ASVRG: Accelerated Proximal SVRG

Fanhua Shang FHSHANG@XIDIAN.EDU.CN
Licheng Jiao LCHJIAO@MAIL.XIDIAN.EDU.CN
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, China

Kaiwen Zhou KWZHOU@CSE.CUHK.EDU.HK
James Cheng JCHENG@CSE.CUHK.EDU.HK
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong

Yan Ren CRANE_ROCK@OUTLOOK.COM
Yufei Jin JESTY@JESTYF.CN
School of Computer Science and Technology, Xidian University, China

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

This paper proposes an accelerated proximal stochastic variance reduced gradient (ASVRG) method, in which we design a simple and effective momentum acceleration trick. Unlike most existing accelerated stochastic variance reduction methods such as Katyusha, ASVRG has only one additional variable and one momentum parameter. Thus, ASVRG is much simpler than those methods, and has much lower per-iteration complexity. We prove that ASVRG achieves the best known oracle complexities for both strongly convex and non-strongly convex objectives. In addition, we extend ASVRG to mini-batch and non-smooth settings. We also empirically verify our theoretical results and show that the performance of ASVRG is comparable with, and sometimes even better than that of the state-of-the-art stochastic methods.

Keywords: Stochastic convex optimization, momentum acceleration, proximal stochastic gradient, variance reduction, SVRG, Prox-SVRG, Katyusha

1. Introduction

Consider the following composite convex minimization:

$$\min_{x\in\mathbb{R}^d} F(x) := f(x) + g(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) + g(x), \tag{1}$$

where $f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ is a convex function that is a finite average of n convex component functions $f_i(x) : \mathbb{R}^d \to \mathbb{R}$, and $g(x)$ is a "simple" possibly non-smooth convex function. This formulation naturally arises in many problems in machine learning, optimization and signal processing, such as regularized empirical risk minimization (ERM)) and eigenvector computation (Shamir, 2015; Garber et al., 2016). To solve Problem (1) with a large sum of n component functions, computing the full gradient of $f(x)$ in first-order methods is expensive, and hence stochastic gradient descent (SGD) has been widely applied to many large-scale problems (Zhang, 2004; Krizhevsky et al., 2012). The update rule of proximal SGD is

$$x_t = \mathop{\arg\min}_{y\in\mathbb{R}^d} \left\{ \frac{1}{2\eta_t}\|y - x_{t-1}\|^2 + y^T\nabla f_{i_t}(x_{t-1}) + g(y) \right\}, \tag{2}$$

where $\eta_t \propto 1/\sqrt{t}$ is the step size, and i_t is chosen uniformly at random from $\{1, \ldots, n\}$. When $g(x) \equiv 0$, the update rule in (2) becomes $x_t = x_{t-1} - \eta_t\nabla f_{i_t}(x_{t-1})$. The standard SGD estimates the gradient from just one example (or a mini-batch), and thus it enjoys a low per-iteration cost as opposed to full gradient methods. The expectation of $\nabla f_i(x_t)$ is an unbiased estimation to $\nabla f(x_t)$, i.e., $\mathbb{E}[\nabla f_i(x_t)] = \nabla f(x_t)$. However, the variance of the stochastic gradient estimator may be large, which leads to slow convergence (Johnson and Zhang, 2013). Even under the strongly convex (SC) and smooth conditions, standard SGD can only attain a sub-linear rate of convergence (Rakhlin et al., 2012; Shamir and Zhang, 2013).

Recently, the convergence rate of SGD has been improved by many variance reduced SGD methods (Roux et al., 2012; Shalev-Shwartz and Zhang, 2013; Johnson and Zhang, 2013; Defazio et al., 2014a; Mairal, 2015) and their proximal variants (Schmidt et al., 2013; Xiao and Zhang, 2014; Shalev-Shwartz and Zhang, 2016). The methods use past gradients to progressively reduce the variance of stochastic gradient estimators, so that a constant step size can be used. In particular, these variance reduced SGD methods converge linearly for SC and Lipschitz-smooth problems. SVRG (Johnson and Zhang, 2013) and its proximal variant, Prox-SVRG (Xiao and Zhang, 2014), are particularly attractive because of their low storage requirement compared with the methods in (Roux et al., 2012; Defazio et al., 2014a; Shalev-Shwartz and Zhang, 2016), which need to store all the gradients of the n component functions $f_i(\cdot)$ (or dual variables), so that $O(nd)$ storage is required in general problems. At the beginning of each epoch of SVRG and Prox-SVRG, the full gradient $\nabla f(\widetilde{x})$ is computed at the past estimate $\widetilde{x}$. Then the key update rule is given by

$$\begin{aligned} \widetilde{\nabla} f_{i_t}(x_{t-1}) &= \nabla f_{i_t}(x_{t-1}) - \nabla f_{i_t}(\widetilde{x}) + \nabla f(\widetilde{x}), \\ x_t &= \mathop{\arg\min}_{y\in\mathbb{R}^d} \left\{ \frac{1}{2\eta}\|y - x_{t-1}\|^2 + y^T\widetilde{\nabla} f_{i_t}(x_{t-1}) + g(y) \right\}. \end{aligned} \tag{3}$$

For SC problems, the oracle complexity (total number of component gradient evaluations to find ε-suboptimal solutions) of most variance reduced SGD methods is $\mathcal{O}((n+L/\mu)\log(1/\varepsilon))$, when each $f_i(x)$ is L-smooth and $g(x)$ is μ-strongly convex. Thus, there still exists a gap between the oracle complexity and the upper bound in (Woodworth and Srebro, 2016). In theory, they also converge slower than accelerated deterministic algorithms (e.g., FISTA (Beck and Teboulle, 2009)) for non-strongly convex (non-SC) problems, namely $\mathcal{O}(1/t)$ vs. $\mathcal{O}(1/t^2)$.

Very recently, several advanced techniques were proposed to further speed up the variance reduction SGD methods mentioned above. These techniques mainly include the Nesterov's acceleration technique (Nitanda, 2014; Lin et al., 2015; Murata and Suzuki, 2017; Lan and Zhou, 2018), the projection-free property of the conditional gradient method (Hazan and Luo, 2016), reducing the number of gradient calculations in the early iterations (Babanezhad et al., 2015; Allen-Zhu and Yuan, 2016; Shang et al., 2017), and the momentum acceleration trick (Hien et al., 2018; Allen-Zhu, 2018; Zhou et al., 2018; Shang et al., 2018b). Lin et al. (2015) and Frostig et al. (2015) proposed two accelerated algorithms with improved oracle complexity of $\mathcal{O}((n + \sqrt{nL/\mu})\log(L/\mu)\log(1/\varepsilon))$ for SC problems. In particular, Katyusha (Allen-Zhu, 2018) attains the optimal oracle complexities of $\mathcal{O}((n+\sqrt{nL/\mu})\log(1/\varepsilon))$ and $\mathcal{O}(n\log(1/\varepsilon)+\sqrt{nL/\varepsilon})$ for SC and non-SC problems, respectively. The main update rules of Katyusha are formulated as follows:

$$\begin{aligned} x_t &= y_{t-1} + \omega_1(z_{t-1} - y_{t-1}) + \omega_2(\widetilde{x} - y_{t-1}), \\ y_t &= \arg\min_{y\in\mathbb{R}^d} \left\{ \frac{3L}{2}\|y - x_t\|^2 + y^T\widetilde{\nabla} f_{i_t}(x_t) + g(y) \right\}, \\ z_t &= \arg\min_{z\in\mathbb{R}^d} \left\{ \frac{1}{2\eta}\|z - z_{t-1}\|^2 + z^T\widetilde{\nabla} f_{i_t}(x_t) + g(z) \right\}, \end{aligned} \tag{4}$$

where $\omega_1, \omega_2 \in [0, 1]$ are two parameters for the key momentum terms, and $\eta = 1/(3\omega_1 L)$. Note that the parameter ω_2 is fixed to 0.5 in (Allen-Zhu, 2018) to avoid parameter tuning.

Our Contributions. In spite of the success of momentum acceleration tricks, most of existing accelerated methods including Katyusha require at least two auxiliary variables and two corresponding momentum parameters (e.g., for the Nesterov's momentum and Katyusha momentum in (4)), which lead to complicated algorithm design and high per-iteration complexity. We address the weaknesses of the existing methods by a simper accelerated proximal stochastic variance reduced gradient (ASVRG) method, which requires only one auxiliary variable and one momentum parameter. Thus, ASVRG leads to much simpler algorithm design and is more efficient than the accelerated methods. Impressively, ASVRG attains the same low oracle complexities as Katyusha for both SC and non-SC objectives. We summarize our main contributions as follows.

- We design a simple momentum acceleration trick to accelerate the original SVRG. Different from most accelerated algorithms such as Katyusha, which require two momentums mentioned above, our update rule has only one momentum accelerated term.
- We prove that ASVRG converges to an ε-minimizer with the oracle complexity of $\mathcal{O}((n+\sqrt{nL/\mu})\log(1/\varepsilon))$ for SC problems, which is the same as that in (Defazio, 2016; Allen-Zhu, 2018), and matches the upper bound in (Woodworth and Srebro, 2016).
- We also prove that ASVRG achieves the optimal convergence rate of $\mathcal{O}(1/t^2)$ and the oracle complexity of $\mathcal{O}(n\log(1/\varepsilon)+\sqrt{nL/\varepsilon})$ for non-SC problems, which is identical to the best known result in (Hien et al., 2018; Allen-Zhu, 2018).
- Finally, we introduce mini-batching, adaptive regularization and smooth techniques into our algorithms, and further analyze their convergence properties and summarize the oracle complexities of ASVRG for the four cases of Problem (1) in Table 1.

2. Preliminaries

Throughout this paper, we use $\|\cdot\|$ to denote the standard Euclidean norm. $\nabla f(x)$ denotes the full gradient of $f(x)$ if it is differentiable, or $\partial f(x)$ the subgradient if $f(x)$ is Lipschitz continuous. We mostly focus on the case of Problem (1) when each component function $f_i(x)$ is L_i-smooth[1], and $g(x)$ is μ-strongly convex.

1. In the following, we mainly consider the more general class of Problem (1), when every $f_i(x)$ can have different degrees of smoothness, rather than the gradients of all component functions having the same Lipschitz constant L.

Table 1: Comparison of the oracle complexities of some stochastic variance reduced algorithms for the four classes of problems in Section 2. Note that $C_1 = \|\widetilde{x}^0 - x^\star\|$ and $C_2 = F(\widetilde{x}^0) - F(x^\star)$.

	L-Smooth		G-Lipschitz	
	μ-Strongly Convex	non-SC	μ-Strongly Convex	non-SC
SVRG	$\mathcal{O}\big((n+L/\mu)\log\frac{C_2}{\varepsilon}\big)$	unknown	unknown	unknown
SAGA	$\mathcal{O}\big((n+L/\mu)\log\frac{C_2}{\varepsilon}\big)$	$\mathcal{O}\big(nC_2/\varepsilon + LC_1^2/\varepsilon\big)$	unknown	unknown
Katyusha	$\mathcal{O}\Big((n+\sqrt{nL/\mu})\log\frac{C_2}{\varepsilon}\Big)$	$\mathcal{O}\Big(n\log\frac{C_2}{\varepsilon} + C_1\sqrt{nL/\varepsilon}\Big)$	$\mathcal{O}\Big(n\log\frac{C_2}{\varepsilon} + \frac{\sqrt{n}G}{\sqrt{\mu\varepsilon}}\Big)$	$\mathcal{O}\Big(n\log\frac{C_2}{\varepsilon} + \frac{\sqrt{n}C_1 G}{\varepsilon}\Big)$
ASVRG	$\mathcal{O}\Big((n+\sqrt{nL/\mu})\log\frac{C_2}{\varepsilon}\Big)$	$\mathcal{O}\Big(n\log\frac{C_2}{\varepsilon} + C_1\sqrt{nL/\varepsilon}\Big)$	$\mathcal{O}\Big(n\log\frac{C_2}{\varepsilon} + \frac{\sqrt{n}G}{\sqrt{\mu\varepsilon}}\Big)$	$\mathcal{O}\Big(n\log\frac{C_2}{\varepsilon} + \frac{\sqrt{n}C_1 G}{\varepsilon}\Big)$

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

Assumption 2 *$g(\cdot)$ is μ-strongly convex (μ-SC), if there exists a constant $\mu > 0$ such that for any $x, y \in \mathbb{R}^d$ and any (sub)gradient ξ (i.e., $\xi = \nabla g(x)$, or $\xi \in \partial g(x)$) of $g(\cdot)$ at x*

$$g(y) \geq g(x) + \xi^T (y - x) + \frac{\mu}{2}\|x - y\|^2.$$

For a non-strongly convex function, the above inequality can always be satisfied with $\mu = 0$. As summarized in (Allen-Zhu and Hazan, 2016), there are mainly four interesting cases of Problem (1):

- Case 1: Each $f_i(x)$ is L_i-smooth and $g(x)$ is μ-SC, e.g., ridge regression, logistic regression, and elastic net regularized logistic regression.
- Case 2: Each $f_i(x)$ is L_i-smooth and $g(x)$ is non-SC, e.g., Lasso and ℓ_1-norm regularized logistic regression.
- Case 3: Each $f_i(x)$ is non-smooth (but Lipschitz continuous) and $g(x)$ is μ-SC, e.g., linear support vector machine (SVM).
- Case 4: Each $f_i(x)$ is non-smooth (but Lipschitz continuous) and $g(x)$ is non-SC, e.g., ℓ_1-norm SVM.

3. Accelerated Proximal SVRG

In this section, we propose an accelerated proximal stochastic variance reduced gradient (ASVRG) method with momentum acceleration for solving both strongly convex and non-strongly convex objectives (e.g., Cases 1 and 2). Moreover, ASVRG incorporates a weighted sampling strategy as in (Xiao and Zhang, 2014; Zhao and Zhang, 2015; Shamir, 2016) to randomly pick i_t based on a general distribution $\{p_i, \ldots, p_n\}$ rather than the uniform distribution.

Algorithm 1 ASVRG for strongly convex objectives

```
Input: The number of epochs $S$, the number of iterations $m$ per epoch, and the step size $\eta$.
Initialize: $\widetilde{x}^0 = x_0^0 = y_0^0$, $\omega$, $m_1$, $\rho \geq 1$, and the probability $P = [p_1, \ldots, p_n]$.
 1: for $s = 1, 2, \ldots, S$ do
 2:   $\widetilde{\mu} = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\widetilde{x}^{s-1})$;
 3:   Option I: $x_0^s = y_0^s = \widetilde{x}^{s-1}$;
 4:   for $t = 1, 2, \ldots, m_s$ do
 5:     Pick $i_t$ from $\{1, \ldots, n\}$ randomly based on $P$;
 6:     $\widetilde{\nabla} f_{i_t}(x_{t-1}^s) = [\nabla f_{i_t}(x_{t-1}^s) - \nabla f_{i_t}(\widetilde{x}^{s-1})]/(np_{i_t}) + \widetilde{\mu}$;
 7:     $y_t^s = \arg\min_y \left\{\langle \widetilde{\nabla} f_{i_t}(x_{t-1}^s),\, y - y_{t-1}^s\rangle + \frac{\omega}{2\eta}\|y - y_{t-1}^s\|^2 + g(y)\right\}$;
 8:     $x_t^s = \widetilde{x}^{s-1} + \omega(y_t^s - \widetilde{x}^{s-1})$;
 9:   end for
10:   $\widetilde{x}^s = \frac{1}{m_s}\sum_{t=1}^{m_s} x_t^s$, $m_{s+1} = \min(\lfloor \rho m_s \rfloor, m)$;
11:   Option II: $x_0^{s+1} = (1-\omega)\widetilde{x}^s + \omega y_{m_s}^s$, and $y_0^{s+1} = y_{m_s}^s$;
12: end for
Output: $\widetilde{x}^S$.
```

3.1. Iterate Averaging for Snapshot

Like SVRG, our algorithms are also divided into S epochs, and each epoch consists of m stochastic updates[2], where m is usually chosen to be $\Theta(n)$ as in Johnson and Zhang (2013). Within each epoch, a full gradient $\nabla f(\widetilde{x}^{s-1})$ is calculated at the snapshot $\widetilde{x}^{s-1}$. Note that we choose $\widetilde{x}^{s-1}$ to be the average of the past t stochastic iterates rather than the last iterate because it has been reported to work better in practice (Xiao and Zhang, 2014; Flammarion and Bach, 2015; Allen-Zhu and Yuan, 2016; Liu et al., 2017; Allen-Zhu, 2018; Shang et al., 2018b). In particular, one of the effects of the choice, i.e., $\widetilde{x}^s = \frac{1}{m}\sum_{t=1}^{m} x_t^s$, is to allow taking larger step sizes, e.g., $1/(3L)$ for ASVRG vs. $1/(10L)$ for SVRG.

3.2. ASVRG in Strongly Convex Case

We first consider the case of Problem (1) when each $f_i(\cdot)$ is L_i-smooth, and $g(\cdot)$ is μ-SC. Different from existing accelerated methods such as Katyusha (Allen-Zhu, 2018), we propose a much simpler accelerated stochastic algorithm with momentum, as outlined in Algorithm 1. Compared with the initialization of $x_0^s = y_0^s = \widetilde{x}^{s-1}$ (i.e., Option I in Algorithm 1), the choices of $x_0^{s+1} = (1-\omega)\widetilde{x}^s + \omega y_{m_s}^s$ and $y_0^{s+1} = y_{m_s}^s$ (i.e., Option II) also work well in practice.

3.2.1. MOMENTUM ACCELERATION

The update rule of y in our proximal stochastic gradient method is formulated as follows:

$$y_t^s = \arg\min_{y \in \mathbb{R}^d} \left\{\langle \widetilde{\nabla} f_{i_t}(x_{t-1}^s),\, y - y_{t-1}^s\rangle + \frac{\omega}{2\eta}\|y - y_{t-1}^s\|^2 + g(y)\right\}, \tag{5}$$

2. In practice, it was reported that reducing the number of gradient calculations in early iterations can lead to faster convergence (Babanezhad et al., 2015; Allen-Zhu and Yuan, 2016; Shang et al., 2017). Thus we set $m_{s+1} = \min(\lfloor \rho m_s \rfloor, m)$ in the early epochs of our algorithms, and fix $m_1 = n/4$, and $\rho = 2$ without increasing parameter tuning difficulties.

where $\omega \in [0, 1]$ is the momentum parameter. Note that the gradient estimator $\widetilde{\nabla} f_{i_t}$ used in this paper is the SVRG estimator in (3). Besides, the algorithms and convergence results of this paper can be generalized to the SAGA estimator in (Defazio et al., 2014a). When $g(x) \equiv 0$, the proximal update rule in (5) degenerates to $y_t^s = y_{t-1}^s - (\eta/\omega)\widetilde{\nabla} f_{i_t}(x_{t-1}^s)$.

Inspired by the Nesterov's momentum in (Nesterov, 1983, 2004; Nitanda, 2014; Shang et al., 2018a) and Katyusha momentum in (Allen-Zhu, 2018), we design a update rule for x as follows:

$$x_t^s = \widetilde{x}^{s-1} + \omega(y_t^s - \widetilde{x}^{s-1}). \tag{6}$$

The second term on the right-hand side of (6) is the proposed momentum similar to the Katyusha momentum in (Allen-Zhu, 2018). It is clear that there is only one momentum parameter ω in our algorithm, compared with the two parameters ω_1 and ω_2 in Katyusha[3].

The per-iteration complexity of ASVRG is dominated by the computation of $\nabla f_{i_t}(x_{t-1}^s)$ and $\nabla f_{i_t}(\widetilde{x}^{s-1})$,[4] and the proximal update in (5), which is as low as that of SVRG (Johnson and Zhang, 2013) and Prox-SVRG (Xiao and Zhang, 2014). In other words, ASVRG has a much lower per-iteration complexity than most of accelerated stochastic methods (Murata and Suzuki, 2017; Allen-Zhu, 2018) such as Katyusha (Allen-Zhu, 2018), which has one more proximal update for z in general.

3.2.2. MOMENTUM PARAMETER

Next we give a selection scheme for the stochastic momentum parameter ω. With the given η, ω can be a constant, and must satisfy the inequality: $0 < \omega \leq 1 - \frac{\widetilde{L}\eta}{1-\widetilde{L}\eta}$, where $\widetilde{L} = \max_j L_j/(np_j)$. As shown in Theorem 3 below, it is desirable to have a small convergence factor ρ. The following proposition obtains the optimal $\omega_\star$, which yields the smallest ρ value.

Proposition 1 *Given a suitable learning rate η, the optimal parameter $\omega_\star$ is $\omega_\star = m\mu\eta/2$.*

In fact, we can fix ω to a constant, e.g., 0.9, which works well in practice as in (Ruder, 2017). When $\omega = 1$ and $g(x)$ is smooth, Algorithm 1 degenerates to Algorithm 3 in the supplementary material, which is almost identical to SVRG (Johnson and Zhang, 2013), and the only differences between them are the choice of $\widetilde{x}^s$ and the initialization of x_0^s.

3.3. ASVRG in Non-Strongly Convex Case

We also develop an efficient algorithm for solving non-SC problems, as outlined in Algorithm 2. The main difference between Algorithms 1 and 2 is the setting of the momentum parameter. That is, the momentum parameter ω_s in Algorithm 2 is decreasing, while that of Algorithm 1 can be a constant. Different from Algorithm 1, ω_s in Algorithm 2 needs to satisfy the following inequalities:

$$\frac{1-\omega_s}{\omega_s^2} \leq \frac{1}{\omega_{s-1}^2} \text{ and } 0 \leq \omega_s \leq 1 - \frac{\widetilde{L}\eta}{1-\widetilde{L}\eta}. \tag{7}$$

3. Although Acc-Prox-SVRG (Nitanda, 2014) also has a momentum parameter, its oracle complexity is no faster than SVRG when the size of mini-batch is less than $n/2$, as discussed in (Allen-Zhu, 2018).
4. For some regularized ERM problems, we can save the intermediate gradients $\nabla f_{i_t}(\widetilde{x}^{s-1})$ in the computation of $\nabla f(\widetilde{x}^{s-1})$, which requires $O(n)$ storage in general as in (Defazio et al., 2014b).

Algorithm 2 ASVRG for non-strongly convex objectives

Input: The number of epochs S, the number of iterations m per epoch, and the step size η.

Initialize: $\widetilde{x}^0 = \widetilde{y}^0$, $\omega_0 = 1 - \frac{\widetilde{L}\eta}{1-\widetilde{L}\eta}$, m_1, $\rho \geq 1$, and $P = [p_1, \ldots, p_n]$.

```
 1: for $s = 1, 2, \ldots, S$ do
 2:   $\widetilde{\mu} = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\widetilde{x}^{s-1})$, $x_0^s = (1-\omega_{s-1})\widetilde{x}^{s-1} + \omega_{s-1}\widetilde{y}^{s-1}$, $y_0^s = \widetilde{y}^{s-1}$;
 3:   for $t = 1, 2, \ldots, m_s$ do
 4:     Pick $i_t$ from $\{1, \ldots, n\}$ randomly based on $P$;
 5:     $\widetilde{\nabla} f_{i_t}(x_{t-1}^s) = [\nabla f_{i_t}(x_{t-1}^s) - \nabla f_{i_t}(\widetilde{x}^{s-1})]/(np_{i_t}) + \widetilde{\mu}$;
 6:     $y_t^s = \arg\min_y \left\{\langle \widetilde{\nabla} f_{i_t}(x_{t-1}^s), y - y_{t-1}^s\rangle + \frac{\omega}{2\eta}\|y - y_{t-1}^s\|^2 + g(y)\right\}$;
 7:     $x_t^s = \widetilde{x}^{s-1} + \omega_{s-1}(y_t^s - \widetilde{x}^{s-1})$;
 8:   end for
 9:   $\widetilde{x}^s = \frac{1}{m_s}\sum_{t=1}^{m_s} x_t^s$, $\widetilde{y}^s = y_{m_s}^s$, $\omega_s = \frac{\sqrt{\omega_{s-1}^4 + 4\omega_{s-1}^2} - \omega_{s-1}^2}{2}$, $m_{s+1} = \min(\lfloor \rho m_s \rfloor, m)$;
10: end for
```

Output: $\widetilde{x}^S$.

It is clear that the condition (7) allows the stochastic momentum parameter to decrease, but not too fast, similar to the requirement on the step-size η_t in classical SGD. Unlike deterministic acceleration methods, where ω_s is only required to satisfy the first inequality in (7), the momentum parameter ω_s in Algorithm 2 must satisfy both inequalities. Inspired by the momentum acceleration techniques in (Tseng, 2010; Su et al., 2014) for deterministic optimization, the update rule for ω_s is defined as follows: $\omega_0 = 1 - \widetilde{L}\eta/(1-\widetilde{L}\eta)$, and for any $s > 1$, $\omega_s = \frac{\sqrt{\omega_{s-1}^4 + 4\omega_{s-1}^2} - \omega_{s-1}^2}{2}$.

4. Convergence Analysis

In this section, we provide the convergence analysis of ASVRG for both strongly convex and non-strongly convex objectives. We first give the following key intermediate result (the proofs to all theoretical results in this paper are given in the supplementary material).

Lemma 2 *Suppose Assumption 1 holds. Let $x^\star$ be an optimal solution of Problem (1), and $\{(x_t^s, y_t^s)\}$ be the sequence generated by Algorithms 1 and 2 with $\omega_s \leq 1 - \frac{\widetilde{L}\eta}{1-\widetilde{L}\eta}$[5]. Then for all $s = 1, \ldots, S$,*

$$\mathbb{E}[F(\widetilde{x}^s) - F(x^\star)] \leq (1-\omega_{s-1})\mathbb{E}\big[F(\widetilde{x}^{s-1}) - F(x^\star)\big] + \frac{\omega_{s-1}^2}{2m\eta}\mathbb{E}\big[\|x^\star - y_0^s\|^2 - \|x^\star - y_m^s\|^2\big].$$

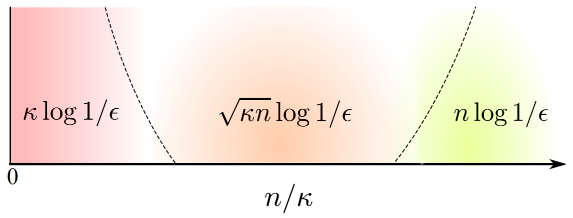
4.1. Analysis for Strongly Convex Objectives

For strongly convex objectives, our first main result is the following theorem, which gives the convergence rate and oracle complexity of Algorithm 1.

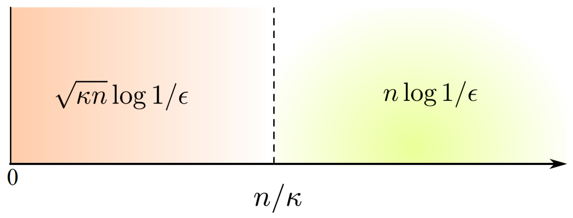
Theorem 3 *Suppose Assumptions 1 and 2 hold, and given the same notation as in Lemma 2, and m is sufficiently large so that*

$$\rho := 1 - \omega + \frac{\omega^2}{m\mu\eta} < 1.$$

5. Note that the momentum parameter ω_s in Algorithm 1 is a constant, that is, $\omega_s \equiv \omega$. In addition, if the length of the early epochs is not sufficiently large, the epochs can be viewed as an initialization step



(*a*) Corollary 4



(*b*) Corollary 5

Figure 1: Comparison of the oracle complexities of Algorithm 1 with Option I and Option II.

Then Algorithm 1 with Option I has the following geometric convergence in expectation:

$$\mathbb{E}[F(\widetilde{x}^s) - F(x^\star)] \leq \rho^s\left[F(\widetilde{x}^0) - F(x^\star)\right].$$

Theorem 3 shows that Algorithm 1 with Option I achieves linear convergence for strongly convex problems. We can easily obtain a similar result for Algorithm 1 with Option II. The following results give the oracle complexities of Algorithm 1 with Option I or Option II, as shown in Figure 1, where $\kappa = \widetilde{L}/\mu$ is the condition number.

Corollary 4 *The oracle complexity of Algorithm 1 with Option I to achieve an ε-suboptimal solution (i.e., $\mathbb{E}[F(\widetilde{x}^s)] - F(x^\star) \leq \varepsilon$) is*

$$\begin{cases} \mathcal{O}\big(\sqrt{n\widetilde{L}/\mu}\log\frac{F(\tilde{x}^0)-F(x^\star)}{\varepsilon}\big), & \textit{if } m\mu/\widetilde{L}\in[0.68623, 145.72],\\ \mathcal{O}\big((n+\widetilde{L}/\mu)\log\frac{F(\tilde{x}^0)-F(x^\star)}{\varepsilon}\big), & \textit{otherwise}. \end{cases}$$

Corollary 5 *The oracle complexity of Algorithm 1 with Option II and restarts every $\mathcal{S} = 2\cdot\big(\frac{1-\omega}{\omega} + \frac{\omega}{\eta m\mu}\big)$ epochs*[6] *to achieve an ε-suboptimal solution (i.e., $\mathbb{E}[F(\widetilde{x}^s)] - F(x^\star) \leq \varepsilon$) is*

$$\mathcal{O}\big((n+\sqrt{n\widetilde{L}/\mu})\log\frac{F(\tilde{x}^0) - F(x^\star)}{\varepsilon}\big).$$

For the most commonly used uniform random sampling (i.e., sampling probabilities $p_i = 1/n$ for all $i = 1, \ldots, n$), the oracle complexity of ASVRG becomes $\mathcal{O}((n+\sqrt{nL/\mu})\log(1/\varepsilon))$, which is identical to that of Katyusha (Allen-Zhu, 2018) and Point-SAGA (Defazio, 2016), and better than those of non-accelerated methods (e.g., $\mathcal{O}((n+L/\mu)\log(1/\varepsilon))$ of SVRG).

Remark 6 *For uniform random sampling, and recalling $L = L_{\max} = \max_j L_j$, the above oracle bound is rewritten as: $\mathcal{O}((n+\sqrt{nL_{\max}/\mu})\log(1/\varepsilon))$. As each $f_i(\cdot)$ generally has different degrees of smoothness L_i, picking the random index i_t from a non-uniform distribution is a much better choice than simple uniform random sampling (Zhao and Zhang, 2015; Needell et al., 2016). For instance, the sampling probabilities for all $f_i(\cdot)$ are proportional to their Lipschitz constants, i.e., $p_i = L_i/\sum_{j=1}^n L_j$. In this case, the oracle complexity becomes $\mathcal{O}((n+\sqrt{nL_{\text{avg}}/\mu})\log(1/\varepsilon))$, where $L_{\text{avg}} = \frac{1}{n}\sum_{j=1}^n L_j \leq L_{\max}$. In other words, the statement of Corollary 5 can be revised by simply replacing $\widetilde{L}$ with L_{avg}. And the proof only needs some minor changes accordingly. In fact, all statements in this section can be revised by replacing $\widetilde{L}$ with L_{avg}.*

6. For each restart, the new initial point is set to $\widetilde{x}^0 = \frac{1}{\mathcal{S}}\sum_{s=1}^{\mathcal{S}}\widetilde{x}^s$. If we choose the snapshot to be the weighted average as in (Allen-Zhu, 2018) rather than the uniform average, our algorithm without restarts can also achieve the tightest possible result. Note that $\mathcal{O}(n+\sqrt{nL/\mu})$ is always less than $\mathcal{O}(n+L/\mu)$.

4.2. Analysis for Non-Strongly Convex Objectives

For non-strongly convex objectives, our second main result is the following theorem, which gives the convergence rate and oracle complexity of Algorithm 2.

Theorem 7 *Suppose Assumption 1 holds. Then the following result holds,*

$$\mathbb{E}\big[F(\widetilde{x}^S) - F(x^\star)\big] \leq \frac{4(\alpha-1)[F(\widetilde{x}^0) - F(x^\star)]}{(\alpha-2)^2(S+1)^2} + \frac{2\|x^\star - \widetilde{x}^0\|^2}{\eta m(S+1)^2},$$

where $\alpha = 1/(\widetilde{L}\eta) > 2$. Furthermore, choosing $m = \Theta(n)$, Algorithm 2 achieves an ε-suboptimal solution, i.e., $\mathbb{E}[F(\widetilde{x}^S)] - F(x^\star) \leq \varepsilon$ using at most $\mathcal{O}(\frac{n\sqrt{F(\widetilde{x}^0)-F(x^\star)}}{\sqrt{\varepsilon}} + \frac{\sqrt{n\widetilde{L}}\|\widetilde{x}^0-x^\star\|}{\sqrt{\varepsilon}})$ iterations.

One can see that the oracle complexity of ASVRG is consistent with the best known result in (Hien et al., 2018; Allen-Zhu, 2018), and all the methods attain the optimal convergence rate $\mathcal{O}(1/t^2)$. Moreover, we can use the adaptive regularization technique in (Allen-Zhu and Hazan, 2016) to the original non-SC problem, and achieve a SC objective $F(x) + \frac{\sigma_s}{2}\|x - \widetilde{x}^0\|$ with a decreasing value σ_s, e.g., $\sigma_s = \sigma_{s-1}/2$ in (Allen-Zhu and Hazan, 2016). Then we have the following result.

Corollary 8 *Suppose Assumption 1 holds, and $g(\cdot)$ is non-SC. By applying the adaptive regularization technique in (Allen-Zhu and Hazan, 2016) for Algorithm 1, then we obtain an ε-suboptimal solution using at most the following oracle complexity:*

$$\mathcal{O}\left(n\log\frac{F(\widetilde{x}^0) - F(x^\star)}{\varepsilon} + \frac{\sqrt{n\widetilde{L}}\|\widetilde{x}^0 - x^\star\|}{\sqrt{\varepsilon}}\right).$$

Corollary 8 implies that ASVRG has a low oracle complexity (i.e., $\mathcal{O}(n\log(1/\varepsilon) + C_1\sqrt{n\widetilde{L}/\varepsilon})$), which is the same as that in (Allen-Zhu, 2018). Both ASVRG and Katyusha have a much faster rate than SAGA (Defazio et al., 2014a), whose oracle complexity is $\mathcal{O}((n+L)/\varepsilon)$.

Although ASVRG is much simpler than Katyusha, all the theoretical results show that ASVRG achieves the same convergence rates and oracle complexities as Katyusha for both SC and non-SC cases. Similar to (Babanezhad et al., 2015; Allen-Zhu and Yuan, 2016), we can reduce the number of gradient calculations in early iterations to further speed up ASVRG in practice.

5. Extensions of ASVRG

In this section, we first extend ASVRG to the mini-batch setting. Then we extend Algorithm 1 and its convergence results to the non-smooth setting (e.g., the problems in Cases 3 and 4).

5.1. Mini-Batch

In this part, we extend ASVRG and its convergence results to the mini-batch setting. Suppose that the mini-batch size is b, the stochastic gradient estimator with variance reduction becomes

$$\widetilde{\nabla} f_{I_t}(x^s_{t-1}) = \frac{1}{b}\sum_{i\in I_t}\frac{1}{np_i}\big[\nabla f_i(x^s_{t-1}) - \nabla f_i(\widetilde{x}^{s-1})\big] + \nabla f(\widetilde{x}^{s-1}),$$

where $I_t \subset [n]$ is a mini-batch of size b. Consequently, the momentum parameters ω and ω_s must satisfy $\omega \leq 1-\frac{\tau(b)\widetilde{L}\eta}{1-\widetilde{L}\eta}$ and $\omega_s \leq 1-\frac{\tau(b)\widetilde{L}\eta}{1-\widetilde{L}\eta}$ for SC and non-SC cases, respectively, where $\tau(b) = (n-b)/[b(n-1)]$. Moreover, the upper bound on the variance of $\widetilde{\nabla} f_{i_t}(x^s_{t-1})$ can be extended to the mini-batch setting as follows.

Lemma 9

$$\mathbb{E}\left[\left\|\widetilde{\nabla} f_{I_t}(x^s_{t-1}) - \nabla f(x^s_{t-1})\right\|^2\right] \leq \frac{2\widetilde{L}(n-b)}{b(n-1)}\left[f(\widetilde{x}^{s-1})-f(x^s_{t-1})+\nabla f(x^s_{t-1})^T(x^s_{t-1}-\widetilde{x}^{s-1})\right].$$

In the SC case, the convergence result of the mini-batch variant[7] of ASVRG is identical to Theorem 3 and Corollary 5. For the non-SC case, we set the initial parameter $\omega_0 = 1-\frac{\tau(b)\widetilde{L}\eta}{1-\widetilde{L}\eta}$. Then Theorem 7 can be extended to the mini-batch setting as follows.

Theorem 10 *Suppose Assumption 1 holds, and given the same notation as in Theorem 7 and* $\omega_0 = 1-\frac{\tau(b)\widetilde{L}\eta}{1-\widetilde{L}\eta}$*, then the following inequality holds*

$$\mathbb{E}[F(\widetilde{x}^s) - F(x^\star)] \leq \frac{4(\alpha-1)\tau(b)[F(\widetilde{x}^0)-F(x^\star)]}{[\alpha-1-\tau(b)]^2(s+1)^2} + \frac{2\widetilde{L}\alpha\|x^\star-\widetilde{x}^0\|^2}{m(s+1)^2}. \tag{8}$$

Remark 11 *For the special case of $b=1$, we have $\tau(1)=1$, and then Theorem 10 degenerates to Theorem 7. When $b=n$ (i.e., the batch setting), then $\tau(n)=0$, and the first term on the right-hand side of (8) diminishes. Then our algorithm degenerates to an accelerated deterministic method with the optimal convergence rate of $\mathcal{O}(1/t^2)$, where t is the number of iterations.*

5.2. Non-Smooth Settings

In addition to the application of the regularization reduction technique in (Allen-Zhu and Hazan, 2016) for a class of smooth and non-SC problems (i.e., Case 2 of Problem (1)), as shown in Section 4.2, ASVRG can also be extended to solve the problems in both Cases 3 and 4, when each component of f is G-Lipschitz continuous, which is defined as follows.

Definition 12 *The function $f_i(x)$ is G-Lipschitz continuous if there exists a constant G such that, for any $x, y \in \mathbb{R}^d$, $|f_i(x) - f_i(y)| \leq G\|x-y\|$.*

The key technique is using a proximal operator to obtain gradients of the δ_s-Moreau envelope of a non-smooth function $f_i(\cdot)$, defined as

$$f_i^{\delta_s}(x) = \inf_{y\in\mathbb{R}^d} f_i(y) + \frac{\delta_s}{2}\|x-y\|^2, \tag{9}$$

where δ_s is an increasing parameter as in (Allen-Zhu and Hazan, 2016). That is, we use the proximal operator to smoothen each component function, and optimize the new, smooth function which approximates the original problem. This technique has been used in Katyusha (Allen-Zhu, 2018) and accelerated SDCA (Shalev-Shwartz and Zhang, 2016) for non-smooth objectives.

7. Note that in the mini-batch setting, the number of stochastic iterations of the inner loop in Algorithms 1 and 2 is reduced from m to $\lfloor m/b \rfloor$.

Property 1 (Nesterov (2005); Bauschke and Combettes (2011); Orabona et al. (2012)) *Let each $f_i(x)$ be convex and G-Lipschitz continuous. For any $\delta > 0$, the following results hold:*
(a) $f_i^{\delta}(x)$ is δ-smooth;
(b) $\nabla(f_i^{\delta})(x) = \delta\Big(x - \text{prox}_{\delta}^{f_i}(x)\Big)$;
(c) $f_i^{\delta}(x) \leq f_i(x) \leq f_i^{\delta}(x) + \frac{G^2}{2\delta}$.

By using the similar techniques in (Allen-Zhu and Hazan, 2016), we can apply ASVRG to solve the smooth problem $F^{\delta_s}(x) := \frac{1}{n}\sum_{i=1}^{n} f_i^{\delta_s}(x) + g(x)$. It is easy to verify that ASVRG satisfies the homogenous objective decrease (HOOD) property in (Allen-Zhu and Hazan, 2016) (see (Allen-Zhu and Hazan, 2016) for the detail of HOOD), as shown below.

Corollary 13 *Algorithm 1 used to solve the problem in Case 1 satisfies the HOOD property with at most $\mathcal{O}(n+\sqrt{n\widetilde{L}/\mu})$ iterations. That is, for every starting point $\widetilde{x}^0$, Algorithm 1 produces an output $\widetilde{x}^s$ satisfying $\mathbb{E}[F(\widetilde{x}^s)] - F(x^\star) \leq [F(\widetilde{x}^0) - F(x^\star)]/4$ in at most $\mathcal{O}(n+\sqrt{n\widetilde{L}/\mu})$ iterations.*

In the following, we extend the result in Theorem 3 to the non-smooth setting as follows.

Corollary 14 *Let $f_i(x)$ be G-Lpischitz continuous, and $g(x)$ be μ-strongly convex. By applying the adaptive smooth technique in (Allen-Zhu and Hazan, 2016) on ASVRG, we obtain an ε-suboptimal solution using at most the following oracle complexity:*

$$\mathcal{O}\left(n \log \frac{F(\widetilde{x}^0) - F(x^\star)}{\varepsilon} + \frac{\sqrt{n}G}{\sqrt{\mu\varepsilon}}\right).$$

Corollary 15 *Let $f_i(x)$ be G-Lpischitz continuous and $g(x)$ be not necessarily strongly convex. By applying both the adaptive regularization and smooth techniques in (Allen-Zhu and Hazan, 2016) on ASVRG, we obtain an ε-suboptimal solution using at most the following oracle complexity:*

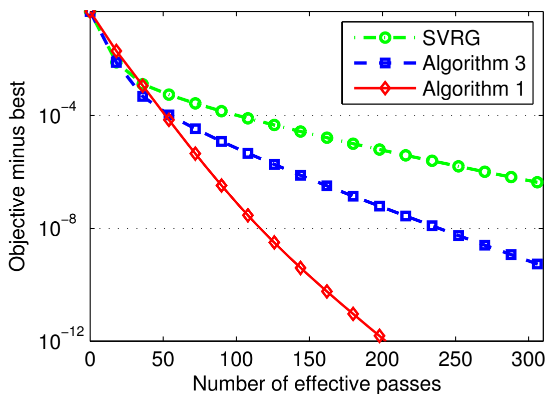
$$\mathcal{O}\left(n \log \frac{F(\widetilde{x}^0) - F(x^\star)}{\varepsilon} + \frac{\sqrt{n}G\|\widetilde{x}^0 - x^\star\|}{\varepsilon}\right).$$

From Corollaries 14 and 15, one can see that ASVRG converges to an ε-accurate solution for Case 3 of Problem (1) in $\mathcal{O}\Big(n \log \frac{C_2}{\varepsilon} + \frac{\sqrt{n}G}{\sqrt{\mu\varepsilon}}\Big)$ iterations and for Case 4 in $\mathcal{O}\Big(n \log \frac{C_2}{\varepsilon} + \frac{\sqrt{n}C_1 G}{\varepsilon}\Big)$ iterations. That is, ASVRG achieves the same low oracle complexities as the accelerated stochastic variance reduction method, Katyusha, for the two classes of non-smooth problems (i.e., Cases 3 and 4 of Problem (1)).

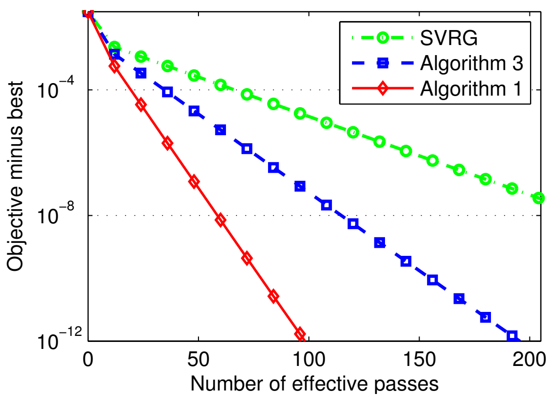
6. Experiments

In this section, we evaluate the performance of ASVRG, and all the experiments were performed on a PC with an Intel i5-2400 CPU and 16GB RAM. We used the two publicly available data sets in our experiments: Covtype and RCV1, which can be downloaded from the LIBSVM Data website[8].

8. https://www.csie.ntu.edu.tw/~cjlin/libsvm/



(a) RCV1, $\lambda = 10^{-6}$



(b) Covtype, $\lambda = 10^{-7}$

Figure 2: Comparison of SVRG Johnson and Zhang (2013), ASVRG without momentum (i.e., Algorithm 3), and ASVRG (i.e., Algorithm 1) for ridge regression.

6.1. Effectiveness of Our Momentum

Figure 2 shows the performance of ASVRG with $\theta_s \equiv 1$ (i.e., Algorithm 3) and ASVRG in order to illustrate the importance and effectiveness of our momentum. Note that the epoch length is set to $m = 2n$ for the two algorithms (i.e., $m_1 = 2n$ and $\rho = 1$), as well as SVRG (Johnson and Zhang, 2013). It is clear that Algorithm 3 is without our momentum acceleration technique. The main difference between Algorithm 3 and SVRG is that the snapshot and starting points of the former are set to the uniform average and last iterate of the previous epoch, respectively, while the two points of the later are the last iterate. The results show that Algorithm 3 outperforms SVRG, suggesting that the iterate average can work better in practice, as discussed in (Shang et al., 2018b). In particular, ASVRG converges significantly faster than ASVRG without momentum and SVRG, meaning that our momentum acceleration technique can accelerate the convergence of ASVRG.

6.2. Comparison with Stochastic Methods

For fair comparison, ASVRG and the compared algorithms (including SVRG (Johnson and Zhang, 2013), SAGA (Defazio et al., 2014a), Acc-Prox-SVRG (Nitanda, 2014), Catalyst (Lin et al., 2015), and Katyusha (Allen-Zhu, 2018)) were implemented in C++ with a Matlab interface. There is only one parameter (i.e., the learning rate) to tune for all these methods except Catalyst and Acc-Prox-SVRG. In particular, we compare their performance in terms of both the number of effective passes over the data and running time (seconds). As in (Xiao and Zhang, 2014), each feature vector has been normalized so that it has norm 1.

Figure 3 shows how the objective gap (i.e., $F(x^s) - F(x^\star)$) of all these methods decreases on elastic net regularized logistic regression ($\lambda_1 = 10^{-4}$, $\lambda_2 = 10^{-5}$) as time goes on. It is clear that ASVRG converges significantly faster than the other methods in terms of both oracle calls and running time, while Catalyst and Katyusha achieve comparable and sometimes even better performance than SVRG and SAGA in terms of the running time (seconds). The main reason is that ASVRG not only takes advantage of the momentum acceleration trick, but also can use much larger step-size (e.g., $1/(3L)$ for ASVRG vs. $1/(10L)$ for SVRG). This empirically verifies our theoretical result in Corollary 5 that ASVRG has the same low oracle complexity as Katyusha.

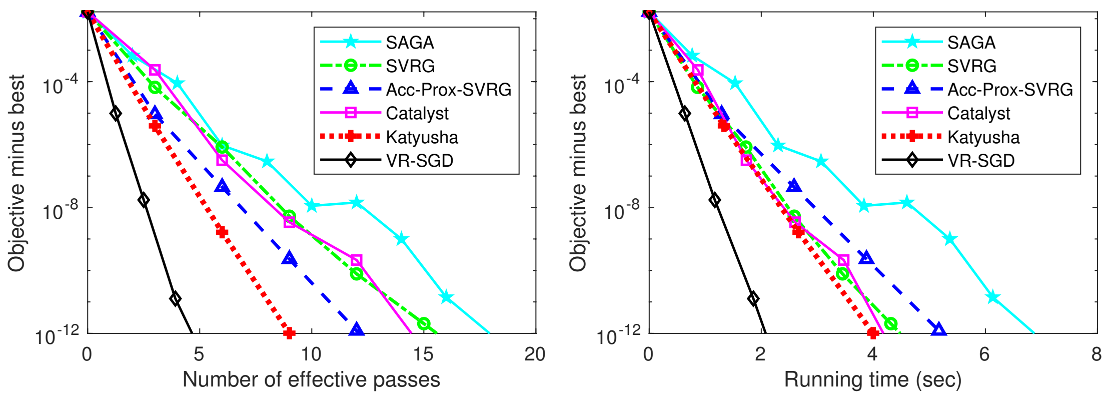


Figure 3: Comparison of SAGA (Defazio et al., 2014a), SVRG (Johnson and Zhang, 2013), Acc-Prox-SVRG (Nitanda, 2014), Catalyst (Lin et al., 2015), Katyusha (Allen-Zhu, 2018) and ASVRG on Covtype.

ASVRG significantly outperforms Katyusha in terms of running time, which implies that ASVRG has a much lower per-iteration cost than Katyusha.

7. Conclusions

We proposed an efficient ASVRG method, which integrates both the momentum acceleration trick and variance reduction technique. We first designed a simple momentum acceleration technique. Then we theoretically analyzed the convergence properties of ASVRG, which show that ASVRG achieves the same low oracle complexities for both SC and non-SC objectives as accelerated methods, e.g., Katyusha (Allen-Zhu, 2018). Moreover, we also extended ASVRG and its convergence results to both mini-batch settings and non-smooth settings.

It would be interesting to consider other classes of settings, e.g., the non-Euclidean norm setting. In practice, ASVRG is much simpler than the existing accelerated methods, and usually converges much faster than them, which has been verified in our experiments. Due to its simplicity, it is more friendly to asynchronous parallel and distributed implementation for large-scale machine learning problems (Zhou et al., 2018), similar to (Reddi et al., 2015; Sra et al., 2016; Mania et al., 2017; Zhou et al., 2018; Lee et al., 2017; Wang et al., 2017). One natural open problem is whether the best oracle complexities can be obtained by ASVRG in the asynchronous and distributed settings.

Acknowledgments

We thank the reviewers for their valuable comments. This work was supported in part by Grants (CUHK 14206715 & 14222816) from the Hong Kong RGC, Project supported the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61621005), the Major Research Plan of the National Natural Science Foundation of China (Nos. 91438201 and 91438103), Project supported the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61621005), the National Natural Science Foundation of China (Nos. 61876220, 61876221, 61836009, U1701267, 61871310, 61573267, 61502369 and 61473215), the Program for Cheung Kong Scholars and Innovative Research Team in University

(No. IRT_15R53), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048), and the Science Foundation of Xidian University (No. 10251180018).

References

Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *J. Mach. Learn. Res.*, 18:1–51, 2018.

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

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

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.

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics, Springer, 2011.

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

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

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, 2014a.

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

N. Flammarion and F. Bach. From averaging to acceleration, there is only a step-size. In *COLT*, pages 658–695, 2015.

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.

D. Garber, E. Hazan, C. Jin, S. M. Kakade, C. Musco, P. Netrapalli, and A. Sidford. Faster eigenvector computation via shift-and-invert preconditioning. In *ICML*, pages 2626–2634, 2016.

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

L. T. K. Hien, C. Lu, H. Xu, and J. Feng. Accelerated stochastic mirror descent algorithms for composite non-strongly convex optimization. *arXiv:1605.06892v5*, 2018.

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

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

G. Lan and Y. Zhou. An optimal randomized incremental gradient method. *Math. Program.*, 171: 167–215, 2018.

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.

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

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

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

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.

T. Murata and T. Suzuki. Doubly accelerated stochastic variance reduced dual averaging method for regularized empirical risk minimization. In *NIPS*, pages 608–617, 2017.

D. Needell, N. Srebro, and R. Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Math. Program.*, 155:549–573, 2016.

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

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

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

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

F. Orabona, A. Argyriou, and N. Srebro. PRISMA: Proximal iterative smoothing algorithm. *arXiv:1206.2372v2*, 2012.

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

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.

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.

S. Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747v2*, 2017.

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. Technical report, INRIA, Paris, 2013.

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

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

O. Shamir. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *ICML*, pages 144–152, 2015.

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

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

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

F. Shang, Y. Liu, K. Zhou, J. Cheng, K. W. Ng, and Y. Yoshida. Guaranteed sufficient decrease for stochastic variance reduced gradient optimization. In *AISTATS*, pages 1027–1036, 2018a.

F. Shang, K. Zhou, H. Liu, J. Cheng, I. W. Tsang, L. Zhang, D. Tao, and L. Jiao. VR-SGD: A simple stochastic variance reduction method for machine learning. *arXiv:1802.09932*, 2018b.

S. Sra, A. W. Yu, M. Li, and A. J. Smola. AdaDelay: Delay adaptive distributed stochastic optimization. In *AISTATS*, pages 957–965, 2016.

W. Su, S. P. Boyd, and E. J. Candes. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. In *NIPS*, pages 2510–2518, 2014.

P. Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Math. Program.*, 125:263–295, 2010.

J. Wang, W. Wang, and N. Srebro. Memory and communication efficient distributed stochastic optimization with minibatch-prox. In *COLT*, pages 1882–1919, 2017.

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

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

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

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

K. Zhou, F. Shang, and J. Cheng. A simple stochastic variance reduced algorithm with fast convergence rates. In *ICML*, pages 5975–5984, 2018.