & \text{with As. 1 and 4}\end{cases}. \tag{7}$$

The modified SARAH is guaranteed to converge linearly; that is,

$$\mathbb{E}\big[\|\nabla F(\tilde{\mathbf{x}}^s)\|^2\big]\le\lambda_m\mathbb{E}\big[\|\nabla F(\tilde{\mathbf{x}}^{s-1})\|^2\big].$$

As opposed to the random draw of $\tilde{\mathbf{x}}^s$ (Line 11 of Alg. 1), Lemma 6 asserts that by properly choosing η and m, setting $\tilde{\mathbf{x}}^s=\mathbf{x}_m^s$ preserves the linear convergence of SARAH.

Note that the convergence with last iteration of SARAH was also studied by [Liu et al.] under Assumptions 1 – 3. However, their analysis requires an undesirably small step size, i.e., $\eta = \mathcal{O}(\mu/L^2)$, while ours enables a much larger one $\eta = \mathcal{O}(1/L)$.

Remark 1. Through Lemma 6 one can establish the complexity of SARAH with $\tilde{\mathbf{x}}^s = \mathbf{x}_m^s$. When Assumptions 1 - 3 hold, the complexity is $\mathcal{O}\big((n+\kappa^2)\ln\frac{1}{\epsilon}\big)$, which is on the same order of SVRG with last iteration [Tan et al., 2016, Hu et al., 2018]. However, when Assumptions 1 and 4 are true, the complexity of SARAH decreases to $\mathcal{O}\big((n+\kappa)\ln\frac{1}{\epsilon}\big)$. This is the property SVRG does not exhibit.

L2S-SC is summarized in Alg. 3, where $\mathbf{v}_t$ obtained in Lines 5 - 11 is a rewrite of (3) using B_t introduced in (4) for the ease of presentation and analysis. L2S-SC differs from L2S in that when $B_t = 1$, $\mathbf{x}_t$ steps back slightly as in Line 7. This "step back" is to allow for a rigorous analysis, and can be viewed as the counterpart of choosing $\tilde{\mathbf{x}}^s = \mathbf{x}_m^s$ instead of $\mathbf{x}_{m+1}^s$ as in Lemma 6. Omitting Line 7 in practice does not deteriorate performance. In addition, the parameter S required to initialize L2S is comparable to the number of outer loops of SARAH, as one can also validate through the S dependence in the linear convergence rate.

Theorem 3. *Choose $\eta < 2/(3L)$ and m large enough such that*

$$\lambda := \frac{2\eta L}{2-\eta L} + \frac{2+2\eta L}{m-1}\frac{\theta(1-\frac{1}{m})}{1-\theta(1-\frac{1}{m})} < 1$$

where θ is defined in (7). *L2S-SC in Alg. 3 guarantees*

$$\mathbb{E}\big[\|\nabla F(\mathbf{x}_T)\|^2\big] \leq \lambda^S \|\nabla F(\mathbf{x}_0)\|^2.$$

The complexities of L2S-SC under different assumptions are established in the next corollaries.

Corollary 5. *Choose $\eta < 2/(3L)$ and $m = \Theta(\kappa^2)$. When Assumptions 1 – 3 hold, the complexity of L2S to find an ϵ-accurate solution is $\mathcal{O}\big((n+\kappa^2)\ln\frac{1}{\epsilon}\big)$.*

Corollary 6. *Choose $\eta < 2/(3L)$ with $m = \Theta(\kappa)$. When Assumptions 1 and 4 hold, the complexity of L2S to find an ϵ-accurate solution is $\mathcal{O}\big((n+\kappa)\ln\frac{1}{\epsilon}\big)$.*

4 DISCUSSIONS

4.1 Comparison with SCSG

L2S can be viewed as SARAH with variable inner loop length. A variant of SVRG (abbreviated as SCSG) with randomized inner loop length has been also developed in [Lei and Jordan, 2017, 2019]. A close relative of SCSG is a loopless version of SVRG [Kovalev et al., 2019, Qian et al., 2019]. Unfortunately, the analysis in [Kovalev et al., 2019] is confined to strongly convex problems, while [Qian et al., 2019] relies on different analyzing schemes that are more involved. The key differences between L2S and SCSG are as follows.

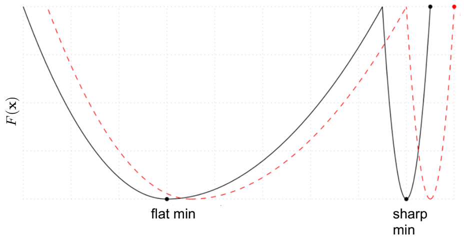


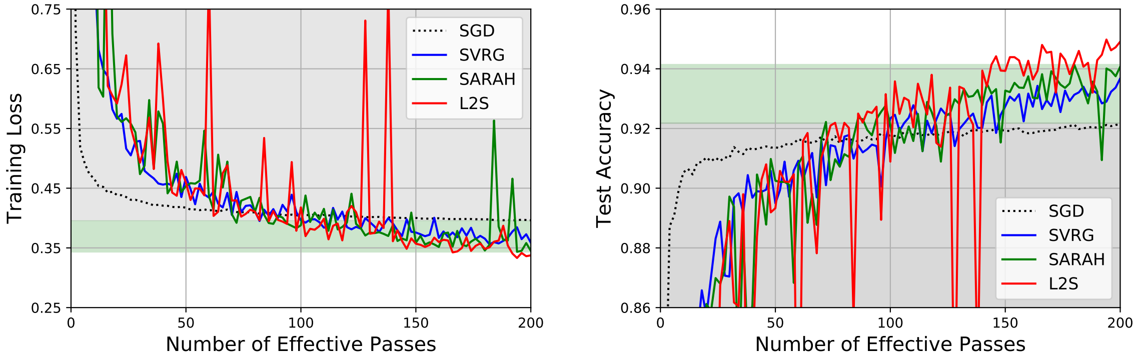
Figure 1: An illustration of sharp and flat minima [Keskar et al., 2016]. The black line is the loss curvature associated with training data; and the red line represents the loss for testing data which slightly deviates from the training loss. The sensitivity of the training function at a sharp minimum degrades its generalization performance.

d1) The random inner loop length of SCSG is assumed geometrically distributed (at least for the analysis) that could be even infinite. Thus, its total number of iterations is also random. In contrast, the total number of L2S iterations is fixed to $T+1$. This is accomplished through a non-geometrically distributed equivalent inner loop length.

d2) The analyses are also different. From a high level, SCSG employs a “forward” analysis, where an iteration t that computes a snapshot gradient is fixed first, and then future iterations $t+1, t+2$ till the computation of the next snapshot gradient are checked; while our analysis takes the “backward” route, that is, after fixing an iteration t the past iterations $t-1, t-2, \ldots, 0$ are checked for a snapshot gradient computation. As a consequence, our “backward” analysis leads to an exponentially moving average structure in the MSE (Lemmas 4 and 5), which is insightful, and is not provided by SCSG.

4.2 Generalization Merits of L2S

SARAH has well-documented merits for its complexity for nonconvex problems, but similar to other variance reduction algorithms, it is not as successful as expected for training neural networks. We conjecture this is related with the reduced MSE of the gradient estimates. To see this, although there is no consensus on analytical justification for this, empirical evidence points out that SGD with large mini-batch size (needed to reduce the variance of the gradient estimates) tends to converge to a sharp minimum. Sharp minima are believed to have worse generalization properties compared with flat ones [Keskar et al., 2016]. Fig. 1 shows that gradient estimators with pronounced variability are more agile to explore the space and escape from a *sharp minimum*, while flat minima are more tolerant to larger variability. This suggests that the gradient estimator could be designed to control its exploration ability, which can in turn improve generalization. Being able to explore



(a)

(b)

Figure 2: (a) Training loss of L2S; (b) Test accuracy of L2S.

the loss function landscape is critical because deepening and widening a neural network does not always endow the stochastic gradient estimator with controllable exploration ability [Defazio and Bottou, 2018].

These empirical results shed light on the important role of exploration ability in the gradient estimates. A natural means of controlling this exploration in algorithms with variance reduction, is to add zero-mean noise in the gradient estimates. However, the issue is that even for convex problems, the convergence rate slows down if the noise is not carefully calibrated; see e.g. [Kulunchakov and Mairal, 2019]. Carefully designed noises for *escaping saddle points* rather than *generalization merits* were studied in e.g., [Jin et al., 2017, Fang et al., 2019]. But even for saddle point escaping, extra information of the loss landscape, e.g., Hessian Lipchitz constant is required for obtaining the variance of injected noise. Unfortunately, the Hessian Lipchitz constant is not always available in advance. To control the exploration ability of algorithms with variance reduction, L2S resorts to randomized snapshot gradient computation that is free of extra knowledge for the loss landscape.

With SARAH as a reference, we can see how our randomized snapshot gradient computation in L2S can benefit variability for exploration. Let $t_2 - t_1$ denote the equivalent length of a L2S inner loop, where t_1 and t_2 are the indices of two consecutive iterations when snapshot gradients are computed. Recall from Lemma 2 that the MSE of $\mathbf{v}_t$ tends to be larger as t approaches t_2. Relative to SARAH, this means that the randomized computation of the snapshot gradient increases the MSE when it so happens that $t_1 + m < t < t_2$.

Test of L2S on Neural Networks. We perform classification on MNIST dataset[3] using a $784 \times 128 \times 10$ feedforward neural network with sigmoid activation function. The network is trained for 200 epochs and the training loss and test accuracy are plotted in Fig. 2. The bound of gray shadowed area indicates the smallest training loss (highest test accuracy) of SGD, while the bound of green shadowed area represents the best performances for SARAH. Figs. 2 (a) and (b) share some common patterns: i) SGD converges much faster in the initial phase compared with variance reduced algorithms; ii) the fluctuate of L2S is larger than that of SARAH, implying the randomized full gradient computation indeed introduces extra chances for exploration; and, iii) when x-axis is around 140, L2S begins to outperform SARAH while in previous epochs their performances are comparable. Note that *before* L2S outperforms SARAH, there is a deep drop on its accuracy. This can be explained as that L2S explores for a local minimum with generalization merits thanks to the randomized snapshot gradient computation and the deep drop in Fig. 2 (b) indicates the transition from a local min to another.

5 NUMERICAL TESTS

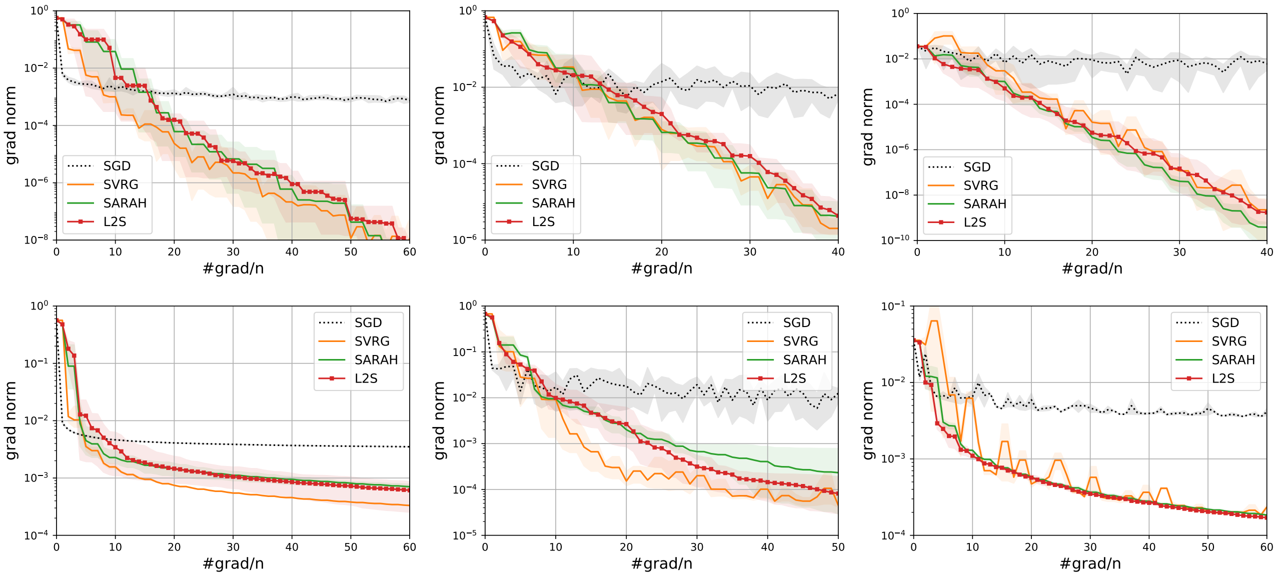
Besides training neural networks, we also apply L2S to logistic regression to showcase the performances in strongly convex and convex cases. Specifically, consider the loss function

$$F(\mathbf{x}) = \frac{1}{n}\sum_{i\in[n]} \ln\left[1 + \exp(-b_i\langle \mathbf{a}_i, \mathbf{x}\rangle)\right] + \frac{\lambda}{2}\|\mathbf{x}\|^2 \quad (8)$$

where $(\mathbf{a}_i, b_i)$ is the (feature, label) pair of datum i. Problem (8) can be rewritten in the format of (1) with $f_i(\mathbf{x}) = \ln\left[1 + \exp(-b_i\langle \mathbf{a}_i, \mathbf{x}\rangle)\right] + \frac{\lambda}{2}\|\mathbf{x}\|^2$. One can verify that in this case Assumptions 1 and 4 are satisfied. Datasets *a9a*, *w7a* and *rcv1.binary*[4] are used in numerical tests presented. Details regarding the datasets and implementation are deferred to Appendix F.

[3] Online available at http://yann.lecun.com/exdb/mnist/

[4] All datasets are from LIBSVM, which is online available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html.



(a) *w7a* (b) *a9a* (c) *rcv1*

Figure 3: Tests of L2S on strongly convex problems (first row) and convex ones (second row) on different datasets.

Test of L2S-SC on Strongly Convex Problems. The performance of L2S-SC is shown in the first row of Fig. 3. SVRG, SARAH and SGD are chosen as benchmarks, where SGD is with modified step size $\eta_k = 1/\big(L(k+1)\big)$ on the k-th effective sample pass. For both SARAH and SVRG, the length of inner loop is chosen as $m = n$. We tune step size and only report the best performance. For a fair comparison we set η and m for L2S the same as SARAH. It can be seen that on datasets *w7a* and *rcv1* L2S-SC has comparable performances with SARAH, while on dataset *a9a*, L2S-SC has similar performance with SARAH. The simulations validate the theoretical results of L2S-SC.

Test of L2S on Convex Problems. The performances of L2S for convex problems ($\lambda = 0$) is listed in the second row of Fig. 3. Again SVRG, SARAH and SGD are adopted as benchmarks. We choose $m = n$ for SVRG, SARAH and L2S. It can be seen that on datasets *w7a* and *rcv1* L2S performs almost the same as SARAH, while outperforms SARAH on dataset *a9a*. Note that the performance of SVRG improves over SARAH on certain datasets. This is because a theoretically unsupported step size ($\eta > 1/(4L)$) is used in SVRG for best empirical performance.

6 CONCLUSIONS AND FUTURE WORK

A unifying framework, L2S, was introduced to efficiently solve (strongly) convex and nonconvex ERM problems. The complexities to find an ϵ-accurate solution were established. Numerical tests validated our theoretical findings.

An interesting question is how to extend L2S and SARAH to stochastic optimization, i.e., solving $\min_{\mathbf{x}} \mathbb{E}_\xi[f(\mathbf{x},\xi)]$, where ξ is a random variable whose distribution is unknown. Such problems can be addressed using SVRG or SCSG; see e.g., [Lei et al., 2017]. Works such as [Nguyen et al., 2018b] is the first attempt for solving stochastic optimization via SARAH. Though addressing certain challenges, the remaining issue is that the gradient estimate is in general not implementable on problems other than ERM. To see this, recall the SARAH based gradient estimate for stochastic optimization is $\mathbf{v}_t = \nabla f(\mathbf{x}_t,\xi_t) - \nabla f(\mathbf{x}_{t-1},\xi_t) + \mathbf{v}_{t-1}$, where ξ_t is the t-th realization of ξ. Obtaining $\nabla f(\mathbf{x}_{t-1},\xi_t)$ via a stochastic first order oracle can be impossible especially when ξ comes from an unknown continuous probability space. To overcome this challenge is included in our research agenda.

Acknowledgements

The authors would like to thank Lingda Wang for helpful discussions, and the anonymous reviewers for their constructive feedback. We also gratefully acknowledge the support from NSF grants 1508993, 1509040, 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.

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *Proc. Intl. Conf. on Machine Learning*, pages 699–707, New York City, NY, 2016.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *arXiv preprint arXiv:1812.04529*, 2018.

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.

Cong Fang, Zhouchen Lin, and Tong Zhang. Sharp analysis for nonconvex sgd escaping from saddle points. *arXiv preprint arXiv:1902.00247*, 2019.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

John A Gubner. *Probability and random processes for electrical and computer engineers*. Cambridge University Press, 2006.

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.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *Proc. Intl. Conf. on Machine Learning*, pages 1724–1732, 2017.

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.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Jakub Konecnỳ and Peter Richtárik. Semi-stochastic gradient descent methods. *arXiv preprint arXiv:1312.1666*, 2013.

Dmitry Kovalev, Samuel Horvath, and Peter Richtarik. 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. *arXiv preprint arXiv:1905.02374*, 2019.

Lihua Lei and Michael Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, pages 148–156, Fort Lauderdale, Florida, 2017.

Lihua Lei and Michael I Jordan. On the adaptivity of stochastic gradient-based optimization. *arXiv preprint arXiv:1904.04480*, 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, 2017.

Bingcong Li, Lingda Wang, and Georgios B Giannakis. Almost tune-free variance reduction. *arXiv preprint arXiv:1908.09345*, 2019.

Yan Liu, Congying Han, and Tiande Guo. A class of stochastic variance reduced methods with an adaptive stepsize. 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.

Lam M Nguyen, Katya Scheinberg, and Martin Takáč. Inexact sarah algorithm for stochastic optimization. *arXiv preprint arXiv:1811.10105*, 2018a.

Lam M Nguyen, Katya Scheinberg, and Martin Takáč. Inexact sarah algorithm for stochastic optimization. *arXiv preprint arXiv:1811.10105*, 2018b.

Lam M Nguyen, Marten van Dijk, Dzung T Phan, Phuong Ha Nguyen, Tsui-Wei Weng, and Jayant R Kalagnanam. Optimal finite-sum smooth nonconvex optimization with SARAH. *arXiv preprint arXiv:1901.07648*, 2019.

Nhan H Pham, Lam M Nguyen, Dzung T Phan, and Quoc Tran-Dinh. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *arXiv preprint arXiv:1902.05679*, 2019.

Xun Qian, Zheng Qu, and Peter Richtarik. L-SVRG and L-Katyusha with arbitrary sampling. *arXiv preprint arXiv:1901.08689*, 2019.

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, 2016a.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Fast incremental method for nonconvex optimization. *arXiv preprint arXiv:1603.06159*, 2016b.

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. volume 14, pages 567–599, 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, 2016.

Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. SpiderBoost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv preprint arXiv:1810.10690*, 2018.

Jingzhao Zhang, Hongyi Zhang, and Suvrit Sra. R-spider: A fast riemannian stochastic optimization algorithm with curvature independent rate. *arXiv preprint arXiv:1811.04194*, 2018.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduction for nonconvex optimization. In *Proc. Advances in Neural Info. Process. Syst.*, pages 3925–3936, 2018.