Guaranteed Sufficient Decrease for Stochastic Variance Reduced Gradient Optimization

Fanhua Shang,   Yuanyuan Liu*,   **Kaiwen Zhou,   James Cheng,   Kelvin K.W. Ng**
Department of Computer Science and Engineering, The Chinese University of Hong Kong

Yuichi Yoshida
National Institute of Informatics, Tokyo, Japan

Abstract

In this paper, we propose a novel sufficient decrease technique for stochastic variance reduced gradient descent methods such as SVRG and SAGA. In order to make sufficient decrease for stochastic optimization, we design a new sufficient decrease criterion, which yields sufficient decrease versions of stochastic variance reduction algorithms such as SVRG-SD and SAGA-SD as a byproduct. We introduce a coefficient to scale current iterate and to satisfy the sufficient decrease property, which takes the decisions to shrink, expand or even move in the opposite direction, and then give two specific update rules of the coefficient for Lasso and ridge regression. Moreover, we analyze the convergence properties of our algorithms for strongly convex problems, which show that our algorithms attain linear convergence rates. We also provide the convergence guarantees of our algorithms for non-strongly convex problems. Our experimental results further verify that our algorithms achieve significantly better performance than their counterparts.

1 Introduction

Stochastic gradient descent (SGD) has been successfully applied to many large-scale machine learning problems [1, 2], due to its low per-iteration cost, $O(d)$. However, standard SGD estimates the gradient from only one or a few samples, and thus the variance of the stochastic gradient estimator may be large [3, 4], which leads to slow convergence and poor performance. In particular, even under the strongly convex (SC) condition, the convergence rate of standard SGD is only sub-linear [5, 6]. Recently, the convergence rate of the standard SGD has been improved by various stochastic variance reduction gradient methods, such as SAG [7], SDCA [8], SVRG [3], SAGA [9], Finito [10], MISO [11], and their proximal variants, such as [12], [13] and [14]. Under the SC condition, these stochastic variance reduction algorithms achieve linear convergence rates.

Very recently, some acceleration techniques were proposed to further speed up the stochastic variance reduction methods mentioned above. These techniques include importance sampling [4], exploiting neighborhood structure in the training data to share and re-use information about past stochastic gradients [15], incorporating Nesterov's acceleration techniques [16, 17] or momentum acceleration tricks [18], reducing the number of gradient computations in the early iterations [19, 20, 21, 22], and the projection-free property of the conditional gradient method [23]. In particular, Katyusha [18] can attain the upper complexity bounds for both SC and non-strongly convex (non-SC) composite problems, respectively, as discussed in [24].

Challenges and Main Contributions. So far the two most popular stochastic gradient estimators are the SVRG estimator independently introduced by [3, 21] and the SAGA estimator [9]. All the estimators may be very different from their full gradient counterparts, thus moving in the direction may not decrease the objective function anymore, as stated in [18]. To address this problem, inspired by the success of sufficient decrease methods for deterministic optimization such as [25, 26], we propose a novel sufficient decrease technique for a class of stochastic variance reduction methods, including the widely-used SVRG and SAGA methods. Notably, our method with partial sufficient decrease achieves average time complexity per-iteration

as low as the original SVRG and SAGA methods. We summarize our main contributions below.

- For making sufficient decrease for stochastic optimization, we design a sufficient decrease strategy to further reduce the cost function, in which we also introduce a coefficient to take the decision to shrink, expand or move in the opposite direction.

- We incorporate our sufficient decrease technique, together with momentum acceleration, into two representative SVRG and SAGA algorithms, which lead to SVRG-SD and SAGA-SD. Moreover, we give two specific update rules of the coefficient for Lasso and ridge regression problems as notable examples.

- Moreover, we analyze the convergence property of SVRG-SD, which shows that SVRG-SD converges linearly for SC composite minimization problems. Unlike most of the stochastic variance reduction methods such as SVRG, we also provide the convergence guarantee of SVRG-SD for non-SC composite minimization problems.

- Finally, we show by experiments that SVRG-SD and SAGA-SD achieve significantly better performance than SVRG [3] and SAGA [9]. Compared with the well-known accelerated method, Katyusha [18], our algorithms also have much better performance in most cases.

2 Preliminary and Related Work

2.1 Notations

Throughout this paper, we use $\|\cdot\|$ to denote the ℓ_2-norm (also known as the standard Euclidean norm), and $\|\cdot\|_1$ is the ℓ_1-norm, i.e., $\|x\|_1 = \sum_{i=1}^{d} |x_i|$. $\nabla f(\cdot)$ denotes the full gradient of the function $f(\cdot)$ if it is differentiable, or $\partial f(\cdot)$ the subgradient if $f(\cdot)$ is only Lipschitz continuous.

In this paper, we consider the following unconstrained composite convex optimization problem:

$$\min_{x \in \mathbb{R}^d} F(x) \overset{\text{def}}{=} f(x) + r(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) + r(x), \quad (1)$$

where $f_i(x) : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \ldots, n$ are the smooth convex component functions, and $r(x)$ is a relatively simple convex (but possibly non-differentiable) function.

We consider the problem (1) under the following standard assumptions.

Assumption 1. *Each convex function $f_i(\cdot)$ is L-smooth, iff there exists a constant $L > 0$ such that for any $x, y \in \mathbb{R}^d$, $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$.*

Algorithm 1 SVRG

Input: The number of epochs S, the number of iterations m per epoch, and the step size η.
Initialize: $\widetilde{x}^0$.
1: **for** $s = 1, 2, \ldots, S$ **do**
2: $\quad \widetilde{\mu}^{s-1} = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\widetilde{x}^{s-1})$;
3: $\quad x_0^s = \widetilde{x}^{s-1}$;
4: $\quad$ **for** $k = 1, 2, \ldots, m$ **do**
5: $\quad\quad$ Pick i_k^s uniformly at random from $[n]$;
6: $\quad\quad \widetilde{\nabla} f_{i_k^s}(x_{k-1}^s) = \nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\widetilde{x}^{s-1}) + \widetilde{\mu}^{s-1}$;
7: $\quad\quad x_k^s = x_{k-1}^s - \eta \widetilde{\nabla} f_{i_k^s}(x_{k-1})$;
8: $\quad$ **end for**
9: $\quad \widetilde{x}^s = x_m^s$;
10: **end for**
Output: $\widetilde{x}^S$

Assumption 2. *$F(\cdot)$ is μ-strongly convex, iff there exists a constant $\mu > 0$ such that for any $x, y \in \mathbb{R}^d$,*

$$F(y) \geq F(x) + \vartheta^T(y - x) + \frac{\mu}{2}\|y - x\|^2, \quad \forall \vartheta \in \partial F(x), \quad (2)$$

where $\partial F(x)$ is the subdifferential of $F(\cdot)$ at x. If $F(\cdot)$ is smooth, we can revise the inequality (2) by simply replacing the sub-gradient $\vartheta \in \partial F(x)$ with $\nabla F(x)$.

2.2 SVRG and SAGA

Recently, many stochastic variance reduced methods [3, 7, 14, 21] have been proposed for some special cases of Problem (1). Under smoothness and SC assumptions, and $r(x) \equiv 0$, SAG [7] achieves a linear convergence rate. A recent line of work, such as [3, 14], has been proposed with similar convergence rates to SAG but without the memory requirements for all gradients. SVRG [3] begins with an initial estimate $\widetilde{x}^0$, sets $x_0^1 = \widetilde{x}^0$ and then generates a sequence of x_k^s at the s-th epoch ($k = 1, 2, \ldots, m$, where m is usually set to $2n$, as suggested in [3, 14]) using

$$x_k^s = x_{k-1}^s - \eta \big[\nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\widetilde{x}^{s-1}) + \widetilde{\mu}^{s-1} \big], \quad (3)$$

where $\eta > 0$ is the step size, $\widetilde{\mu}^{s-1} = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\widetilde{x}^{s-1})$ is the full gradient of $f(\cdot)$ at the snapshot $\widetilde{x}^{s-1}$, and i_k^s is chosen uniformly at random from $\{1, 2, \ldots, n\}$. After every m stochastic iterations, we set $\widetilde{x}^s = x_m^s$, and reset $k = 1$ and $x_0^{s+1} = \widetilde{x}^s$ (see Algorithm 1 for details).

In particular, the stochastic gradient estimator in Line 6 of Algorithm 1, i.e., the SVRG estimator independently introduced in [3, 21], is a popular choice for stochastic gradient estimators. Unfortunately, most of the stochastic variance reduction methods [8, 10, 14], including SVRG, only have convergence guarantees for smooth and SC problems. However, $F(\cdot)$ may be non-SC in many machine learning applications (e.g., Lasso)

Fanhua Shang, Yuanyuan Liu*, Kaiwen Zhou, James Cheng, Kelvin K.W. Ng

or non-smooth and SC problems, e.g., elastic-net. For solving the non-smooth problem (1), a proximal variant of SVRG, i.e., Prox-SVRG [14], has been proposed and relies on the use of the proximal operator, $\text{prox}_\eta^r(\cdot)$, which is defined as:

$$\text{prox}_\eta^r(y) = \arg\min_x \left\{ (1/2\eta) \cdot \|x-y\|^2 + r(x) \right\}.$$

[9] proposed SAGA, a fast incremental gradient method in the spirit of SAG and SVRG, which works for both SC and non-SC objective functions, as well as in the proximal setting. Unlike SVRG and Prox-SVRG, both of which are two-stage algorithms, SAGA is a single-stage incremental gradient algorithm. Its main update rules are formulated as follows:

$$\widetilde{\nabla} f_{i_k}(x_{k-1}) = g_{i_k}^k - g_{i_k}^{k-1} + \frac{1}{n}\sum_{j=1}^n g_j^{k-1}, \qquad (3a)$$

$$x_k = \text{prox}_\eta^r\left(x_{k-1} - \eta\widetilde{\nabla} f_{i_k}(x_{k-1})\right), \qquad (3b)$$

where g_j^k is updated for all $j=1,\ldots,n$ as follows: $g_j^k = \nabla f_{i_k}(x^{k-1})$ if $i_k=j$, and $g_j^k=g_j^{k-1}$ otherwise (see [9, 27] for details). Here, the stochastic gradient estimator in (3a) is called as the SAGA estimator.

2.3 Sufficient Decrease for Deterministic Optimization

The technique of sufficient decrease (e.g., the well-known line search technique [28]) has been studied for deterministic optimization [25, 26]. For example, [25] proposed the following sufficient decrease condition:

$$F(x_k) \le F(x_{k-1}) - \delta\|y_k - x_{k-1}\|^2, \qquad (4)$$

where $\delta > 0$ is a small constant, and $y_k = \text{prox}_{\eta_k}^r(x_{k-1} - \eta_k\nabla f(x_{k-1}))$. Inspired by the strategy for deterministic optimization, in this paper we design a novel sufficient decrease technique for stochastic optimization, which is used to further reduce the cost function and speed up its convergence.

3 SVRG with Sufficient Decrease

In this section, we propose a novel sufficient decrease technique for stochastic variance reduction optimization methods, which include the widely-used SVRG method. To make sufficient decrease for stochastic optimization, we design an effective sufficient decrease strategy that can further reduce the cost function. Then a coefficient θ is introduced to satisfy the sufficient decrease condition, and takes the decisions to shrink, expand or move in the opposite direction. Moreover, we present a variant of SVRG with sufficient decrease and momentum acceleration (called SVRG-SD). We also give two specific schemes to compute θ for Lasso and ridge regression.

Algorithm 2 SVRG-SD

Input: the number of epochs S, the number of iterations m per epoch, and the step size η.
Initialize: $\widetilde{x}^0$ for Case of SC, or $\widetilde{x}^0 = \widetilde{y}^0$ for Case of non-SC.
 1: **for** $s = 1, 2, \ldots, S$ **do**
 2:     Case of SC: $x_0^s = \widehat{x}_0^s = \widetilde{x}^{s-1}$;
 3:     Case of non-SC: $x_0^s = \widehat{x}_0^s = \widetilde{y}^{s-1}$;
 4:     $\widetilde{\mu}^{s-1} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\widetilde{x}^{s-1})$;
 5:     **for** $k = 1, 2, \ldots, m$ **do**
 6:         Pick i_k^s uniformly at random from $[n]$;
 7:         $\widetilde{\nabla} f_{i_k^s}(x_{k-1}^s) = \nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\widetilde{x}^{s-1}) + \widetilde{\mu}^{s-1}$;
 8:         Update y_k^s, θ_k, and x_k^s by (8), (5), and (7);
 9:     **end for**
10:     $\widetilde{x}^s = \frac{1}{m}\sum_{k=1}^m \widehat{x}_k^s$;
11:     Case of non-SC: $\widetilde{y}^s = [x_m^s - (1-\sigma)\widehat{x}_m^s]/\sigma$;
12: **end for**
Output: $\overline{x} = \widetilde{x}^S$ (SC), or $\overline{x} = \widetilde{x}^S$ if $F(\frac{1}{S}\sum_{s=1}^S \widetilde{x}^s) \ge F(\widetilde{x}^S)$ and $\overline{x} = \frac{1}{S}\sum_{s=1}^S \widetilde{x}^s$ otherwise (non-SC)

3.1 Our Sufficient Decrease Technique

Suppose $x_k^s = \text{prox}_\eta^r(x_{k-1}^s - \eta[\nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\widetilde{x}^{s-1}) + \widetilde{\mu}^{s-1}])$ for the s-th outer-iteration and the k-th inner-iteration. Unlike the full gradient method, the stochastic gradient estimator is somewhat inaccurate (i.e., it may be very different from $\nabla f(x_{k-1}^s)$), then further moving in the updating direction may not decrease the objective value anymore [18]. That is, $F(x_k^s)$ may be larger than $F(x_{k-1}^s)$ even for very small step length $\eta > 0$. Motivated by this observation, we design a factor θ to scale the current iterate x_{k-1}^s for the decrease of the objective function. For SVRG-SD, the cost function with respect to θ is formulated as follows:

$$\min_{\theta\in\mathbb{R}} F(\theta x_{k-1}^s) + \frac{\zeta(1-\theta)^2}{2}\|\nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\widetilde{x}^{s-1})\|^2, \quad (5)$$

where $\zeta = \frac{\delta\eta}{1-L\eta}$ is a trade-off parameter between the two terms, δ is a small constant and set to 0.1. The second term in (5) involves the norm of the residual of stochastic gradients, and plays the same role as the second term of the right-hand side of (4). Different from existing sufficient decrease techniques including (4), a varying factor θ instead of a constant is introduced to scale x_{k-1}^s and the coefficient of the second term of (5), and θ plays a similar role as the step-size parameter optimized via a line-search for deterministic optimization. However, line search techniques have a high computational cost in general, which limits their applicability to stochastic optimization [29].

Note that θ is a scalar and takes the decisions to shrink, expand x_{k-1}^s or move in the opposite direction of x_{k-1}^s. The detailed schemes to calculate θ for Lasso and ridge regression are given in Section 3.3. We first present the

following sufficient decrease condition in the statistical sense for stochastic optimization.

Property 1. *For given x_{k-1}^s and the solution θ_k of Problem (5), then the following inequality holds*

$$F(\theta_k x_{k-1}^s) \le F(x_{k-1}^s) - \frac{\zeta(1-\theta_k)^2}{2}\|\widetilde{p}_{i_k^s}\|^2, \quad (6)$$

where $\widetilde{p}_{i_k^s} = \nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\widetilde{x}^{s-1})$ for SVRG-SD.

It is not hard to verify that $F(\cdot)$ can be further decreased via our sufficient decrease technique, when the current iterate x_{k-1}^s is scaled by the coefficient θ_k. Indeed, for the special case when $\theta_k = 1$ for some k, the inequality in (6) can be still satisfied. Unlike the sufficient decrease condition for deterministic optimization [25, 26], θ_k may be a negative number, which means to move in the opposite direction of x_{k-1}^s.

3.2 Momentum Acceleration

In this part, we first design the update rule for the key variable x_k^s with the coefficient θ_k as follows:

$$x_k^s = y_k^s + (1-\sigma)(\widehat{x}_k^s - \widehat{x}_{k-1}^s), \quad (7)$$

where $\widehat{x}_k^s = \theta_k x_{k-1}^s$, $\sigma \in [0,1]$ is a constant and can be set to $\sigma = 1/2$ which also works well in practice. In fact, the second term of the right-hand side of (7) plays a momentum acceleration role as in stochastic and deterministic optimization [17, 18, 30, 31, 32, 33, 34]. That is, by introducing this term, we can utilize the previous information of gradients to update x_k^s. Then the update rule of y_k^s is given by

$$y_k^s = \text{prox}_\eta^r \left(x_{k-1}^s - \eta \widetilde{\nabla} f_{i_k^s}(x_{k-1}^s) \right), \quad (8)$$

where $\eta = 1/(L\alpha)$, $L > 0$ is a Lipschitz constant (see Assumption 1), and $\alpha \ge 1$ denotes a constant. From the update rule in (8), it is clear that SVRG-SD can tackle the non-smooth minimization problem (1) directly as Prox-SVRG [14] and SAGA [9]. In addition, $\widetilde{\nabla} f_{i_k^s}(x_{k-1}^s)$ can be the two most popular choices for stochastic gradient estimators: the SVRG estimator [3, 21] and the SAGA estimator [9]. For SVRG-SD, $\widetilde{\nabla} f_{i_k^s}(x_{k-1}^s)$ is defined as follows:

$$\widetilde{\nabla} f_{i_k^s}(x_{k-1}^s) = \nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\widetilde{x}^{s-1}) + \widetilde{\mu}^{s-1}. \quad (9)$$

In summary, we propose a novel variant of SVRG with sufficient decrease (called SVRG-SD) to solve both SC and non-SC problems, as outlined in **Algorithm** 2. For the case of SC, $x_0^s = \widehat{x}_0^s = \widetilde{x}^{s-1}$, while $x_0^s = \widehat{x}_0^s = \widetilde{y}^{s-1}$ and $\widetilde{y}^s = [x_m^s - (1-\sigma)\widehat{x}_m^s]/\sigma$ are set for the case of non-SC. Note that when $\theta_k \equiv 1$ and $\sigma = 1$, the proposed SVRG-SD degenerates to the original SVRG or its proximal variant, Prox-SVRG [14]. In this sense, SVRG and Prox-SVRG can be seen as the special cases of the proposed SVRG-SD algorithm.

3.3 Computing Coefficients for Lasso and Ridge Regression

In this part, we give the closed-form solutions of the coefficient θ for Lasso and ridge regression.

For Lasso problems and given x_{k-1}^s, we have

$$F(\theta x_{k-1}^s) = \frac{1}{2n}\sum_{i=1}^n (\theta a_i^T x_{k-1}^s - b_i)^2 + \lambda\|\theta x_{k-1}^s\|_1.$$

The closed-form solution of Problem (5) for SVRG-SD can be obtained as follows:

$$\theta_k = \mathcal{S}_\tau \left(\frac{\frac{1}{n} b^T A x_{k-1}^s + \zeta\|\widetilde{p}_{i_k^s}\|^2}{\|A x_{k-1}^s\|^2/n + \zeta\|\widetilde{p}_{i_k^s}\|^2} \right), \quad (10)$$

where $A = [a_1, \ldots, a_n]^T$ is the data matrix containing n data samples, $b = [b_1, \ldots, b_n]^T$, and $\mathcal{S}_\tau$ is the so-called soft thresholding operator [35] with the following threshold,

$$\tau = \frac{\lambda\|x_{k-1}^s\|_1}{\|A x_{k-1}^s\|^2/n + \zeta\|\widetilde{p}_{i_k^s}\|^2}.$$

For ridge regression problems, we have

$$F(\theta x_{k-1}^s) = \frac{1}{2n}\sum_{i=1}^n (\theta a_i^T x_{k-1}^s - b_i)^2 + \frac{\lambda}{2}\|\theta x_{k-1}^s\|^2.$$

The closed-form solution of Problem (5) for SVRG-SD is given by

$$\theta_k = \frac{\frac{1}{n} b^T A x_{k-1}^s + \zeta\|\widetilde{p}_{i_k^s}\|^2}{\|A x_{k-1}^s\|^2/n + \zeta\|\widetilde{p}_{i_k^s}\|^2 + \lambda\|x_{k-1}^s\|^2}. \quad (11)$$

We can compute the coefficient for other loss functions (e.g., logistic regression) using line search (e.g., Armijo's line search [36]) or their approximations [37].

4 Convergence Analysis

In this section, we provide the convergence analysis of SVRG-SD for solving both SC and non-SC minimization problems.

4.1 Convergence Property for SC Problems

In this part, we analyze the convergence property of SVRG-SD for solving the SC problem (1). The first main result is the following theorem, which provides the convergence rate of SVRG-SD.

Theorem 1. *Suppose Assumption 1 holds. Let x^* be the optimal solution of Problem (1), and $\{(x_k^s, y_k^s, \theta_k)\}$ be the sequence produced by SVRG-SD, $\eta = 1/(L\alpha)$, and $\frac{2}{\alpha-1} < \sigma$, then*

$$\mathbb{E}[F(\widetilde{x}^s) - F(x^*)] \le \left(\frac{\frac{1-\sigma}{m} + \frac{2}{\alpha-1}}{\sigma - \frac{2}{\alpha-1} + \widehat{\beta}} \right) \mathbb{E}\left[F(\widetilde{x}^{s-1}) - F(x^*) \right]$$

$$+ \frac{L\alpha\sigma^2}{2m\left(\sigma - \frac{2}{\alpha-1} + \widehat{\beta}\right)} \mathbb{E}\left[\|x^* - z_0^s\|^2 - \|x^* - z_m^s\|^2 \right],$$

Fanhua Shang, Yuanyuan Liu*, Kaiwen Zhou, James Cheng, Kelvin K.W. Ng

where $z_0^s = [x_0^s - (1-\sigma)\widehat{x}_0^s]/\sigma$, $z_m^s = [x_m^s - (1-\sigma)\widehat{x}_{m-1}^s]/\sigma$, $\widehat{\beta} = \min_{s=1,\ldots,S} \widehat{\beta}^s \geq 0$, and $\widehat{\beta}^s = \mathbb{E}[\sum_{k=1}^m \frac{2c_k\beta_k}{\alpha-1}(F(\widehat{x}_k^s) - F(x^*))]/\mathbb{E}[\sum_{k=1}^m (F(\widehat{x}_k^s) - F(x^*))]$.

The proof of Theorem 1 and the definitions of c_k and β_k are given in the Supplementary Material. The linear convergence of SVRG-SD follows immediately.

Corollary 1 (SC). *Suppose each $f_i(\cdot)$ is L-smooth, and $F(\cdot)$ is μ-strongly convex. Setting $\alpha = 19$, $\sigma = 1/2$, and m sufficiently large so that*

$$\rho = \frac{9}{(7+18\widehat{\beta})m} + \frac{2}{7+18\widehat{\beta}} + \frac{171L}{(14+36\widehat{\beta})m\mu} < 1,$$

then SVRG-SD has the geometric convergence in expectation:

$$\mathbb{E}[F(\overline{x}) - F(x^*)] \leq \rho^S [F(\widetilde{x}^0) - F(x^*)].$$

The proof of Corollary 1 is given in the Supplementary Material. From Corollary 1, we can see that SVRG-SD has a linear convergence rate for SC problems. As discussed in [14], $\rho \approx \frac{L/\mu}{\nu(1-4\nu)m} + \frac{4\nu}{1-4\nu}$ for the proximal variant of SVRG [14], where $\nu = 1/\alpha$. For a reasonable comparison, we use the same parameter settings for SVRG and SVRG-SD, e.g., $\alpha = 19$ and $m = 57L/\mu$. Then we can see that $\rho_{\text{SVRG}} \approx 31/45$ for SVRG and $\rho_{\text{SVRG-SD}} \approx 7/(14+36\widehat{\beta}) < 1/2$ for SVRG-SD, that is, $\rho_{\text{SVRG-SD}}$ is smaller than ρ_{SVRG}. Note that setting α to 19 is for the analysis only and not necessary in practice. That is, α can be set to a much smaller value, e.g., $\alpha = 1$, which means that SVRG-SD can use a much larger step size, e.g., $\eta = 1/L$ for SVRG-SD vs. $\eta = 1/(10L)$ for SVRG. Thus, SVRG-SD can significantly improve the convergence rate of SVRG and Prox-SVRG in practice, which will be confirmed by our experiments in Section 6.

4.2 Convergence Property for Non-SC Problems

Unlike most of the stochastic variance reduction methods [3, 14], including SVRG and Prox-SVRG, which only have the convergence guarantees for some SC problems, the convergence result of SVRG-SD for the non-SC case is also provided, as shown below.

Theorem 2 (Non-SC). *Suppose each $f_i(\cdot)$ is L-smooth. Setting $\alpha = 19$, $\sigma = 1/2$, and m sufficiently large, then*

$$\mathbb{E}[F(\overline{x}) - F(x^*)] \leq \frac{171L}{(16+40\widehat{\beta})mS}\|x^* - \widetilde{x}^0\|^2$$

$$+ \left(\frac{9}{(4+8\widehat{\beta})mS} + \frac{1}{(2+4\widehat{\beta})S}\right)[F(\widetilde{x}^0) - F(x^*)].$$

Algorithm 3 SAGA-SD

Input: the number of epochs S, the number of iterations m per epoch, and the step size η.
Initialize: $\widetilde{x}^0$.
1: **for** $s = 1, 2, \ldots S$ **do**
2:    $x_0^s = \widehat{x}_0^s = \widetilde{x}^{s-1}$;
3:    **for** $k = 1, \ldots, m$ **do**
4:      Pick i_k^s uniformly at random from $[n]$;
5:      Take $\phi_{i_k^s}^k = x_{k-1}^s$ and store $\nabla f_{i_k^s}(\phi_{i_k^s}^k)$ in the table;
6:    $\widetilde{\nabla}f_{i_k^s}(x_{k-1}^s) = \nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\phi_{i_k^s}^{k-1}) + \frac{1}{n}\sum_j \nabla f_j(\phi_j^{k-1})$;
7:      $y_k^s = \text{prox}_\eta^r\left(x_{k-1}^s - \eta\widetilde{\nabla}f_{i_k^s}(x_{k-1}^s)\right)$;
8:      $\theta_k = \arg\min_{\theta\in\mathbb{R}} F(\theta x_{k-1}^s) + \frac{\zeta(1-\theta)^2}{2}\|\nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\phi_{i_k^s}^{k-1})\|^2$;
9:      $x_k^s = y_k^s + (1-\sigma)(\widehat{x}_k^s - \widehat{x}_{k-1}^s)$ and $\widehat{x}_k^s = \theta_k x_{k-1}^s$;
10:   **end for**
11:   $\widetilde{x}^s = \frac{1}{m}\sum_{k=1}^m \widehat{x}_k^s$;
12: **end for**
Output: $\overline{x} = \widetilde{x}^S$

The proof of Theorem 2 is provided in the Supplementary Material. The constant $\widehat{\beta} \geq 0$ is from the sufficient decrease strategy, which thus implies that the convergence bound in Theorem 2 can be further improved using our sufficient decrease strategy with an even larger $\widehat{\beta}$. Theorem 2 shows that SVRG-SD attains the convergence rate of $\mathcal{O}(1/S)$ for the non-strongly convex objective function (1). Although SVRG-SD is guaranteed to have a slower theoretical convergence rate than the accelerated stochastic variance reduction method, Katyusha [18], whose convergence rate is $\mathcal{O}(1/S^2)$, SVRG-SD usually converges much faster than Katyusha in practice (see Section 6.3 for details).

5 Extensions and Implementations

5.1 SAGA-SD

As mentioned above, we can use the proposed sufficient decrease technique to accelerate other stochastic variance reduction methods, e.g., SAGA [9], and present a novel variant of SAGA with sufficient decrease (called SAGA-SD), as shown in Algorithm 3. Like SVRG-SD, SAGA-SD is also a two-stage algorithm, whereas the original SAGA is a single-stage algorithm. More specifically, for SAGA-SD, the cost function with respect to θ in (5) can be revised by simply replacing $\nabla f_{i_k^s}(\widetilde{x}^{s-1})$ with $\nabla f_{i_k^s}(\phi_{i_k^s}^{k-1}) = g_{i_k}^{k-1}$ defined in (3a). Moreover, Property 1 can be extended for SAGA-SD by setting $\widetilde{p}_{i_k^s} = \nabla f_{i_k^s}(x_{k-1}^s) - \nabla f_{i_k^s}(\phi_{i_k^s}^s)$, as well as for other stochastic variance reduction algorithms such as SAG. The main differences between SVRG-SD and SAGA-SD are the stochastic gradient estimators in (9) and

(3a), and the update rule of the sufficient decrease coefficient in (5). For the convergence analysis of SAGA-SD, we refer to [38].

Similar to Katyusha [18], SVRG-SD and SAGA-SD trivially extend to the mini-batch setting. As suggested in [39] and [16], we can add a proximal term into a non-strongly convex objective function $F(x)$ as follows: $F_\tau(x,y) = f(x) + \frac{\tau}{2}\|x-y\|^2 + r(x)$, where $\tau \geq 0$ is a constant that can be determined as in [16, 39], and $y \in \mathbb{R}^d$ is a proximal point. Then the condition number of this proximal function $F_\tau(x,y)$ can be much smaller than that of the original function $F(x)$, if τ is sufficiently large. However, adding the proximal term may degrade the performance of the involved algorithms both in theory and in practice [19]. Therefore, we directly use SVRG-SD and SAGA-SD to solve the non-strongly convex objective function (1). If some component functions in (1) are non-smooth (e.g., support vector machine), we can use the proximal operator oracle [40] or the Nesterov's smoothing [41] technique to smooth them, and thereby obtain the smoothed approximations of the functions $f_i(x)$.

5.2 Efficient Implementations

Both (10) and (11) require the calculation of $b^T A$, thus we need to precompute and save $b^T A$ in the initial stage. To further reduce the computational complexity of $\|Ax_{k-1}^s\|^2$ in (10) and (11), we use the fast partial singular value decomposition (SVD) to obtain the best rank-r approximation $U_r S_r V_r^T$ to A and save $S_r V_r^T$. Then $\|Ax_{k-1}^s\| \approx \|S_r V_r^T x_{k-1}^s\|$. In practice, e.g., in our experiments, r can be set to a small number to capture 99.5% of the spectral energy of the data matrix A, e.g., $r = 10$ for the Covtype data set, similar to inexact line search methods [28] for deterministic optimization.

For high-dimensional sparse data A (e.g., Rcv1), we use the lazy computing trick for the calculation of $\|Ax_{k-1}^s\|$, as well as $b^T A$, instead of the partial SVD. That is, we calculate $\|Ax_{k-1}^s\|$ and $b^T A$ only for the non-zero dimensions of each sample, rather than all dimensions. In particular, we can introduce the lazy update tricks in [42] to our algorithms, and perform the update steps only for the non-zero dimensions of each example. That is, the average per-iteration complexity can be improved from $O(d)$ to $O(d')$ as stated below, where $d' \leq d$ is the sparsity of feature vectors.

The time complexity of each inter-iteration in SVRG-SD, as well as SAGA-SD, with full sufficient decrease is $O(rd)$, which is a little higher than that of SVRG. In fact, we can just randomly select only a small fraction (e.g., $1/10^3$) of stochastic gradient iterations in each epoch to update with sufficient decrease, while the remaining iterations are executed without sufficient de-
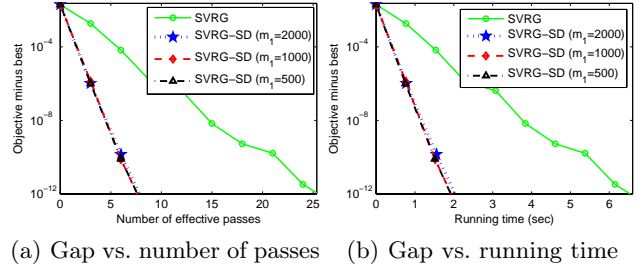
(a) Gap vs. number of passes   (b) Gap vs. running time

Figure 1: Comparison of SVRG and SVRG-SD with different values of m_1 for solving ridge regression problems on the Covtype dataset.

crease technique, i.e., $\widehat{x}_k^s = x_{k-1}^s$. Let m_1 be the number of iterations with our sufficient decrease technique in each epoch. By fixing $m_1 = \lfloor m/10^3 \rfloor$ and thus without increasing parameters tuning difficulties, SVRG-SD[1] can always converge much faster than its counterpart, SVRG, as shown in Figure 1 (similar results are also observed for SAGA-SD, as shown in the Supplementary Materials). It is easy to see that our algorithms are very robust with respect to the choice of m_1, and achieve average per-iteration time complexity as low as the original SVRG and SAGA. Thus, we mainly consider SVRG-SD and SAGA-SD with partial sufficient decrease for real-world machine learning applications.

6 Experiments

In this section, we evaluate the performance of SVRG-SD and SAGA-SD, and compare their performance with their counterparts including SVRG [3], its proximal variant (Prox-SVRG) [14], and SAGA [9]. Moreover, we also report the performance of the well-known accelerated stochastic variance reduction methods, Catalyst [16] and Katyusha [18].

Table 1: Summary of dense and sparse data sets.

Data sets	Sizes n	Dimensions d	Sparsity
Ijcnn1	49,990	22	59.09%
Rcv1	20,242	47,236	0.16%
Covtype	581,012	54	22.12%
SUSY	5,000,000	18	98.82%

6.1 Experimental Setup

In our experiments, we used the four publicly available data sets: Ijcnn1, Rcv1, Covtype, and SUSY, all of which can be downloaded from the LIBSVM Data website. Note that each sample of all the date sets

---

[1]Note that SVRG-SD with partial sufficient decrease possesses the similar convergence properties as SVRG-SD with full sufficient decrease because Property 1 still holds in the case when $\theta_k = 1$.

Fanhua Shang,   Yuanyuan Liu*,   Kaiwen Zhou,   James Cheng,   Kelvin K.W. Ng
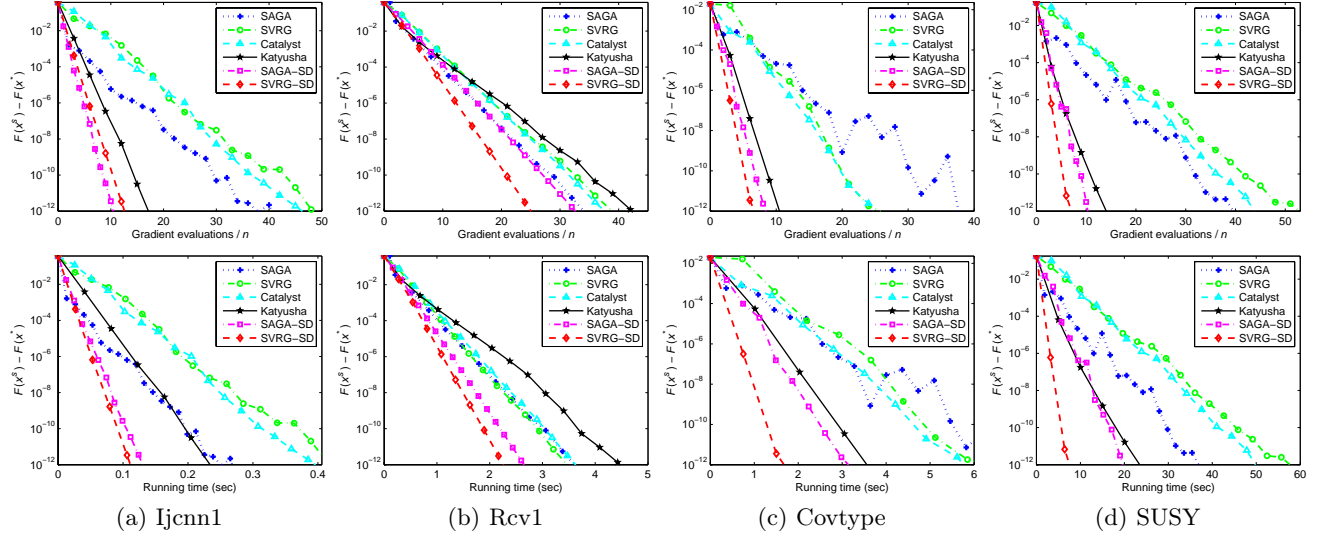
Figure 2: Comparison of the stochastic variance reduction methods for solving ridge regression problems ($\lambda_1 = 10^{-4}$) on Ijcnn1, Rcv1, Covtype, and SUSY. The vertical axis is the objective value minus the minimum, and the horizontal axis denotes the number of effective passes over the data (top) or running time (bottom).

was normalized so that they have unit length as in [14], which leads to the same upper bound on the Lipschitz constants L_i, i.e., $L = L_i$ for all $i = 1, \ldots, n$. This step is for comparison only and not necessary in practice. As suggested in [3, 14, 18], the epoch length is set to $m = 2n$ for the stochastic variance reduced methods, SVRG [3], Prox-SVRG [14], Catalyst [16], and Katyusha [18], as well as SVRG-SD. In addition, unlike SAGA [9], we fixed $m = n$ for each epoch of SAGA-SD. Then the only parameter we have to tune by hand is the step size, η. More specifically, we select step sizes from $\{10^j, 2.5 \times 10^j, 5 \times 10^j, 7.5 \times 10^j, 10^{j+1}\}$ as in [18], where $j \in \{-2, -1, 0\}$. Each of these methods had its step size parameter chosen so as to give the fastest convergence. Since Katyusha has a much higher per-iteration complexity than SVRG, we compare their performance in terms of both the number of effective passes and running time (seconds), where computing a single full gradient or evaluating n component gradients is considered as one effective pass over the data. For fair comparison, we implemented all the methods in C++ with a Matlab interface (all codes are made available, see link in the Supplementary Materials), and performed all the experiments on a PC with an Intel i5-2400 CPU and 16GB RAM.

6.2 Ridge Regression

In this part, we focus on the following ridge regression (i.e., $\lambda_2 \equiv 0$) problem as the SC example:

$$\min_{x \in \mathbb{R}^d} \frac{1}{2n} \sum_{i=1}^n (a_i^T x - b_i)^2 + \frac{\lambda_1}{2} \|x\|^2 + \lambda_2 \|x\|_1, \quad (12)$$

where $\lambda_1, \lambda_2 \geq 0$ are the regularization parameters.
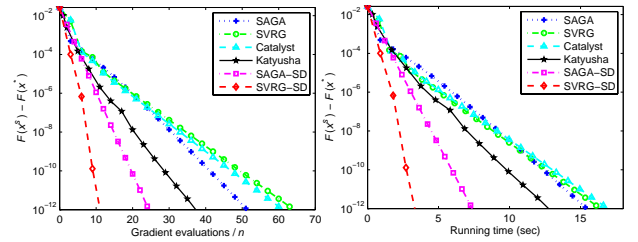

Figure 3: Comparison of the stochastic variance reduction methods for solving ridge regression problems with $\lambda_1 = 10^{-6}$ on Covtype.

Figure 2 shows how the objective gap, i.e., $F(x^s) - F(x^*)$, of all these algorithms decreases for ridge regression problems with the regularization parameters $\lambda_1 = 10^{-4}$ and $\lambda_2 \equiv 0$ (more results are given in the Supplementary Material). For SVRG-SD and SAGA-SD, we set $\sigma = 1/2$ on the four data sets. As seen in Figure 2, SVRG-SD and SAGA-SD achieve consistent speedups for all the data sets, and significantly outperform their counterparts, SVRG and SAGA, in all the settings. This confirms that our stochastic sufficient decrease technique is able to accelerate SVRG and SAGA. Impressively, SVRG-SD and SAGA-SD usually converge much faster than the well-known accelerated methods, Catalyst and Katyusha, which further justifies the effectiveness of our sufficient decrease technique. As SVRG-SD and SAGA-SD have a much lower per-iteration complexity than Katyusha, they have more obvious advantage over Katyusha in terms of running time, especially for the case of sparse data (e.g., Rcv1), as shown in Fig. 2(b).
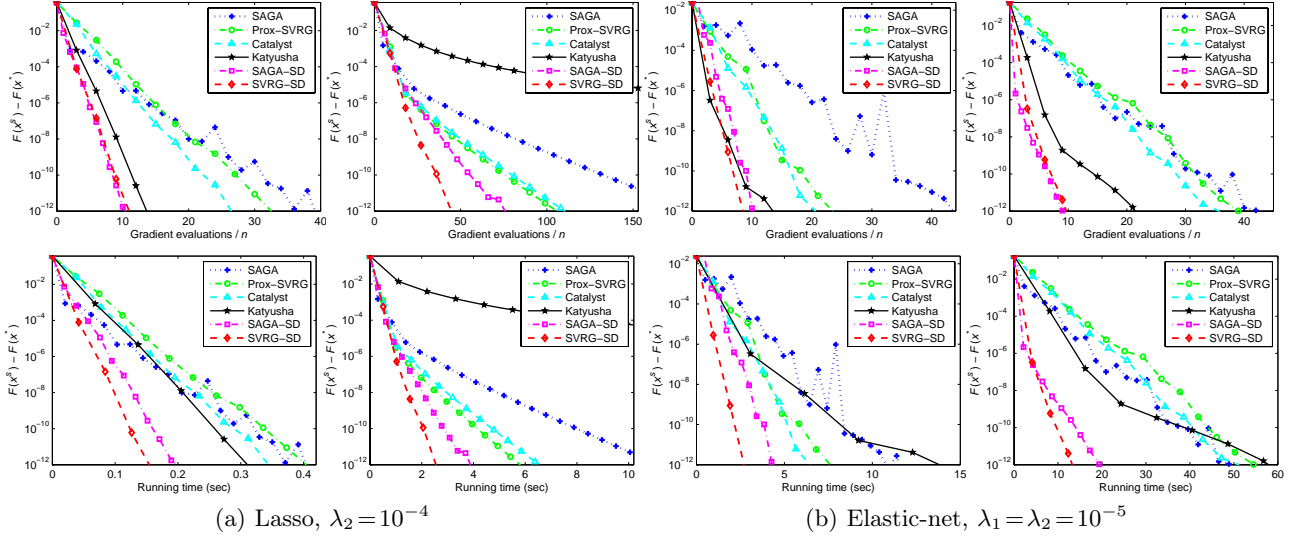
We also compared the performance of all these meth-

Figure 4: Comparison of all the stochastic methods for solving Lasso and elastic-net problems on Ijcnn1 (first column), Rcv1 (second column), Covtype (third column), and SUSY (last column).

ods, when the regularization parameter is relatively small, as shown in Figure 3. The result shows that Katyusha, SVRG-SD and SAGA-SD converge significantly faster than the other methods. In particular, SVRG-SD and SAGA-SD also outperform the well-known accelerated methods, Catalyst and Katyusha, which further verifies the importance of our sufficient decrease technique for stochastic optimization.

6.3 Lasso and Elastic-Net

Finally, we conducted some experiments for the generalized regression problem (12), including Lasso (i.e., $\lambda_1 \equiv 0$) and elastic-net (i.e., $\lambda_1 \neq 0$ and $\lambda_2 \neq 0$) problems as non-SC and non-smooth examples.

We plot some representative results in Figure 4 (see the Supplementary Material for more results), which show that SVRG-SD and SAGA-SD significantly outperform their counterparts (i.e., Prox-SVRG and SAGA) in all the settings, and are considerably better than the accelerated methods, Catalyst and Katyusha, in most cases. This empirically verifies that our sufficient decrease technique allows us to take much larger step sizes for SVRG-SD and SAGA-SD than the original SVRG/Prox-SVRG and SAGA (e.g., $\eta = 1$ for SVRG-SD vs. $\eta = 0.4$ for SVRG), and can accelerate them for solving both SC and non-SC objective functions.

7 Conclusions & Future Work

To the best of our knowledge, this is the first work to design an efficient sufficient decrease technique for stochastic optimization. Moreover, we proposed two different schemes for Lasso and ridge regression to ef-

ficiently update the coefficient θ, which takes the important decisions to shrink, expand or move in the opposite direction. This is very different from adaptive learning rate methods, e.g., [43], and line search methods, e.g., [29], all of which cannot address the issue in Section 3.1 whatever value the step size is. Unlike most stochastic variance reduction methods [3, 8, 14], which only have convergence guarantees for SC problems[2], we provided the convergence guarantees of our algorithm for both SC and non-SC objective functions. Experimental results verified the effectiveness of our sufficient decrease technique for stochastic optimization, especially for the case when the regularization parameter is relatively small (or the condition number $\kappa := L/\mu$ is relatively large). Naturally, it can also be used to further speed up accelerated methods such as [16, 18, 19, 44].

As each component function $f_i(\cdot)$ may have different degrees of smoothness, to select the random index i_k^s from a non-uniform distribution is a much better choice than simple uniform random sampling [4], as well as without-replacement sampling vs. with-replacement sampling [45]. On the practical side, both our algorithms tackle the non-SC and non-smooth problems directly, without using any quadratic regularizer as in [16, 18], as well as the proximal setting. Note that some asynchronous parallel and distributed variants [46, 47, 48, 49] of stochastic variance reduction methods have also been proposed for such stochastic settings. We leave these variations out from our comparison and consider similar extensions to our stochastic sufficient decrease method as future work.

---

[2][15] proposed a simple SVRG version with convergence guarantees for both SC and non-SC problems as in SAGA.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[2] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, pages 919–926, 2004.

[3] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

[4] P. Zhao and T. Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *ICML*, pages 1–9, 2015.

[5] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, pages 449–456, 2012.

[6] O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML*, pages 71–79, 2013.

[7] N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, pages 2672–2680, 2012.

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

[9] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654, 2014.

[10] A. Defazio, T. Caetano, and J. Domke. Finito: A faster, permutable incremental gradient method for big data problems. In *ICML*, pages 1125–1133, 2014.

[11] J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM J. Optim.*, 25(2):829–855, 2015.

[12] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Math. Program.*, 162:83–112, 2017.

[13] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Math. Program.*, 155:105–145, 2016.

[14] L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM J. Optim.*, 24(4):2057–2075, 2014.

[15] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams. Variance reduced stochastic gradient descent with neighbors. In *NIPS*, pages 2296–2304, 2015.

[16] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *NIPS*, pages 3366–3374, 2015.

[17] A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *NIPS*, pages 1574–1582, 2014.

[18] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *STOC*, pages 1200–1205, 2017.

[19] Z. Allen-Zhu and Y. Yuan. Improved SVRG for non-strongly-convex or sum-of-non-convex objectives. In *ICML*, pages 1080–1089, 2016.

[20] R. Babanezhad, M. O. Ahmed, A. Virani, M. Schmidt, J. Konecny, and S. Sallinen. Stop wasting my gradients: Practical SVRG. In *NIPS*, pages 2242–2250, 2015.

[21] L. Zhang, M. Mahdavi, and R. Jin. Linear convergence with condition number independent access of full gradients. In *NIPS*, pages 980–988, 2013.

[22] F. Shang, Y. Liu, J. Cheng, and J. Zhuo. Fast stochastic variance reduced gradient method with momentum acceleration for machine learning. *arXiv:1703.07948v2*, 2017.

[23] E. Hazan and H. Luo. Variance-reduced and projection-free stochastic optimization. In *ICML*, pages 1263–1271, 2016.

[24] B. Woodworth and N. Srebro. Tight complexity bounds for optimizing composite objectives. In *NIPS*, pages 3639–3647, 2016.

[25] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In *NIPS*, pages 379–387, 2015.

[26] P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2):226–235, 1969.

[27] A. Defazio. A simple practical accelerated method for finite sums. In *NIPS*, pages 676–684, 2016.

[28] J. More and D. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM T. Math. Software*, 20:286–307, 1994.

[29] M. Mahsereci and P. Hennig. Probabilistic line searches for stochastic optimization. In *NIPS*, pages 181–189, 2015.

[30] Y. Liu, F. Shang, and J. Cheng. Accelerated variance reduced stochastic ADMM. In *AAAI*, pages 2287–2293, 2017.

[31] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Doklady*, 27:372–376, 1983.

[32] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publ., Boston, 2004.

[33] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[34] W. Su, S. Boyd, and E. J. Candes. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. *J. Mach. Learn. Res.*, 17:5312–5354, 2016.

[35] D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41(3):613–627, 1995.

[36] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific J. Math.*, 16(1):1–3, 1966.

[37] F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *NIPS*, pages 773–781, 2013.

[38] F. Shang, Y. Liu, J. Cheng, K. Ng, and Y. Yoshida. Guaranteed sufficient decrease for variance reduced stochastic gradient descent. *arXiv: 1703.06807v2*, 2017.

[39] R. Frostig, R. Ge, S. M. Kakade, and A. Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML*, pages 2540–2548, 2015.

[40] Z. Allen-Zhu and E. Hazan. Optimal black-box reductions between optimization objectives. In *NIPS*, pages 1606–1614, 2016.

[41] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103:127–152, 2005.

[42] Jakub Koneeny, Jie Liu, Peter Richtarik, , and Martin Takae. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE J. Sel. Top. Sign. Proces.*, 10(2):242–255, 2016.

[43] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[44] F. Shang, K. Zhou, J. Cheng, I. W. Tsang, L. Zhang, and D. Tao. VR-SGD: A simple stochastic variance reduction method for machine learning. *arXiv:1704.04966v2*, 2017.

[45] O. Shamir. Without-replacement sampling for stochastic gradient methods. In *NIPS*, pages 46–54, 2016.

[46] S. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola. On variance reduction in stochastic gradient descent and its asynchronous variants. In *NIPS*, pages 2629–2637, 2015.

[47] H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. I. Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM J. Optim.*, 27(4):2202–2229, 2017.

[48] F. Pedregosa, R. Leblond, and S. Lacoste-Julien. Breaking the nonsmooth barrier: A scalable parallel method for composite optimization. In *NIPS*, pages 55–64, 2017.

[49] J. D. Lee, Q. Lin, T. Ma, and T. Yang. Distributed stochastic variance reduced gradient methods by sampling extra data with replacement. *J. Mach. Learn. Res.*, 18:1–43, 2017.