Stochastic Variance Reduction for Nonconvex Optimization

Sashank J. Reddi† SJAKKAMR@CS.CMU.EDU
Ahmed Hefny† AHEFNY@CS.CMU.EDU
Suvrit Sra^ SUVRIT@MIT.EDU
Barnabás Póczós† BAPOCZOS@CS.CMU.EDU
Alex Smola† ALEX@SMOLA.ORG
† Machine Learning Department, School of Computer Science, Carnegie Mellon University
^ Laboratory for Information & Decision Systems, Massachusetts Institute of Technology

Abstract

We study nonconvex finite-sum problems and analyze stochastic variance reduced gradient (SVRG) methods for them. SVRG and related methods have recently surged into prominence for convex optimization given their edge over stochastic gradient descent (SGD); but their theoretical analysis almost exclusively assumes convexity. In contrast, we obtain non-asymptotic rates of convergence of SVRG for nonconvex optimization, showing that it is provably faster than SGD and gradient descent. We also analyze a subclass of nonconvex problems on which SVRG attains linear convergence to the *global* optimum. We extend our analysis to mini-batch variants, showing (theoretical) linear speedup due to minibatching in parallel settings.

1. Introduction

We study nonconvex *finite-sum* problems of the form

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{n}\sum\nolimits_{i=1}^{n} f_i(x), \qquad (1)$$

where both f and f_i ($i \in [n]$) may be nonconvex and have Lipschitz continuous gradients. We denote the class of such finite-sum Lipschitz smooth functions by $\mathcal{F}_n$. We optimize functions in $\mathcal{F}_n$ in the Incremental First-order Oracle (IFO) framework (Agarwal & Bottou, 2014) defined below.

Definition 1. *For $f \in \mathcal{F}_n$, an IFO takes an index $i \in [n]$ and a point $x \in \mathbb{R}^d$, and returns the pair $(f_i(x), \nabla f_i(x))$.*

IFO based analysis was introduced to study lower bounds for finite-sum problems. Algorithms that use IFOs are favored in large-scale applications as they, usually, require only a small amount of first-order information at each iteration. Two fundamental models in machine learning that profit from IFO algorithms are (i) empirical risk minimization, which typically uses convex finite-sum models; and (ii) deep learning, which uses nonconvex ones.

The prototypical IFO algorithm, stochastic gradient descent (SGD),[1] has witnessed tremendous progress in the recent years. By now a variety of accelerated, parallel, and faster converging variants are known. Among these, of particular importance are variance reduced (VR) stochastic methods (Schmidt et al., 2013; Johnson & Zhang, 2013; Defazio et al., 2014a), which have delivered progress such as linear convergence rates (for strongly convex functions) as opposed to sublinear rates of ordinary SGD (Robbins & Monro, 1951; Nemirovski et al., 2009). Similar (but not same) benefits of VR methods can also be seen in smooth convex functions. The SVRG algorithm of (Johnson & Zhang, 2013) is particularly attractive here because of its low storage requirement in comparison to the algorithms in (Schmidt et al., 2013; Defazio et al., 2014a).

Despite the meteoric rise of VR methods, their analysis for general nonconvex problems is largely missing. Johnson & Zhang (2013) remark on convergence of SVRG when $f \in \mathcal{F}_n$ is locally strongly convex and provide compelling experimental results (Fig. 4 in (Johnson & Zhang, 2013)). However, problems encountered in practice are typically not even locally convex, let alone strongly convex. The current analysis of SVRG does not extend to nonconvex functions as it relies heavily on convexity for controlling the variance. Given the dominance of stochastic gradient methods in optimizing deep neural nets and other large nonconvex models, theoretical investigation of faster nonconvex stochastic methods is much needed.

Convex VR methods are known to enjoy the faster convergence rate of batch gradient descent (GRADDESCENT) but

Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

[1] We use 'incremental gradient' and 'stochastic gradient' interchangeably, though we are only interested in finite-sum problems.

Algorithm	Nonconvex	Convex	Gradient Dominated	Fixed Step Size?
SGD	$O\left(1/\epsilon^2\right)$	$O\left(1/\epsilon^2\right)$	$O\left(1/\epsilon^2\right)$	×
GradientDescent	$O\left(n/\epsilon\right)$	$O\left(n/\epsilon\right)$	$O\left(n\tau\log(1/\epsilon)\right)$	√
SVRG	$O\big(n + (n^{2/3}/\epsilon)\big)$	$O\big(n + (\sqrt{n}/\epsilon)\big)$	$O\big((n + n^{2/3}\tau)\log(1/\epsilon)\big)$	√
MSVRG	$O\big(\min\left\{1/\epsilon^2, n^{2/3}/\epsilon\right\}\big)$	$O\big(\min\{1/\epsilon^2, \sqrt{n}/\epsilon\}\big)$	–	×

Table 1. Table comparing the *best* IFO complexity of different algorithms discussed in the paper. The complexity is measured in terms of the number of oracle calls required to achieve an ϵ-accurate solution (see Definition 2). Here, by fixed step size, we mean that the step size of the algorithm is fixed and does not depend on ϵ (or alternatively on T, the total number of iterations). The complexity of gradient dominated functions refers to the number of IFO calls required to obtain an ϵ-accurate solution for a τ-gradient dominated function (see Sec. 2 for the definition). For SGD, we are not aware of any specific results for gradient dominated functions. Also, we hide the dependence of IFO complexity on the Lipschitz constant L (see Section 2), $[f(x^0) - f(x^*)]$ and $\|x^0 - x^*\|$ (where x^0 is the initial point and x^* is an optimal solution to (1)) to make a cleaner comparison. The results marked in red are the contributions of this paper.

with a much weaker dependence on n, without compromising the rate like SGD. However, it is not clear if these benefits carry beyond convex problems, prompting the central question of this paper:

> *For nonconvex functions in $\mathcal{F}_n$, can one obtain convergence rates faster than both* SGD *and* GradDescent *using an IFO?*

Perhaps surprisingly, we provide an affirmative answer to this question and show how a careful selection of parameters in SVRG indeed yields faster convergence rates.

Main Contributions. We summarize our main contributions below and also list the key results in Table 1.

- We analyze nonconvex stochastic variance reduced gradient (SVRG), and prove that it has faster rates of convergence than GradDescent and ordinary SGD. We show that SVRG is faster than GradDescent by a factor of $n^{1/3}$ (see Table 1).
- We provide new theoretical insights into the interplay between step-size, iteration complexity and convergence of nonconvex SVRG (see Corr. 2).
- We analyze mini-batch nonconvex SVRG and show that it provably benefits from mini-batching. Specifically, we show theoretical linear speedups in parallel settings for large mini-batch sizes. By using a mini-batch of size b $(< n^{2/3})$, we show that mini-batch nonconvex SVRG is faster by a factor of b (Thm. 6). We are not aware of any prior work on mini-batch first-order stochastic methods that shows linear speedup in parallel settings for nonconvex optimization.
- For an interesting nonconvex subclass of $\mathcal{F}_n$ called gradient dominated functions (Polyak, 1963; Nesterov & Polyak, 2006), we propose a variant of SVRG that attains a *global* linear rate of convergence. We improve upon many prior results for this subclass of functions (see Section 3.1). To the best of our knowledge, ours is the first work that shows a stochastic method with linear convergence for gradient dominated functions.
- Our analysis yields as a byproduct a direct convergence analysis for SVRG for smooth convex functions (Sec. 4).
- We examine a variant of SVRG (called MSVRG) that has faster rates than both SGD and GradDescent.

Concurrent to our work, Allen-Zhu & Hazan (2016) have also obtained an SVRG-based $O(n^{1/3})$ improvement over GradDescent. However, both our *algorithm* and *analysis* are somewhat simpler; our analysis also yields better minibatching with speedups linear in b, and an interesting hybrid variant MSVRG. Moreover, we also provide global linear convergence rate analysis of SVRG for the class of gradient-dominated functions.

1.1. Other Related Work

Convex. Bertsekas (2011) surveys several incremental gradient methods for convex problems. A key reference for stochastic convex optimization (for $\min \mathbb{E}_z[F(x,z)]$) is (Nemirovski et al., 2009). Faster rates of convergence are attained for problems in $\mathcal{F}_n$ by VR methods, see e.g., (Defazio et al., 2014a; Johnson & Zhang, 2013; Schmidt et al., 2013; Konečný et al., 2015; Shalev-Shwartz & Zhang, 2013; Defazio et al., 2014b). Asynchronous VR frameworks are developed in (Reddi et al., 2015). Agarwal & Bottou (2014); Lan & Zhou (2015) study lower-bounds for convex finite-sum problems. Shalev-Shwartz (2015) prove linear convergence of stochastic dual coordinate ascent when the individual f_i $(i \in [n])$ are nonconvex but f is strongly convex. They do not study the general nonconvex case. Moreover, even in their special setting our results improve upon theirs for the high condition number regime.

Nonconvex. SGD dates at least to the seminal work (Robbins & Monro, 1951); and since then it has been developed in several directions (Poljak & Tsypkin, 1973; Ljung, 1977; Bottou, 1991; Kushner & Clark, 2012). In the (nonsmooth) finite-sum setting, Sra (2012) considers proximal splitting methods, and analyzes asymptotic convergence (without rates) with nonvanishing gradient errors.

The first nonasymptotic convergence (with rate) analysis for SGD is in (Ghadimi & Lan, 2013), who show $O(1/\epsilon^2)$ convergence for SGD. A similar rate for parallel and distributed SGD was shown in (Lian et al., 2015). GRADDESCENT's $O(1/\epsilon)$ convergence is well-known (Nesterov, 2003, Chap. 1.2.3). The first analysis of nonconvex SVRG is due to Shamir (2014), who considers the special problem of computing a few top eigenvectors (e.g., for PCA); see also (Shamir, 2015). As sequels to this paper, we now also have extensions to *nonconvex* SAGA and *proximal nonsmooth nonconvex* VR methods (Reddi et al., 2016a;b).

2. Background & Problem Setup

We say f is *L-smooth* if there is a constant L such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall\, x, y \in \mathbb{R}^d.$$

Throughout, we assume that the f_i are L-smooth, so that $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$ ($i \in [n]$). Such an assumption is common in the analysis of first-order methods. We say f is *λ-strongly convex* if there is a $\lambda \geq 0$, such that

$$f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle + \tfrac{\lambda}{2}\|x - y\|^2 \quad \forall x, y \in \mathbb{R}^d.$$

The quantity $\kappa := L/\lambda$ is called the *condition number* of f, whenever f is L-smooth and λ-strongly convex. We say f is non-strongly convex when f is 0-strongly convex.

We also recall the class of gradient dominated functions (Polyak, 1963; Nesterov & Polyak, 2006), where a function f is called *τ-gradient dominated* if for any $x \in \mathbb{R}^d$

$$f(x) - f(x^*) \leq \tau\|\nabla f(x)\|^2, \tag{2}$$

where x^* is a global minimizer of f. Note that such a function f need not be convex. It is also easy to show that a λ-strongly convex function is $1/2\lambda$-gradient dominated.

We analyze convergence rates for the above classes of functions. Following Nesterov (2003); Ghadimi & Lan (2013) we use $\|\nabla f(x)\|^2 \leq \epsilon$ to judge when is iterate x approximately stationary. Contrast this with SGD for convex f, where one uses $[f(x) - f(x^*)]$ or $\|x - x^*\|^2$ as a convergence criteria. Unfortunately, such criteria cannot be used for nonconvex functions due to the hardness of the problem. While the quantities $\|\nabla f(x)\|^2$ and $f(x) - f(x^*)$ or $\|x - x^*\|^2$ are not comparable in general (see (Ghadimi & Lan, 2013)), they are typically assumed to be of similar magnitude. Throughout our analysis, we do *not* assume n to be constant, and report dependence on it in our results. For our analysis, we need the following definition.

Definition 2. *A point x is called ϵ-accurate if $\|\nabla f(x)\|^2 \leq \epsilon$. A stochastic iterative algorithm is said to achieve ϵ-accuracy in t iterations if $\mathbb{E}[\|\nabla f(x^t)\|^2] \leq \epsilon$, where the expectation is over the stochasticity of the algorithm.*

We measure the efficiency of the algorithms in terms of the number of IFO calls made by the algorithm (IFO complexity) to achieve an ϵ-accurate solution. Throughout the paper, we hide the dependence of IFO complexity on the Lipschitz constant L, and the initial point (in terms of $\|x^0 - x^*\|^2$ and $f(x^0) - f(x^*)$) for a clean comparison. We introduce one more definition, useful in the analysis of SGD methods for bounding the variance.

Definition 3. *We say $f \in \mathcal{F}_n$ has σ-bounded gradients if $\|\nabla f_i(x)\| \leq \sigma$ for all $i \in [n]$ and $x \in \mathbb{R}^d$.*

2.1. Nonconvex SGD: Convergence Rate

Stochastic gradient descent (SGD) is one of the simplest algorithms for solving (1). The update at the t^{th} iteration of SGD is of the following form:

$$x^{t+1} = x^t - \eta_t \nabla f_{i_t}(x). \tag{SGD}$$

By using a uniformly randomly chosen (with replacement) index i_t from $[n]$, SGD uses an unbiased estimate of the gradient at each iteration. Under appropriate conditions, Ghadimi & Lan (2013) establish convergence rate of SGD to a stationary point of f. Their results include the following theorem.

Theorem 1. *Suppose $f \in \mathcal{F}_n$ has σ-bounded gradients; let $\eta_t = \eta = c/\sqrt{T}$ where $c = \sqrt{\frac{2(f(x^0)-f(x^*))}{L\sigma^2}}$, and x^* is an optimal solution to (1). Then, the iterates of SGD satisfy*

$$\min_{0 \leq t \leq T-1} \mathbb{E}[\|\nabla f(x^t)\|^2] \leq \sqrt{\frac{2(f(x^0) - f(x^*))L}{T}}\sigma.$$

For completeness we present a proof in the appendix. Note that our choice of step size η requires knowing the total number of iterations T in advance. A more practical approach is to use a $\eta_t \propto 1/\sqrt{t}$ or $1/t$. A bound on IFO calls made by SGD follows as a corollary of Thm. 1.

Corollary 1. *For a function $f \in \mathcal{F}_n$ with σ-bounded gradient, the IFO complexity of SGD to obtain an ϵ-accurate solution is $O(1/\epsilon^2)$.*

As seen in Thm. 1, SGD has a convergence rate of $O(1/\sqrt{T})$. This rate is not improvable in general, even when the function is (non-strongly) convex (Nemirovski & Yudin, 1983). This barrier is due to the variance introduced by the stochasticity of the gradients.

3. Nonconvex SVRG

We now turn our focus to variance reduced methods. We use SVRG (Johnson & Zhang, 2013), an algorithm recently

Algorithm 1 SVRG$\big(x^0, T, m, \{p_i\}_{i=0}^m, \{\eta_i\}_{i=0}^{m-1}\big)$

```
Input: $\tilde{x}^0 = x_m^0 = x^0 \in \mathbb{R}^d$, epoch length $m$, step sizes $\{\eta_i > 0\}_{i=0}^{m-1}$, $S = \lceil T/m \rceil$, discrete probability distribution $\{p_i\}_{i=0}^m$
for $s = 0$ to $S - 1$ do
    $x_0^{s+1} = x_m^s$
    $g^{s+1} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{x}^s)$
    for $t = 0$ to $m - 1$ do
        Uniformly randomly pick $i_t$ from $\{1, \dots, n\}$
        $v_t^{s+1} = \nabla f_{i_t}(x_t^{s+1}) - \nabla f_{i_t}(\tilde{x}^s) + g^{s+1}$
        $x_{t+1}^{s+1} = x_t^{s+1} - \eta_t v_t^{s+1}$
    end for
    $\tilde{x}^{s+1} = \sum_{i=0}^m p_i x_i^{s+1}$
end for
Output: Iterate $x_a$ chosen uniformly random from $\{\{x_t^{s+1}\}_{t=0}^{m-1}\}_{s=0}^{S-1}$.
```

shown to be very effective for reducing variance in convex problems. As a result, it has gained considerable interest in both machine learning and optimization communities. We seek to understand its benefits for *nonconvex* optimization. Alg. 1 presents SVRG's pseudocode.

Observe that Alg. 1 operates in epochs. At the end of epoch s, a full gradient is calculated at the point $\tilde{x}^s$, requiring n calls to the IFO. Within its inner loop SVRG performs m stochastic updates. The total number of IFO calls for each epoch is thus $\Theta(m+n)$. For $m = 1$, the algorithm reduces to the classic GRADDESCENT algorithm. Suppose m is chosen to be $O(n)$ (typically used in practice), then the total IFO calls per epoch is $\Theta(n)$. To enable a fair comparison with SGD, we assume that the total number of inner iterations across all epochs in Alg. 1 is T. Also note a simple but important implementation detail: as written, Alg. 1 requires storing all the iterates x_t^{s+1} $(0 \le t \le m)$. This storage can be avoided by keeping a running average with respect to the probability distribution $\{p_i\}_{i=0}^m$.

Alg. 1 attains linear convergence for strongly convex f (Johnson & Zhang, 2013); for non-strongly convex functions, rates faster than SGD can be shown by using an *indirect* perturbation argument—see e.g., (Konečný & Richtárik, 2013; Xiao & Zhang, 2014).

We first state an intermediate result for the iterates of nonconvex SVRG. To ease exposition, we define

$$\Gamma_t = \Big(\eta_t - \frac{c_{t+1}\eta_t}{\beta_t} - \eta_t^2 L - 2c_{t+1}\eta_t^2\Big), \qquad (3)$$

for some parameters c_{t+1} and β_t (to be defined shortly).

Our first main result is the following theorem that provides convergence rate of Alg. 1.

Theorem 2. *Let $f \in \mathcal{F}_n$. Let $c_m = 0$, $\eta_t = \eta > 0$, $\beta_t = \beta > 0$, and $c_t = c_{t+1}(1 + \eta\beta + 2\eta^2 L^2) + \eta^2 L^3$ such that $\Gamma_t > 0$ for $0 \le t \le m-1$. Define the quantity $\gamma_n := \min_t \Gamma_t$. Further, let $p_i = 0$ for $0 \le i < m$ and $p_m = 1$, and let T be a multiple of m. Then for the output x_a of Alg. 1 we have*

$$\mathbb{E}[\|\nabla f(x_a)\|^2] \le \frac{f(x^0) - f(x^*)}{T\gamma_n},$$

where x^* is an optimal solution to (1).

Furthermore, we can also show that nonconvex SVRG exhibits expected descent (in objective) after every epoch. The condition that T is a multiple of m is solely for convenience and can be removed by slight modification of the theorem statement. Note that the value γ_n above can depend on n. To obtain an explicit dependence, we simplify it using specific choices for η and β, as formalized below.

Theorem 3. *Suppose $f \in \mathcal{F}_n$. Let $\eta = \mu_0/(Ln^\alpha)$ $(0 < \mu_0 < 1$ and $0 < \alpha \le 1)$, $\beta = L/n^{\alpha/2}$, $m = \lfloor n^{3\alpha/2}/(3\mu_0)\rfloor$ and T is some multiple of m. Then there exists universal constants $\mu_0, \nu > 0$ such that we have the following: $\gamma_n \ge \frac{\nu}{Ln^\alpha}$ in Thm. 2 and*

$$\mathbb{E}[\|\nabla f(x_a)\|^2] \le \frac{Ln^\alpha[f(x^0) - f(x^*)]}{T\nu},$$

where x^* is an optimal solution to the problem in (1) and x_a is the output of Alg. 1.

By rewriting the above result in terms IFO calls, we get the following general corollary for nonconvex SVRG.

Corollary 2. *Suppose $f \in \mathcal{F}_n$. Then the IFO complexity of Alg. 1 (with parameters from Thm. 3) for achieving an ϵ-accurate solution is:*

$$\textit{IFO calls} = \begin{cases} O\left(n + (n^{1-\frac{\alpha}{2}}/\epsilon)\right), & \text{if } \alpha < 2/3, \\ O\left(n + (n^\alpha/\epsilon)\right), & \text{if } \alpha \ge 2/3. \end{cases}$$

Corr. 2 shows the interplay between step size and the IFO complexity. We observe that the number of IFO calls is minimized in Corr. 2 when $\alpha = 2/3$. This gives rise to the following key results of the paper.

Corollary 3. *Suppose $f \in \mathcal{F}_n$. Let $\eta = \mu_1/(Ln^{2/3})$ $(0 < \mu_1 < 1)$, $\beta = L/n^{1/3}$, $m = \lfloor n/(3\mu_1)\rfloor$ and T is some multiple of m. Then there exists universal constants $\mu_1, \nu_1 > 0$ such that we have the following: $\gamma_n \ge \frac{\nu_1}{Ln^{2/3}}$ in Theorem 2 and*

$$\mathbb{E}[\|\nabla f(x_a)\|^2] \le \frac{Ln^{2/3}[f(x^0) - f(x^*)]}{T\nu_1},$$

where x^* is an optimal solution to the problem in (1) and x_a is the output of Alg. 1.

Corollary 4. *If $f \in \mathcal{F}_n$, then the IFO complexity of Alg. 1 (with parameters in Corr. 3) to obtain an ϵ-accurate solution is $O(n + (n^{2/3}/\epsilon))$.*

Note the rate of $O(1/T)$ in the above results, as opposed to slower $O(1/\sqrt{T})$ rate of SGD (Thm. 1). For a more comprehensive comparison of the rates, refer to Sec. 6.

Algorithm 2 GD-SVRG$\big(x^0, K, T, m, \{p_i\}_{i=0}^m, \{\eta_i\}_{i=0}^{m-1}\big)$

```
Input: $x^0 \in \mathbb{R}^d$, $K$, epoch length $m$, step sizes $\{\eta_i > 0\}_{i=0}^{m-1}$,
discrete probability distribution $\{p_i\}_{i=0}^m$
for $k = 0$ to $K$ do
    $x^k = \text{SVRG}(x^{k-1}, T, m, \{p_i\}_{i=0}^m, \{\eta_i\}_{i=0}^{m-1})$
end for
Output: $x^K$
```

3.1. Gradient Dominated Functions

Before ending our discussion on convergence of nonconvex SVRG, we prove a linear convergence rate for the class of τ-gradient dominated functions (2). In fact, for τ-gradient dominated functions we can prove a stronger result of *global* linear convergence. For ease of exposition, assume that $\tau > n^{1/3}$, a property analogous to the "high condition number regime" for strongly convex functions typical in machine learning. Note that gradient dominated functions can be nonconvex.

Theorem 4. *Suppose $f \in \mathcal{F}_n$ is τ-gradient dominated ($\tau > n^{1/3}$). Then, the iterates of Alg. 2 with $T = \lceil 2L\tau n^{2/3}/\nu_1 \rceil$, $m = \lfloor n/(3\mu_1) \rfloor$, $\eta_t = \mu_1/(Ln^{2/3})$ for $0 \le t < m$, $p_m = 1$ and $p_i = 0$ for $0 \le i < m$ satisfy*

$$\mathbb{E}[\|\nabla f(x^k)\|^2] \le 2^{-k}[\|\nabla f(x^0)\|^2],$$
$$\mathbb{E}[f(x^k) - f(x^*)] \le 2^{-k}[f(x^0) - f(x^*)].$$

Here μ_1 and ν_1 are the constants used in Corr. 3.

An immediate consequence is the following.

Corollary 5. *If $f \in \mathcal{F}_n$ is τ-gradient dominated, the IFO complexity of Alg. 2 (with parameters from Thm. 4) to compute an ϵ-accurate solution is $O((n + \tau n^{2/3})\log(1/\epsilon))$.*

Note that GRADDESCENT can also achieve linear convergence rate for gradient dominated functions (Polyak, 1963). However, GRADDESCENT requires $O(n + n\tau \log(1/\epsilon))$ IFO calls to obtain an ϵ-accurate solution as opposed to $O(n + n^{2/3}\tau\log(1/\epsilon))$ for SVRG. Similar (but not the same) gains can be seen for SVRG for strongly convex functions (Johnson & Zhang, 2013). Also notice that we did not assume anything except smoothness on the *individual* functions f_i in the above results. In particular, the following corollary is also an immediate consequence.

Corollary 6. *If $f \in \mathcal{F}_n$ is λ-strongly convex and the functions $\{f_i\}_{i=1}^n$ are possibly nonconvex, then the IFO complexity of Alg. 2 (with parameters from Thm. 4) to compute an ϵ-accurate solution is $O((n + n^{2/3}\kappa)\log(1/\epsilon))$.*

Recall that here κ denotes the condition number L/λ for a λ-strongly convex function. Corr. 6 follows from Corr. 5 upon noting that λ-strongly convex function is $1/2\lambda$-gradient dominated. Thm. 4 generalizes the linear convergence result in (Johnson & Zhang, 2013) since it allows nonconvex f_i. Observe that Corr. 6 also applies when f_i is strongly convex ($i \in [n]$), though in this case a more refined result can be proved (Johnson & Zhang, 2013).

Finally, we note that our result also improves on a recent result on SDCA in the setting of Corr. 6 when the condition number κ is reasonably large. More precisely, for l_2-regularized empirical loss minimization, Shalev-Shwartz (2015) show that SDCA requires $O((n + \kappa^2)\log(1/\epsilon))$ iterations when the f_i's are possibly nonconvex but their sum f is strongly convex. In comparison, we show that Alg. 2 requires $O((n + n^{2/3}\kappa)\log(1/\epsilon))$ iterations, which is an improvement over SDCA when $\kappa > n^{2/3}$.

4. Convex Case

In the previous section, we showed nonconvex SVRG converges to a stationary point at the rate $O(n^{2/3}/T)$. A natural question is whether this rate can be improved if we assume convexity? We provide an affirmative answer. For non-strongly convex functions, this yields a *direct* analysis (i.e., not based on strongly convex perturbations) for SVRG. While we state our results in terms of stationarity gap $\|\nabla f(x)\|^2$ for the ease of comparison, our analysis also provides rates with respect to the optimality gap $[f(x) - f(x^*)]$ (see the proof of Thm. 5 in the appendix).

Theorem 5. *If $f \in \mathcal{F}_n$ and f_i is convex ($i \in [n]$), $p_i = 1/m$ for $0 \le i < m$, and $p_m = 0$. Then for Alg. 1, we have*

$$\mathbb{E}[\|\nabla f(x_a)\|^2] \le \frac{L\|x^0 - x^*\|^2 + 4mL^2\eta^2[f(x^0) - f(x^*)]}{T\eta(1 - 4L\eta)},$$

where x^* is optimal for (1) and x_a is the output of Alg. 1.

We now state corollaries of this theorem that explicitly show the dependence on n in the convergence rates.

Corollary 7. *If $m = n$ and $\eta = 1/(8L\sqrt{n})$ in Thm. 5, then we have the following bound:*

$$\mathbb{E}[\|\nabla f(x_a)\|^2] \le \frac{L\sqrt{n}(16L\|x^0 - x^*\|^2 + [f(x^0) - f(x^*)])}{T},$$

where x^* is optimal for (1) and x_a is the output of Alg. 1.

The above result uses a step size that depends on n. For the convex case, we can also use step sizes independent of n. The following corollary states the associated result.

Corollary 8. *If $m = n$ and $\eta = 1/(8L)$ in Thm. 5, then we have the following bound:*

$$\mathbb{E}[\|\nabla f(x_a)\|^2] \le \frac{L(16L\|x^0 - x^*\|^2 + n[f(x^0) - f(x^*)])}{T},$$

where x^* is optimal for (1) and x_a is the output of Alg. 1.

We can rewrite these corollaries in terms of IFO complexity to get the following corollaries.

Corollary 9. *If $f \in \mathcal{F}_n$ and f_i is convex for all $i \in [n]$, then the IFO complexity of Alg. 1 (with parameters from Corr. 7) to compute an ϵ-accurate solution is $O(n + \sqrt{n}/\epsilon)$.*

Corollary 10. *If $f \in \mathcal{F}_n$ and f_i is convex for all $i \in [n]$, then the IFO complexity of Alg. 1 (with parameters from Corr. 8) to compute ϵ-accurate solution is $O(n/\epsilon)$.*

These results follow from Corr. 7 and Corr. 8 and noting that for $m = O(n)$, the total IFO calls made by Alg. 1 is $O(n)$. It is instructive to quantitatively compare Corr. 9 and Corr. 10. With a step size independent of n, the convergence rate of SVRG has a dependence that is in the order of n (Corr. 8). But this dependence can be reduced to $\sqrt{n}$ by either carefully selecting a step size that diminishes with n (Corr. 7) or by using a good initial point x^0 obtained by, say, running $O(n)$ iterations of SGD.

We emphasize that the convergence rate for convex case can be improved significantly by slightly modifying the algorithm (either by adding an appropriate strongly convex perturbation (Xiao & Zhang, 2014) or by using a choice of m that changes with epoch (Zhu & Yuan, 2015)). However, it is not clear if these strategies provide any theoretical gains for the general nonconvex case.

5. Mini-batch Nonconvex SVRG

In this section, we study the mini-batch version of Alg. 1. Mini-batching is a popular strategy, especially in multicore and distributed settings as it greatly helps one exploit parallelism and reduce the communication costs. The pseudocode for mini-batch nonconvex SVRG (Alg. 3) is provided in the supplement due to lack of space. The key difference between the mini-batch SVRG and Alg. 1 lies in lines 6 to 8. To use mini-batches we replace line 6 with sampling (with replacement) a mini-batch $I_t \subset [n]$ of size b; lines 7 to 8 are replaced with the following updates:

$$u_t^{s+1} = \tfrac{1}{|I_t|} \sum\nolimits_{i_t \in I_t} \left(\nabla f_{i_t}(x_t^{s+1}) - \nabla f_{i_t}(\tilde{x}^s)\right) + g^{s+1},$$
$$x_{t+1}^{s+1} = x_t^{s+1} - \eta_t u_t^{s+1}$$

When $b = 1$, this reduces to Alg. 1. Mini-batch is typically used to reduce the variance of the stochastic gradient and increase the parallelism. Lem. 4 (in Sec. G of the appendix) shows the reduction in the variance of stochastic gradients with mini-batch size b. Using this lemma, one can derive the mini-batch equivalents of Lem. 1, Thm. 2 and Thm. 3. However, for the sake of brevity, we directly state the following main result for mini-batch SVRG.

Theorem 6. *Let $f \in \mathcal{F}_n$ and $\overline{\gamma}_n$ denote the following:*

$$\overline{\gamma}_n := \min_{0 \le t \le m-1} \quad \left(\eta - \tfrac{\overline{c}_{t+1}\eta}{\beta} - \eta^2 L - 2\overline{c}_{t+1}\eta^2\right),$$

where $\overline{c}_m = 0$, $\overline{c}_t = \overline{c}_{t+1}(1 + \eta\beta + {}^{2\eta^2L^2}\!/_b) + {}^{\eta_t^2 L^3}\!/_b$ for $0 \le t < m$. Suppose $\eta = \mu_2 b/(Ln^{2/3})$ $(0 < \mu_2 < 1)$, $\beta = L/n^{1/3}$, $m = \lfloor n/(3b\mu_2) \rfloor$ and T is some multiple of m. Then for $b < n^{2/3}$, there exists universal constants $\mu_2, \nu_2 > 0$ such that: $\overline{\gamma}_n \ge \frac{\nu_2 b}{Ln^{2/3}}$ and

$$\mathbb{E}[\|\nabla f(x_a)\|^2] \le \frac{Ln^{2/3}[f(x^0) - f(x^*)]}{bT\nu_2},$$

where x^* is optimal for (1) and x_a is the output of the mini-batch version of Alg. 1.

It is important to compare this result with mini-batched SGD. For a mini-batch size of b, SGD obtains a rate of $O(1/\sqrt{bT} + 1/T)$ (Dekel et al., 2012) (obtainable by a modification of Thm. 1). Specifically, SGD has a $1/\sqrt{b}$ dependence on the batch size. In contrast, Thm. 6 shows that SVRG has a much better dependence of $1/b$ on the batch size. Hence, compared to SGD, SVRG allows more efficient mini-batching. More formally, in terms of IFO queries we have the following result.

Corollary 11. *If $f \in \mathcal{F}_n$, then the IFO complexity of the mini-batch version of Alg. 1 (with parameters from Thm. 6 and mini-batch size $b < n^{2/3}$) to obtain an ϵ-accurate solution is $O(n + (n^{2/3}/\epsilon))$.*

Corr. 11 shows an interesting property of mini-batch SVRG. First, note that b IFO calls are required for calculating the gradient on a mini-batch of size b. Hence, SVRG does not gain on IFO complexity by using mini-batches. However, if the b gradients are calculated in parallel, then this leads to a theoretical linear speedup in multicore and distributed settings. In contrast, SGD does not yield an efficient mini-batch strategy (Li et al., 2014).

6. Comparison of the convergence rates

In this section, we give a comprehensive comparison of results obtained in this paper. In particular, we compare key aspects of the convergence rates for SGD, GRADDESCENT, and SVRG. The comparison is based on IFO complexity to achieve an ϵ-accurate solution.

Dependence on n: The number of IFO calls of SVRG and GRADDESCENT depend explicitly on n. In contrast, the number of oracle calls of SGD is independent of n (Thm. 1). However, this comes at the expense of worse dependence on ϵ. The number of IFO calls in GRADDESCENT is proportional to n. But for SVRG this dependence reduces to $n^{1/2}$ for convex (Corr. 7) and $n^{2/3}$ for nonconvex (Corr. 3) problems. Whether this difference in dependence on n is due to nonconvexity or just an artifact of our analysis is an interesting open problem.

Dependence on ϵ: The dependence on ϵ (or alternatively T) follows from the convergence rates of the algorithms. SGD is seen to depend as $O(1/\epsilon^2)$ on ϵ, regardless of convexity or nonconvexity. In contrast, for both convex and nonconvex settings, SVRG and GRADDESCENT converge as $O(1/\epsilon)$. Furthermore, for gradient dominated functions, SVRG and GRADDESCENT have global linear convergence. This speedup in convergence over SGD is especially significant when medium to high accuracy solutions are required (i.e., ϵ is small).

Assumptions used in analysis: It is important to understand the assumptions used in deriving the convergence rates. All algorithms assume Lipschitz continuous gradients. However, SGD requires two additional subtle but important assumptions: σ-bounded gradients and advance knowledge of T (since its step sizes depend on T). On the other hand, both SVRG and GRADDESCENT do not require these assumptions, and thus, are more flexible.

Step size / learning rates: It is valuable to compare the step sizes used by the algorithms. The step sizes of SGD shrink as the number of *iterations* T increases—an undesirable property. On the other hand, the step sizes of SVRG and GRADDESCENT are independent of T. Hence, both these algorithms can be executed with a fixed step size. However, SVRG uses step sizes that depend on n (see Corr. 3 and Corr. 7). A step size independent of n can be used for SVRG for convex f, albeit at cost of worse dependence on n (Corr. 8). GRADDESCENT does not have this issue as its step size is independent of both n and T.

Dependence on initial point and mini-batch: SVRG is more sensitive to the initial point in comparison to SGD. This can be seen by comparing Corr. 3 (of SVRG) to Thm. 1 (of SGD). Hence, it is important to use a good initial point for SVRG. Similarly, a reasonably large mini-batch can be beneficial to SVRG. For SVRG, mini-batches not only provides parallelism but also good theoretical guarantees (see Thm. 6). In contrast, the performance gain in SGD with mini-batches is not very pronounced (see Sec. 5).

7. Best of two worlds

We have seen in the previous section that SVRG combines the benefits of both GRADDESCENT and SGD. We now show that these benefits of SVRG can be made more pronounced by an appropriate step size under additional assumptions. In this case, the IFO complexity of SVRG is lower than those of SGD and GRADDESCENT. This variant of SVRG (MSVRG) chooses a step size based on the total number of iterations T (or alternatively ϵ). For our discussion below, we assume that $T > n$.

Theorem 7. *Suppose $f \in \mathcal{F}_n$ has σ-bounded gradients. Let $\eta_t = \eta = \max\{c/\sqrt{T}, \mu_1/(Ln^{2/3})\}$ (μ_1 is the constant from Corr. 3), $m = \lfloor n/(3\mu_1) \rfloor$, and $c = \sqrt{\frac{f(x^0)-f(x^*)}{2L\sigma^2}}$. Further, let T be a multiple of m, $p_m = 1$, and $p_i = 0$ for $0 \leq i < m$. Then, the output x_a of Alg. 1 satisfies*

$$\mathbb{E}[\|\nabla f(x_a)\|^2]$$
$$\leq \bar{\nu} \min\Big\{2\sqrt{\frac{2(f(x^0) - f(x^*))L}{T}}\sigma, \frac{Ln^{2/3}[f(x^0) - f(x^*)]}{T\nu_1}\Big\},$$

where $\bar{\nu} > 0$ is a universal constant, ν_1 is the universal constant from Corr. 3 and x^* is an optimal solution to (1).

Corollary 12. *If $f \in \mathcal{F}_n$ has σ-bounded gradients, the IFO complexity of Alg. 1 (with parameters from Thm. 7) to achieve an ϵ-accurate solution is $O(\min\{1/\epsilon^2, n^{2/3}/\epsilon\})$.*

An almost identical reasoning can be applied when f is convex to get the bounds specified in Table 1. Hence, we omit the details and directly state the following result.

Corollary 13. *Suppose $f \in \mathcal{F}_n$ has σ-bounded gradients and f_i is convex for $i \in [n]$, then the IFO complexity of Alg. 1 (with step size $\eta = \max\{1/(L\sqrt{T}), 1/(8L\sqrt{n})\}$, $m = n$ and $p_i = 1/m$ for $0 \leq i \leq m-1$ and $p_m = 0$) to achieve an ϵ-accurate solution is $O(\min\{1/\epsilon^2, \sqrt{n}/\epsilon\})$.*

MSVRG has a convergence rate faster than those of both SGD and SVRG, though this benefit is not without cost. MSVRG, in contrast to SVRG, uses the additional assumption of σ-bounded gradients. Furthermore, its step size is *not* fixed since it depends on the number of iterations T. While it is often difficult to compute the step size of MSVRG (Thm. 7) in practice, it is typical to try multiple step sizes and choose the one with the best results.

8. Experiments

We present our empirical results in this section. In particular, we study multiclass classification using neural networks. This is typical nonconvex problem encountered in machine learning.

Experimental Setup. We train neural networks with one fully-connected hidden layer of 100 nodes and 10 softmax output nodes. We use ℓ_2-regularization for training. We use CIFAR-10[2], MNIST[3], and STL-10[4] datasets for our experiments. These datasets are standard in the neural networks literature. The ℓ_2 regularization is 1e-3 for CIFAR-10 and MNIST, and 1e-2 for STL-10. The features in the datasets are normalized to the interval $[0, 1]$. All the datasets come with a predefined split into training and test datasets.

We compare SGD (the *de facto* algorithm for training neural networks) against nonconvex SVRG. The step size is critical for SGD; we set it using the popular t-inverse schedule $\eta_t = \eta_0(1+\eta'\lfloor t/n \rfloor)^{-1}$, where η_0 and η' are chosen so that SGD gives the best performance on the training loss. In our experiments, we also use $\eta' = 0$; this results in a fixed step size for SGD. For SVRG, we use a fixed step size as suggested by our analysis. Again, the step size is chosen so that SVRG gives the best performance on the training loss.

Initialization & mini-batching. Initialization is critical to training of neural networks. We use the normalized initialization in (Glorot & Bengio, 2010) where parameters are

[2] www.cs.toronto.edu/ kriz/cifar.html

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

[4] https://cs.stanford.edu/ acoates/stl10/

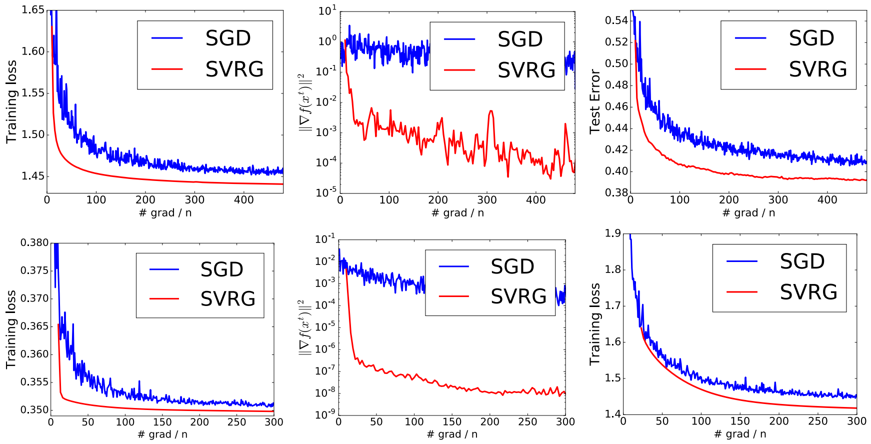

Figure 1. Neural network results for CIFAR-10, MNIST and STL-10 datasets. The top row represents the results for CIFAR-10 dataset. The bottom left and middle figures represent the results for MNIST dataset. The bottom right figure represents the result for STL-10.

chosen uniformly from $[-\sqrt{6/(n_i+n_o)}, \sqrt{6/(n_i+n_o)}]$ where n_i and n_o are the number of input and output layers of the neural network, respectively.

For SVRG, we use n iterations of SGD for CIFAR-10 and MINST and $2n$ iterations of SGD for STL-10 before running Alg. 1. Such initialization is standard for variance reduced schemes even for convex problems (Johnson & Zhang, 2013; Schmidt et al., 2013). As noted earlier in Sec. 6, SVRG is more sensitive than SGD to the initial point, so such an initialization is typically helpful. We use mini-batches of size 10 in our experiments. SGD with mini-batches is common in training neural networks. Note that mini-batch training is especially beneficial for SVRG, as shown by our analysis in Sec. 5. Along the lines of theoretical analysis provided by Thm. 6, we use an epoch size $m = n/10$ in our experiments.

Results. We report objective function (training loss), test error (classification error on the test set), and $\|\nabla f(x^t)\|^2$ (convergence criterion in our analysis). For all algorithms, we compare these criteria against the number of *effective passes* through the data, i.e., IFO calls divided by n. This includes the cost of calculating the full gradient at the end of each epoch of SVRG. Due to the SGD initialization in SVRG and mini-batching, the SVRG plots start from an x-axis value of 10 for CIFAR-10 and MNIST and 20 for STL-10. Figure 1 shows the results. It can be seen that for SVRG $\|\nabla f(x^t)\|^2$ is lower compared to SGD, suggesting faster convergence. Furthermore, training loss is also lower compared to SGD in all the datasets. Notably, the test error for CIFAR-10 is lower for SVRG, indicating better generalization; we did not notice substantial difference in test error for MNIST and STL-10 (see Sec. H in the appendix). Overall, these results on a network with one hidden layer are promising; it will be interesting to study SVRG for deep neural networks in the future.

9. Discussion

In this paper, we examined a VR scheme for nonconvex optimization. We showed that by employing VR in stochastic methods, one can outperform both SGD and GRADDESCENT even for nonconvex optimization. When the function f in (1) is gradient dominated, we proposed a variant of SVRG that has linear convergence to the *global* minimum. Our analysis shows that SVRG has a number of interesting properties that include convergence with fixed step size, descent (in expectation) after every epoch; a property that need not hold for SGD. We also showed that SVRG, in contrast to SGD, enjoys efficient mini-batching, attaining speedups linear in the size of the mini-batches in parallel settings. Our analysis also reveals that the initial point and use of mini-batches are important to SVRG.

Before concluding the paper, we would like to discuss the implications of our work and few caveats. One should exercise some caution while interpreting the results in the paper. All our theoretical results are based on the stationarity gap. In general, this does not necessarily translate to optimality gap or low training loss and test error. One criticism against VR schemes in nonconvex optimization is the general wisdom that variance in the stochastic gradients of SGD can actually help it escape local minimum and saddle points. In fact, Ge et al. (2015) add additional noise to the stochastic gradient in order to escape saddle points. However, one can reap the benefit of VR schemes even in such scenarios. For example, one can envision an algorithm which uses SGD as an exploration tool to obtain a good initial point and then uses a VR algorithm as an exploitation tool to quickly converge to a *good* local minimum. In either case, we believe variance reduction can be used as an important tool alongside other tools like momentum, adaptive learning rates for faster and better nonconvex optimization.

Acknowledgment: SS and AS were partially supported by NSF grant: IIS-1409802.

References

Agarwal, Alekh and Bottou, Leon. A lower bound for the optimization of finite sums. *arXiv:1410.0723*, 2014.

Allen-Zhu, Zeyuan and Hazan, Elad. Variance reduction for faster non-convex optimization. In *ICML*, 2016.

Bertsekas, Dimitri P. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. In S. Sra, S. Nowozin, S. Wright (ed.), *Optimization for Machine Learning*. MIT Press, 2011.

Bottou, Léon. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nîmes*, 91(8), 1991.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS 27*, pp. 1646–1654, 2014a.

Defazio, Aaron J, Caetano, Tibério S, and Domke, Justin. Finito: A faster, permutable incremental gradient method for big data problems. *arXiv:1407.2710*, 2014b.

Dekel, Ofer, Gilad-Bachrach, Ran, Shamir, Ohad, and Xiao, Lin. Optimal distributed online prediction using mini-batches. *The Journal of Machine Learning Research*, 13(1):165–202, January 2012. ISSN 1532-4435.

Ge, Rong, Huang, Furong, Jin, Chi, and Yuan, Yang. Escaping from saddle points - online stochastic gradient for tensor decomposition. In *COLT 2015*, pp. 797–842, 2015.

Ghadimi, Saeed and Lan, Guanghui. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. doi: 10.1137/120880811.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, 2010.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS 26*, pp. 315–323, 2013.

Konečný, Jakub and Richtárik, Peter. Semi-Stochastic Gradient Descent Methods. *arXiv:1312.1666*, 2013.

Konečný, Jakub, Liu, Jie, Richtárik, Peter, and Takáč, Martin. Mini-Batch Semi-Stochastic Gradient Descent in the Proximal Setting. *arXiv:1504.04407*, 2015.

Kushner, Harold Joseph and Clark, Dean S. *Stochastic approximation methods for constrained and unconstrained systems*, volume 26. Springer Science & Business Media, 2012.

Lan, Guanghui and Zhou, Yi. An optimal randomized incremental gradient method. *arXiv:1507.02000*, 2015.

Li, Mu, Zhang, Tong, Chen, Yuqiang, and Smola, Alexander J. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 661–670. ACM, 2014.

Lian, Xiangru, Huang, Yijun, Li, Yuncheng, and Liu, Ji. Asynchronous Parallel Stochastic Gradient for Nonconvex Optimization. In *NIPS*, 2015.

Ljung, Lennart. Analysis of recursive stochastic algorithms. *Automatic Control, IEEE Transactions on*, 22 (4):551–575, 1977.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4): 1574–1609, 2009.

Nemirovski, Arkadi and Yudin, D. *Problem Complexity and Method Efficiency in Optimization*. John Wiley and Sons, 1983.

Nesterov, Yurii. *Introductory Lectures On Convex Optimization: A Basic Course*. Springer, 2003.

Nesterov, Yurii and Polyak, Boris T. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Poljak, BT and Tsypkin, Ya Z. Pseudogradient adaptation and training algorithms. *Automation and Remote Control*, 34:45–67, 1973.

Polyak, B.T. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, January 1963.

Reddi, Sashank, Hefny, Ahmed, Sra, Suvrit, Poczos, Barnabas, and Smola, Alex J. On variance reduction in stochastic gradient descent and its asynchronous variants. In *NIPS 28*, pp. 2629–2637, 2015.

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

Reddi, Sashank J., Sra, Suvrit, Póczos, Barnabás, and Smola, Alexander J. Fast stochastic methods for nonsmooth nonconvex optimization. *arXiv:1605.06900*, 2016b.

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

Schmidt, Mark W., Roux, Nicolas Le, and Bach, Francis R. Minimizing Finite Sums with the Stochastic Average Gradient. *arXiv:1309.2388*, 2013.

Shalev-Shwartz, Shai. SDCA without duality. *arXiv:1502.06177*, 2015.

Shalev-Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

Shamir, Ohad. A stochastic PCA and SVD algorithm with an exponential convergence rate. *arXiv:1409.2848*, 2014.

Shamir, Ohad. Fast stochastic algorithms for SVD and PCA: Convergence properties and convexity. *arXiv:1507.08788*, 2015.

Sra, Suvrit. Scalable nonconvex inexact proximal splitting. In *NIPS*, pp. 530–538, 2012.

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

Zhu, Zeyuan Allen and Yuan, Yang. Univr: A universal variance reduction framework for proximal stochastic gradient method. *arXiv:1506.01972*, 2015.