Accelerated Incremental Gradient Descent using Momentum Acceleration with Scaling Factor

Yuanyuan Liu[1,2], **Fanhua Shang**[1,2*] and **Licheng Jiao**[1,2]

[1]Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education
[2]School of Artificial Intelligence, Xidian University, China
{yyliu, fhshang}@xidian.edu.cn, lchjiao@mail.xidian.edu.cn

Abstract

Recently, research on variance reduced incremental gradient descent methods (e.g., SAGA) has made exciting progress (e.g., linear convergence for strongly convex (SC) problems). However, existing accelerated methods (e.g., point-SAGA) suffer from drawbacks such as inflexibility. In this paper, we design a novel and simple momentum to accelerate the classical SAGA algorithm, and propose a direct accelerated incremental gradient descent algorithm. In particular, our theoretical result shows that our algorithm attains a best-known oracle complexity for SC minimization problems and an improved convergence rate for the case of $n \geq L/\mu$. We also give experimental results justifying our theoretical results and showing the effectiveness of our algorithm.

1 Introduction

Recently, stochastic/incremental first-order methods have received extensive attention due to their low per-iteration cost and the ability to handle large-scale problems including unconstrained/constrained composite convex minimization [Allen-Zhu, 2018; Liu *et al.*, 2017; Shang *et al.*, 2018a]. In particular, the research on stochastic variance reduced gradient descent methods (e.g., SAG [Roux *et al.*, 2012], SVRG [Johnson and Zhang, 2013], SDCA [Shalev-Shwartz and Zhang, 2013], SAGA [Defazio *et al.*, 2014a]) and their proximal variants (e.g., Prox-SVRG [Xiao and Zhang, 2014] and VR-SGD [Shang *et al.*, 2018a]) has made exciting progress, e.g., linear convergence for strongly convex (SC) problems. These methods use past gradients to progressively reduce the variance of stochastic gradient estimators, which leads to a revolution in the area of first-order optimization [Shang *et al.*, 2019]. In this paper, we consider the following composite convex minimization problem in many problems of machine learning, statistics, and operations research, such as regularized empirical risk minimization (ERM).

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

where $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ is the finite average of n convex component functions $f_i(x) : \mathbb{R}^d \to \mathbb{R}$, and $h(x)$ is a simple but possibly non-smooth convex function.

For solving SC problems of the formulation (1), the oracle complexity (total number of gradient evaluations to find an ϵ-suboptimal solution) of the stochastic variance reduction methods mentioned above (including SVRG and SAGA) is $\mathcal{O}((n+\kappa)\log(1/\epsilon))$, where κ is the condition number of Problem (1), and n is the number of samples, while the oracle complexity of accelerated deterministic methods, e.g., FISTA [Beck and Teboulle, 2009], is $\mathcal{O}(n\sqrt{\kappa}\log(1/\epsilon))$. Obviously, the oracle complexities show that those stochastic variance reduction methods always converge faster than deterministic methods (including their accelerated variants) as long as $\kappa \leq \mathcal{O}(n^2)$. However, there is still a gap between the complexity of those stochastic variance reduction methods and the upper bound provided in [Woodworth and Srebro, 2016].

More recently, there is a surge of interests in accelerating stochastic variance reduction gradient optimization. The acceleration techniques in accelerated methods mainly include the Nesterov's acceleration technique [Nitanda, 2014; Frostig *et al.*, 2015; Lin *et al.*, 2015; Murata and Suzuki, 2017], the choice of growing epoch length [Mahdavi *et al.*, 2013; Allen-Zhu and Yuan, 2016; Shang *et al.*, 2017], and the momentum acceleration tricks [Shang *et al.*, 2018b; Allen-Zhu, 2018; Zhou *et al.*, 2018; Hien *et al.*, 2019]. [Lin *et al.*, 2015] proposed a Catalyst framework for accelerating some stochastic variance reduction algorithms (including SVRG and SAGA) and proved that their accelerated variants achieve an oracle complexity of $\mathcal{O}((n+\sqrt{n\kappa})\log(\kappa)\log(1/\epsilon))$ for SC problems. In particular, as the accelerated variants of SVRG, Katyusha [Allen-Zhu, 2018] and MiG [Zhou *et al.*, 2018] achieve the best-known oracle complexity of $\mathcal{O}((n+\sqrt{n\kappa})\log(1/\epsilon))$ for SC problems, which is identical to the upper bound in [Woodworth and Srebro, 2016].

As an accelerated variant of SAGA, point-SAGA [Defazio, 2016] requires a proximal operator oracle of each f_i and can attain the same oracle complexity as Katyusha. However, the proximal operator oracle for point-SAGA may not be efficiently calculated in practice. As we all know that a large amount of work has been done for accelerating SVRG, while the notable incremental gradient method, SAGA, does not have a direct accelerated variant until recently, except for SSNM [Zhou *et al.*, 2019], which can also obtain the best-known

*Corresponding author

oracle complexity. However, the memory complexity of SSNM is always $O(nd)$. Therefore, we will propose a simple and direct accelerated variant for SAGA.

Contributions: Existing accelerated methods mentioned above can attain the theoretical upper complexity bounds provided in [Woodworth and Srebro, 2016]. We ask the following question in this paper: *Using only gradient information, can we further improve the convergence rates of those methods such as Katyusha [Allen-Zhu, 2018], point-SAGA [Defazio, 2016] and SSNM [Zhou* et al.*, 2019]*? In this paper, we propose a novel accelerated incremental gradient descent (AIGD) algorithm to push towards the convergence rates.

- We design a general momentum acceleration scheme for the direct acceleration of SAGA, in which we introduce a novel momentum to replace the Nesterov's momentum and Katyusha momentum used in [Allen-Zhu, 2018].
- We prove that AIGD achieves a linear convergence rate and the oracle complexity of $\mathcal{O}((n+\sqrt{n\kappa})\log(1/\epsilon))$ for strongly convex problems, which is identical to the convergence results of existing accelerated algorithms such as Katyusha [Allen-Zhu, 2018], point-SAGA [Defazio, 2016], and SSNM [Zhou *et al.*, 2019].
- In particular, our convergence results also show that AIGD can slightly improve the convergence rates of Katyusha, point-SAGA and SSNM for the case of $n \geq L/\mu$, as shown in Table 1. It means that this study can partly answer the above-mentioned question.
- We also discuss some subtle differences between AIGD and existing accelerated incremental algorithms such as point-SAGA and SSNM, which imply that AIGD is more suitable for solving various large-scale machine learning problems.
- AIGD can also be extended to the non-convex setting. Our experimental results further verify that AIGD is usually faster than the state-of-the-art accelerated stochastic/incremental methods.

2 Preliminaries and Notations

Throughout this paper, the norm $\|\cdot\|$ is the standard Euclidean norm. We denote by $\nabla f(x)$ the full gradient of $f(x)$ if it is differentiable, or $\partial f(x)$ a sub-gradient of $f(x)$ if $f(x)$ is only Lipschitz continuous. We mostly focus on the case of Problem (1) when each $f_i(x)$ is L-smooth.

Assumption 1. *Each $f_i(\cdot)$ is L-smooth, i.e., there exists a constant $L>0$ such that for all $x, y \in \mathbb{R}^d$,*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|.$$

Assumption 2. *The function $h(\cdot)$ is μ-strongly convex, i.e., there exists $\mu>0$ such that for all $x, y \in \mathbb{R}^d$,*

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

where $\partial h(x)$ is the subdifferential of $h(\cdot)$ at x.

Recently, there emerges a stream of studies on stochastic variance reduced methods, such as [Zhang *et al.*, 2013; Johnson and Zhang, 2013; Xiao and Zhang, 2014; Defazio *et al.*, 2014a; Shang *et al.*, 2019]. The two most popular choices for stochastic gradient estimators are the SVRG estimator in [Zhang *et al.*, 2013; Johnson and Zhang, 2013] and the SAGA estimator in [Defazio *et al.*, 2014a]. The main update steps of SAGA [Defazio *et al.*, 2014a] are

$$w_k = x_{k-1} - \eta\left[\nabla f_i(x_{k-1}) - \nabla f_i(\phi_i^{k-1}) + \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(\phi_j^{k-1})\right],$$

$$x_k = \text{prox}_\gamma^h(w_k),$$

where $\eta>0$ is a learning rate, and $\text{prox}_\gamma^h(\cdot)$ is a proximal operator. More details can be found in [Defazio *et al.*, 2014a]. We start with an initial vector $x_0 \in \mathbb{R}^d$ and the known derivatives $\nabla f_j(\phi_j^0)$ with $\phi_j^0 = x_0$ for each $j \in \{1, 2, \ldots, n\}$. These derivatives are stored in a table data structure of length n. At the k-th iteration, we choose randomly sample i, and take $\phi_i^k = x_k$ and store the gradient $\nabla f_i(\phi_i^k)$ in the table, and all other entries in the table remain unchanged.

Algorithms	Convergence rates	Memory	Direct
Katyusha	$\mathcal{O}(\rho_1^K)/\mathcal{O}(\rho_2^K)$	$O(n)$	Yes
point-SAGA	$\mathcal{O}(\rho_1^K)/\mathcal{O}(\rho_2^K)$	$O(nd)$ or $O(n)$	No
SSNM	$\mathcal{O}(\rho_1^K)/\mathcal{O}(\rho_2^K)$	Always $O(nd)$	Yes
AIGD	$\mathcal{O}(\rho_3^K)/\mathcal{O}(\rho_2^K)$	$O(nd)$ or $O(n)$	Yes

Table 1: Comparison of some accelerated stochastic methods. Note that $\rho_1 = 1-1/(2n)$ with $\kappa := L/\mu \leq 4n/3$, $\rho_2 = 1-\sqrt{1/(3\kappa n)}$ with $\kappa > 4n/3$, and $\rho_3 = 1-3/4n$ with $\kappa \leq 4n/3$.

Algorithm 1 AIGD for Strongly Convex Objectives

Input: The number of iterations K.
Initialize: $\phi_1^0 = \phi_2^0 = \ldots = \phi_n^0 = x_0$, $z_0 = x_0$, $\theta = \frac{1}{L\alpha\eta}$. $\alpha = 8n/\beta$ if $k <= K-1$; otherwise, $\alpha = 5$. We set $\eta = \frac{\sqrt{3}}{\sqrt{L\mu n}}$ if $n < \frac{3L}{4\mu}$; otherwise, $\eta = \frac{3}{4n\mu}$.

1: **for** $k = 1, 2, \ldots, K$ **do**
2: Pick i_k uniformly at random from $\{1, 2, \ldots, n\}$, and update y_k by (2);
3: $\widetilde{\nabla} f_{i_k}(y_k) = \nabla f_{i_k}(y_k) - \nabla f_{i_k}(\phi_{i_k}^{k-1}) + \frac{1}{n}\sum_{j=1}^{n}\nabla f_j(\phi_j^{k-1})$;
4: Update z_k by solving (4);
5: Update x_k by (3);
6: Take $\phi_{i_k}^k = x_k$, calculate and store $\nabla f_{i_k}(\phi_{i_k}^k)$ in the table;
7: **end for**

Output: $\overline{x} = x_K$.

3 Accelerated Incremental Gradient Descent

In section, we propose a simple and direct accelerated incremental gradient descent (AIGD) algorithm.

3.1 A Novel Momentum with Scaling Factor

In this subsection, we design a novel momentum acceleration scheme for incremental gradient descent optimization, as shown in Algorithm 1. In particular, a scaling factor is introduced into our momentum acceleration scheme to improve

the convergence rate. For solving Problem (1), our main update rules with momentum acceleration are

$$y_k = \theta z_{k-1} + (1-\beta\theta)x_{k-1}, \tag{2}$$
$$x_k = y_k + \theta(z_k - z_{k-1}), \tag{3}$$

where θ is the momentum parameter, β is a scaling factor using which we can improve the convergence rate, and z_k is the solution to the following problem:

$$z_k = \arg\min_z \{\beta\theta h(z/\beta) + \theta\langle\widetilde{\nabla} f_{i_k}(y_k), z - z_{k-1}\rangle + \frac{\theta^2}{2\eta}\|z - z_{k-1}\|^2\}, \tag{4}$$

where $\widetilde{\nabla} f_i(y_k) := \nabla f_i(y_k) - \nabla f_i(\phi_i^{k-1}) + \frac{1}{n}\sum_{j=1}^n \nabla f_j(\phi_j^{k-1})$. Note that in our algorithm, we also use $h(z/\beta)$ instead of $h(z)$ used in SNNM. The parameters θ, η and β are set in Algorithm 1.

Remark 1. *There is an important difference between our method and SSNM. That is, we use $y_k = \theta z_{k-1} + (1-\beta\theta)x_{k-1}$ instead of $y_k = \theta z_{k-1} + (1-\theta)\phi_i^{k-1}$ used in SSNM. Our setting has the following two advantages: We only need to store the gradient table, while SSNM requires to store both the "point" and gradient tables. The second advantage is that our method removes the independent sample assumption used in SSNM. This means that our algorithm has a much weaker convergence condition than SSNM.*

3.2 Comparison with SAGA, Point-SAGA and SSNM

- This paper uses $F(x) - F(x^\star)$ as a convergence critical rule instead of the Lyapunov function used in SAGA, point-SAGA and SSNM, making our algorithm easier to extend for solving structure optimization problems, such as graph-guided fuzed Lasso [Kim *et al.*, 2009] and generalized Lasso [Tibshirani and Taylor, 2011].
- In particular, we design a novel momentum scheme for accelerating SAGA instead of that in SSNM, such that our algorithm only needs to store the gradient table, while SSNM requires to store both the "point" and gradient tables. Therefore, the memory complexity of SSNM is always $O(nd)$, as analyzed in [Zhou *et al.*, 2019]. This is a disadvantage of SSNM when the objective is a linear model, e.g., linear logistic regression and ridge regression. In contrast, our algorithm only requires an $O(d)$ memory complexity to simply store a scalar to represent the gradient of each component function.
- Our AIGD method is a direct accelerated incremental gradient method, while point-SAGA [Defazio, 2016] requires the proximal operator oracle of each component function. However, the proximal operator may not be efficiently computed in practice, which makes point-SAGA not suitable for many real-world problems.
- Moreover, the update rules of our algorithm are more elegant than Katyusha and MiG, both of which require a tricky weighted averaged scheme at the end of each inner loop. In particular, our algorithm can further improve the convergence rate of Katyusha, MiG, point-SAGA and SSNM for the case of $n \geq 3\kappa/4$, where $\kappa = L/\mu$.

3.3 Extensions and Complexity Analysis

AIGD can be extended to non-smooth and Lipschitz continuous settings. By using adaptive regularization and smoothing techniques as in [Allen-Zhu and Hazan, 2016; Allen-Zhu, 2018], one can get a new and smooth function, which approximates the original function. That is, our AIGD method can attain at least the same complexity bounds as Katyusha, i.e., $\mathcal{O}(n\log(1/\epsilon) + L\sqrt{n/(\mu\epsilon)})$ for strongly convex and Lipschitz continuous problems and $\mathcal{O}(n\log(1/\epsilon) + \sqrt{n}L/\epsilon)$ for non-strongly convex and Lipstize continuous problems.

Each iteration of AIGD computes the stochastic gradients $\nabla f_{i_k}(y_k)$ and $\nabla f_{i_k}(\phi_{i_k}^{k-1})$, which is the same as existing stochastic methods such as SVRG [Johnson and Zhang, 2013] and Katyusha [Allen-Zhu, 2018]. In ERM problems, the loss function $f_i(\cdot)$ takes the form $f_i(a_i^T x)$ for a_i, where a_i is a constant vector. With such a structure, we apply the widely used scheme as in [Roux *et al.*, 2012; Defazio *et al.*, 2014b] and only need to store the scalar $a_{i_k}^T \phi_{i_k}^{k-1}$ for $\nabla f_{i_k}(\phi_{i_k}^{k-1})$ rather than the vector in each iteration. Therefore, the storage cost of AIGD can be reduced from $O(nd)$ to $O(n)$.

4 Convergence Analysis

In this section, we analyze the convergence property of our AIGD algorithm. Before giving our main convergence result, we first give and prove the following lemmas.

Lemma 1 (Variance Upper Bound)**.** *Let $x^\star$ be the optimal solution of Problem (1). Suppose each $f_i(\cdot)$ is convex and L-smooth, and let $\widetilde{\nabla}_k := \widetilde{\nabla} f_{i_k}(y_k)$, then we have*

$$\begin{aligned}&\mathbb{E}_{i_k}[\|\widetilde{\nabla}_k - \nabla f(y_k)\|^2] \\ &\leq 4L\left(f(x^\star) - f(y_k) + \langle\nabla f(y_k), y_k - x^\star\rangle\right) \\ &\quad + \frac{4L}{n}\sum_{j=1}^n \left[f_j(\phi_j^{k-1}) - f(x^\star) - \langle\nabla f_j(x^\star),\ \phi_j^{k-1} - x^\star\rangle\right].\end{aligned}$$

Proof. By the definition of $\widetilde{\nabla}_k$ and taking expectation over the random choice of i_k, we have

$$\begin{aligned}&\mathbb{E}_{i_k}\left[\left\|\widetilde{\nabla}_k - \nabla f(y_k)\right\|^2\right] \\ &= \mathbb{E}_{i_k}\left[\left\|\nabla f(y_k) - \frac{1}{n}\sum_{j=1}^n \nabla f_j(\phi_j^{k-1}) - \nabla f_{i_k}(y_k) + \nabla f_{i_k}(\phi_{i_k}^{k-1})\right\|^2\right] \\ &\leq \mathbb{E}_{i_k}\left[\left\|\nabla f_{i_k}(y_k) - \nabla f_{i_k}(\phi_{i_k}^{k-1})\right\|^2\right] \\ &\leq 2\mathbb{E}_{i_k}\left[\|\nabla f_{i_k}(y_k) - \nabla f_{i_k}(x^\star)\|^2 + \left\|\nabla f_{i_k}(x^\star) - \nabla f_{i_k}(\phi_{i_k}^{k-1})\right\|^2\right] \\ &\leq 4L[f(x^\star) - f(y_k) + \langle\nabla f(y_k),\ y_k - x^\star\rangle] \\ &\quad + \frac{2}{n}\sum_{j=1}^n \|\nabla f_j(\phi_j^{k-1}) - \nabla f_j(x^\star)\|^2 \\ &\leq 4L[f(x^\star) - f(y_k) + \langle\nabla f(y_k),\ y_k - x^\star\rangle] \\ &\quad + \frac{4L}{n}\sum_{j=1}^n \left[f_j(\phi_j^{k-1}) - f(x^\star) - \langle\nabla f_j(x^\star),\ \phi_j^{k-1} - x^\star\rangle\right],\end{aligned}$$

where the first inequality follows from the facts that $\mathbb{E}_{i_k}[\nabla f_{i_k}(\phi_{i_k}^{k-1})] = \frac{1}{n}\sum_{j=1}^n \nabla f_j(\phi_j^{k-1})$, $\mathbb{E}_{i_k}[\nabla f_{i_k}(y_k)] = \nabla f(y_k)$, $\mathbb{E}[f_{i_k}(\phi_{i_k}^{k-1})] = \frac{1}{n}\sum_{j=1}^n f_j(\phi_j^{k-1})$ and $\mathbb{E}_{i_k}[\|\mathbb{E}_{i_k}[x] - x\|^2] = \mathbb{E}_{i_k}[\|x\|^2] - \|\mathbb{E}_{i_k}[x]\|^2 \le \mathbb{E}_{i_k}[\|x\|^2]$; the third inequality holds due to $\mathbb{E}_{i_k}[\nabla f_{i_k}(y_k)] = \nabla f(y_k)$, $\mathbb{E}_{i_k}[\nabla f_{i_k}(x^\star)] = \nabla f(x^\star)$, and $\|\nabla f_j(x) - \nabla f_j(y)\|^2 \le 2L[f_j(y) - f_j(x) + \langle \nabla f_j(x),\ x - y\rangle]$ for all $x, y \in \mathbb{R}^d$. □

Lemma 2. *Suppose each $f_i(\cdot)$ is convex and L-smooth, then we have*

$$\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n \left(f_j(\phi_j^{k-1}) - f(x^\star) - \langle \nabla f_j(x^\star), \phi_j^{k-1} - x^\star\rangle\right)\right] \le \mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n F_j(\phi_j^{k-1}) - F(x^*)\right],$$

where $\mathbb{E}$ denotes the expectation with respect to all randomness, and $F_j(\cdot) = f_j(\cdot) + h(\cdot)$.

Proof.

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n \left(f_j(\phi_j^{k-1}) - f(x^\star) - \langle \nabla f_j(x^\star),\ \phi_j^{k-1} - x^\star\rangle\right)\right] \\
=&\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n \left(f_j(\phi_j^{k-1}) - f(x^\star) + \langle \partial h(x^\star),\ \phi_j^{k-1} - x^\star\rangle\right)\right] \\
&- \mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n \langle \partial h(x^\star) + \nabla f_j(x^\star),\ \phi_j^{k-1} - x^\star\rangle\right] \\
\le&\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n \left(f_j(\phi_j^{k-1}) - f(x^\star) + h(\phi_j^{k-1}) - h(x^\star)\right)\right] \\
&- \mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n \langle \partial h(x^\star) + \nabla f_j(x^\star),\ \phi_j^{k-1} - x^\star\rangle\right] \\
\le&\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n F_j(\phi_j^{k-1}) - F(x^\star)\right] \\
&- \mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n \langle \partial h(x^\star) + \nabla f_j(x^\star),\ \phi_j^{k-1} - x^\star\rangle\right],
\end{aligned}$$

where $\partial h(x^\star)$ is a subgradient of $h(\cdot)$ and $\partial h(x^\star) + \nabla f(x^\star) = 0$. Let $F_j'(x^\star) := \nabla f_j(x^\star) + \partial h(x^\star)$, and using the result in SAGA [Defazio *et al.*, 2014a], we have

$$\begin{aligned}
&\mathbb{E}_{i_{k-1}}[\frac{1}{n}\sum_{j=1}^n \langle F_j'(x^\star),\ \phi_j^{k-1} - x^\star\rangle] \\
=&\frac{1}{n}\langle F'(x^\star),\ x_{k-1} - x^\star\rangle + \frac{n-1}{n^2}\sum_{j=1}^n \langle F_j'(x^\star),\ \phi_j^{k-2} - x^\star\rangle \\
=&\frac{n-1}{n^2}\sum_{j=1}^n \langle F_j'(x^\star),\ \phi_j^{k-2} - x^\star\rangle.
\end{aligned}$$

Let $F'(x^\star) := \nabla f(x^\star) + \partial h(x^\star)$, and we have $F'(x^\star) = 0$. Thus, the following result holds:

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n \langle \partial h(x^\star) + \nabla f_j(x^\star),\ \phi_j^{k-1} - x^\star\rangle\right] \\
=&\mathbb{E}\left[\left(1 - \frac{1}{n}\right)\frac{1}{n}\sum_{j=1}^n \langle F_j'(x^\star),\ \phi_j^{k-2} - x^\star\rangle\right] \\
&\vdots \\
=&\mathbb{E}\left[\left(1 - \frac{1}{n}\right)^{k-1}\frac{1}{n}\sum_{j=1}^n \langle F_j'(x^\star),\ \phi_j^{0} - x^\star\rangle\right] \\
=&\mathbb{E}\left[\left(1 - \frac{1}{n}\right)^{k-1}\langle F'(x^\star),\ x_0 - x^\star\rangle\right] \\
=&0.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n \left(f_j(\phi_j^{k-1}) - f(x^\star) - \langle \nabla f_j(x^\star),\ \phi_j^{k-1} - x^\star\rangle\right)\right] \\
\le&\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^n F_j(\phi_j^{k-1}) - F(x^\star)\right].
\end{aligned}$$

This completes the proof. □

Theorem 1. *Suppose $h(\cdot)$ is μ-strongly convex, and let $\{(x_k, y_k, z_k)\}$ be the sequence generated by Algorithm 1. Then Algorithm 1 has the following geometric convergence in expectation:*

$$\mathbb{E}\left[F(x_K) - F(x^\star)\right] \le O(\rho^K)\left[F(x_0) - F(x^\star)\right],$$

where

$$\rho = \begin{cases} \dfrac{1}{1 + \sqrt{\frac{\mu}{3Ln}}}, & \text{if } n < \dfrac{3L}{4\mu}, \\ 1 - \dfrac{3}{4n}, & \text{if } n \ge \dfrac{3L}{4\mu}. \end{cases}$$

That is, Algorithm 1 achieves an ϵ-suboptimal solution using at most

$$\mathcal{O}\left((n + \sqrt{n\kappa})\log\frac{F(x_0) - F(x^\star)}{\epsilon}\right) \text{ iterations.}$$

Remark 2. *Theorem 1 shows that AIGD achieves the oracle complexity of $\mathcal{O}\left((n + \sqrt{n\kappa})\log(1/\epsilon)\right)$ for strongly convex problems, which is the same as the best-known result in [Defazio, 2016; Allen-Zhu, 2018], and also matches the upper complexity bound provided in [Woodworth and Srebro, 2016]. In particular, in the case of $n \ge 3L/4\mu$, the convergence rate of both SAGA in [Defazio* et al.*, 2014a] and Finito [Defazio* et al.*, 2014b] is $(1 - \frac{1}{2n})^n \approx \exp(-1/2) = 0.606$, and Katyusha [Allen-Zhu, 2018] can obtain the rate of $1/1.5 = 0.667$, while the convergence rate of AIGD is $(1 - \frac{3}{4n})^n \approx 0.517$. That is, our AIGD algorithm improves the best-known convergence rates.*

Before giving the proof of Theorem 1, we first give the following property.

Property 1. *Given any $w_1, w_2, w_3, w_4 \in \mathbb{R}^d$, then we have*

$$\langle w_1-w_2, w_1-w_3\rangle=\frac{1}{2}\big[\|w_1-w_2\|^2+\|w_1-w_3\|^2-\|w_2-w_3\|^2\big].$$

Proof of Theorem 1:

Proof. From the optimality condition of (4) in Algorithm 1 with respect to z_k with $\eta=\frac{1}{L\alpha\theta}$, we have

$$\theta\langle\widetilde{\nabla}_k+\beta\partial\widetilde{h}(z_k)+L\alpha\theta(z_k-z_{k-1}), z-z_k\rangle\geq 0, \text{for any } z\in\mathbb{R}^d,$$

where $\widetilde{h}(z):=h(z/\beta)$, and $\partial\widetilde{h}(z)=\partial h(z/\beta)/\beta$. Using the above inequality with $z=\beta x^\star$, then we have

$$\begin{aligned}
&\theta\langle\widetilde{\nabla}_k, z_k-\beta x^\star\rangle\\
&\leq\theta\langle\partial h(z_k/\beta), \beta x^\star-z_k\rangle+L\alpha\theta^2\langle z_k-z_{k-1}, \beta x^\star-z_k\rangle\\
&\leq\beta\theta h(x^\star)-\beta\theta h(z_k/\beta)-\frac{\mu\theta}{2\beta}\|\beta x^\star-z_k\|^2\\
&\quad+L\alpha\theta^2\langle z_k-z_{k-1}, \beta x^\star-z_k\rangle\\
&\leq\beta\theta h(x^\star)+(1-\beta\theta)h(x_{k-1})-h(x_k)-\frac{\mu\theta}{2\beta}\|\beta x^\star-z_k\|^2\\
&\quad+\frac{L\alpha\theta^2}{2}\big(\|\beta x^\star-z_{k-1}\|^2-\|\beta x^\star-z_k\|^2-\|z_{k-1}-z_k\|^2\big),
\end{aligned}\tag{5}$$

where the second inequality holds due to the strong convexity of $h(\cdot)$; the last inequality holds due to the update rule $x_k=\theta z_k+(1-\beta\theta)x_{k-1}$, Property 1 and the convexity of $h(\cdot)$.

Since $f(\cdot)$ is L-smooth, and by the update rule $y_k=\theta z_{k-1}+(1-\beta\theta)x_{k-1}$, $x_k=\theta z_k+(1-\beta\theta)x_{k-1}$, and taking expectation over the random choice of i_k, we have

$$\begin{aligned}
&\mathbb{E}_{i_k}[F(x_k)]\\
&\leq\mathbb{E}_{i_k}[f(y_k)+h(x_k)+\theta\langle\widetilde{\nabla}_k,\ z_k-\beta x^\star+\beta x^\star-z_{k-1}\rangle]\\
&\quad+\mathbb{E}_{i_k}[\langle\nabla f(y_k)-\widetilde{\nabla}_k,\ x_k-y_k\rangle+\frac{L\theta^2}{2}\|z_k-z_{k-1}\|^2]\\
&\overset{\text{a}}{\leq}\mathbb{E}_{i_k}\Big[f(y_k)+h(x_k)+\theta\langle\widetilde{\nabla}_k,\ z_k-\beta x^\star+\beta x^\star-z_{k-1}\rangle\Big]\\
&\quad+\mathbb{E}_{i_k}[\frac{1}{2L(\alpha-1)}\|\nabla f(y_k)-\widetilde{\nabla}_k\|^2+\frac{L\alpha\theta^2}{2}\|z_k-z_{k-1}\|^2]\\
&\overset{\text{b}}{\leq}\mathbb{E}_{i_k}[f(y_k)+h(x_k)+\beta\theta h(x^\star)+(1-\beta\theta)h(x_{k-1})-h(x_k)]\\
&\quad+\mathbb{E}_{i_k}\Big[-\frac{\mu\theta}{2\beta}\|\beta x^\star-z_k\|^2+\frac{L\alpha\theta^2}{2}(\|x^\star-z_{k-1}\|^2-\|x^\star-z_k\|^2)\Big]\\
&\quad+\mathbb{E}_{i_k}\Big[\theta\langle\widetilde{\nabla}_k,\ \beta x^\star-z_{k-1}\rangle+\frac{1}{2L(\alpha-1)}\|\nabla f(y_k)-\widetilde{\nabla}_k\|^2\Big]\\
&\overset{\text{c}}{\leq}\mathbb{E}_{i_k}\Big[f(y_k)+\beta\theta h(x^\star)+(1-\beta\theta)h(x_{k-1})-\frac{\mu\theta}{2\beta}\|\beta x^\star-z_k\|^2\Big]\\
&\quad+\frac{L\alpha\theta^2}{2}\mathbb{E}\left[\|\beta x^\star-z_{k-1}\|^2-\|\beta x^\star-z_k\|^2)\right]\\
&\quad+\beta\theta f(x^\star)+(1-\beta\theta)f(x_{k-1})-f(y_k)+A_{k-1}\\
&=\beta\theta F(x^\star)+(1-\beta\theta)F(x_{k-1})+A_{k-1}\\
&\quad+\mathbb{E}_{i_k}[\frac{L\alpha\theta^2}{2}\|\beta x^\star-z_{k-1}\|^2-(\frac{L\alpha\theta^2}{2}+\frac{\mu\theta}{2\beta})\|\beta x^\star-z_k\|^2],
\end{aligned}$$

where the inequality $\overset{\text{a}}{\leq}$ follows from the Young's inequality $\langle\nabla f(y_k)-\widetilde{\nabla}_k, x_k-y_k\rangle\leq\frac{1}{2La}\|\nabla f(y_k)-\widetilde{\nabla}_k\|^2+\frac{La}{2}\|x_k-y_k\|^2$; the inequality $\overset{\text{b}}{\leq}$ follows from the inequality (5), and $A_{k-1}:=\frac{1}{n}\sum_{j=1}^n\big(f_j(\phi_j^{k-1})-f(x^\star)-\langle\nabla f_j(x^\star), \phi_j^{k-1}-x^\star\rangle\big)$; the inequality $\overset{\text{c}}{\leq}$ holds due to the fact that

$$\begin{aligned}
&\mathbb{E}_{i_k}\left[\frac{1}{2L(\alpha-1)}\left\|\widetilde{\nabla}_k-\nabla f(y_k)\right\|^2+\langle\widetilde{\nabla}_k,\ \beta\theta x^\star-\theta z_{k-1}\rangle\right]\\
&=\langle\nabla f(y_k), 2(y_k-x^\star)/(\alpha-1)\rangle+\frac{2}{L(\alpha-1)}[f(x^\star)-f(y_k)]\\
&\quad+\langle\nabla f(y_k),\ \beta\theta x^\star-\theta z_{k-1}\rangle+A_{k-1}\\
&=\langle\nabla f(y_k), (\beta\theta-2/(\alpha-1))\,x^\star-\theta z_{k-1}+2y_{k-1}/(\alpha-1)\rangle\\
&\quad+\frac{2}{\alpha-1}[f(x^\star)-f(y_k)]+A_{k-1}\\
&\leq\beta\theta f(x^\star)+(1-\beta\theta)f(x_{k-1})+A_{k-1},
\end{aligned}$$

where the first equation follows from Lemma 1 and $\mathbb{E}_{i_k}[\widetilde{\nabla}_k]=\nabla f(y_k)$, and the last inequality holds due to the convexity of the function $f(\cdot)$ and $\beta\theta-2/(\alpha-1)>0$. By using Lemma 2 and the above analysis, we have

$$\begin{aligned}
&\mathbb{E}[F(x_k)-F(x^\star)]\\
&\leq(1-\beta\theta)\mathbb{E}[F(x_{k-1})-F(x^\star)]+\frac{2}{\alpha-1}\sum_{j=1}^n\frac{1}{n}F_j(\phi_j^{k-1})-F(x^\star)\\
&\quad+\frac{L\alpha\theta^2}{2}\mathbb{E}\big[\|\beta x^\star-z_{k-1}\|^2\big]-(\frac{L\alpha\theta^2}{2}+\frac{\mu\theta}{2\beta})\mathbb{E}\big[\|\beta x^\star-z_k\|^2\big].
\end{aligned}\tag{6}$$

Using the result in SAGA [Defazio *et al.*, 2014a], we have

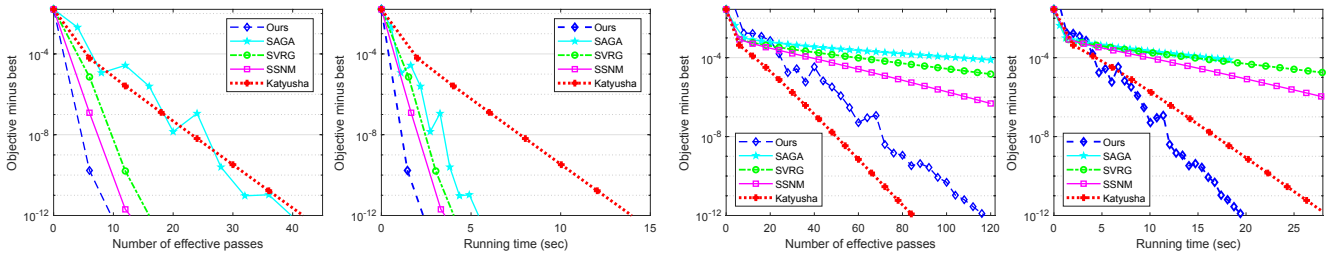
$$\mathbb{E}\left[\sum_{j=1}^n\frac{1}{n}F_j(\phi_j^k)\right]=\mathbb{E}\left[\frac{1}{n}F(x_{k-1})+\frac{n-1}{n^2}\sum_{j=1}^nF_j(\phi_j^{k-1})\right].$$

Thus, $\frac{1}{n}\mathbb{E}[G_{k-1}]=P_k-(1-\frac{1}{n})P_{k-1}$, where $G_k:=F(x_k)-F(x^\star)$, $P_k:=\frac{1}{n}\sum_{j=1}^nF_j(\phi_j^k)-F(x^*)$ and $\mathbb{E}[P_k]=\Sigma_{j=0}^{k-1}(1-/n)^{k-j-1}1/nP_j$. Let $Q_k:=\|\beta x^\star-z_k\|^2$, and using (6) and multiplying each side of the above inequality by γ^{K-k}, $0<\gamma\leq 1$, and summing it over $k=1,\cdots,K-1$, we have

$$\begin{aligned}
&\sum_{k=1}^{K-1}\gamma^{K-k}\mathbb{E}\left[G_k+\frac{1}{2}P_k\right]\\
&\leq\sum_{k=1}^{K-1}\gamma^{K-k}\mathbb{E}\left[(1-\beta\theta+\frac{1}{2n})G_{k-1}+\left(\frac{n-1}{2n}+\frac{2}{\alpha-1}\right)P_{k-1}\right]\\
&\quad+\sum_{k=1}^{K-1}\gamma^{K-k}\left(\frac{L\alpha\theta^2}{2}\mathbb{E}[Q_{k-1}]-(\frac{L\alpha\theta^2}{2}+\frac{\mu\theta}{2})\mathbb{E}[Q_k]\right).
\end{aligned}\tag{7}$$

Next, we consider the following two cases.

Case 1: $n<\frac{3L}{4\mu}$. We set $\gamma=\frac{1}{1+\sqrt{\frac{\mu}{3Ln}}}$. With $\eta=\frac{\sqrt{3}}{\sqrt{L\mu n}}$, $\alpha=8n/\beta$, and $\theta=\frac{1}{L\alpha\eta}$, we have $1>\gamma(1-\beta\theta+\frac{1}{2n})$, $1/2\geq\gamma\left((1-1/n)/2+2/(\alpha-1)\right)$, and $L\alpha\theta^2=\gamma(L\alpha\theta^2+\mu\theta/\beta)$.



(a) The well-conditioned case: $\lambda = 10^{-4}$ (b) The ill-conditioned case: $\lambda = 10^{-7}$

Figure 1: Comparison of all the algorithms for solving logistic regression problems on Covtype.

Thus, (7) is rewritten as follows:

$$\begin{aligned}&\gamma\mathbb{E}[G_{K-1} + \frac{1}{2}P_{K-1}]\\ &\leq O(\gamma^{K-1})\tau\mathbb{E}[G_0] - \mathbb{E}[\frac{\gamma\theta(L\alpha\theta+\mu/\beta)}{2}Q_{K-1}],\end{aligned} \tag{8}$$

where $\tau = 3/2 - 1/(4n) + 1/8$, since $P_0 = G_0$ and $\|x^\star - x_0\| \leq \frac{1}{\mu}G_0$, we have $(1-\beta\theta)G_0 + (\frac{1}{2} - \frac{1}{2n} + \frac{2}{\alpha-1})P_0 + \frac{L\alpha\theta^2}{2}Q_0 = \tau G_0$. Using (6) with $k = K$ and $\alpha = 5$, we have

$$\begin{aligned}\mathbb{E}[G_K] \leq &\mathbb{E}\left[(1-\beta\theta)G_{K-1} + \frac{1}{2}P_{K-1}\right]\\ &+ \mathbb{E}\left[\frac{5L\theta^2}{2}Q_{K-1} - (\frac{5L\theta^2}{2} + \frac{\mu\theta}{2\beta})Q_K\right].\end{aligned} \tag{9}$$

Since $\gamma > (1-\beta\theta)$ and $\frac{5L\theta^2}{2} < \frac{\gamma\theta(L\alpha\theta+\mu/\beta)}{2}$, and by using the above two inequalities, we have

$$\mathbb{E}[G_K] \leq O(\gamma^{K-1})\tau\mathbb{E}[G_0] - \mathbb{E}[(\frac{5L\theta^2}{2} + \frac{\mu\theta}{2})Q_K].$$

That is,

$$\mathbb{E}[G_k] \leq O\left(\left(\frac{1}{1+\sqrt{\frac{\mu}{3Ln}}}\right)^K\right)G_0.$$

Case 2: $n \geq \frac{3L}{4\mu}$. We set $\gamma = \frac{1}{1+3/(4n)} = 1 - 3/(4n+3)$. With $\eta = \frac{3}{4\mu n}$, $\alpha = 8n/\beta$ and $\theta = \frac{1}{L\alpha\eta}$, and using the similar derivation, we have the following result

$$\mathbb{E}\,[G_K] \leq O\left(\left(1 - \frac{3}{4n}\right)^K\right)G_0.$$

This completes the proof. □

5 Experimental Results

In this section, we evaluate the performance of our algorithm for justifying our theoretical results.

We conducted many experiments of the strongly convex logistic regression problem on the two real-world data sets: Covtype (581,012 examples and 54 features) and a9a (32,562 examples and 123 features). Note that each feature vector of the two data sets was scaled down by the average Euclidean norm of the whole data set as in [Allen-Zhu and Yuan, 2016]. For SVRG and Katyusha, we set the epoch size $m = 2n$, as suggested in [Johnson and Zhang, 2013; Allen-Zhu, 2018]. Figs. 1 and 2 show how the objective gap (i.e., $F(x_K) - F(x^\star)$) of all these algorithms decreases for logistic regression with different regularization parameters $\lambda = 10^{-4}, 10^{-7}, 10^{-8}$. All the results show that the accelerated methods (i.e., Katyusha, SSNM and our algorithm) usually perform much better than the non-accelerated methods, SVRG and SAGA, especially with relatively small regularization parameters, e.g., $\lambda = 10^{-7}$. SAGA and Katyusha achieve similar performance for the well-conditioned case, while Katyusha is significantly faster than SAGA for the ill-conditioned case (i.e., the case of small regularization parameters). Our algorithm outperforms the other methods (including the accelerated algorithms, Katyusha and SSNM) in terms of running time. This further justifies the effectiveness of our momentum acceleration technique for accelerating SAGA.

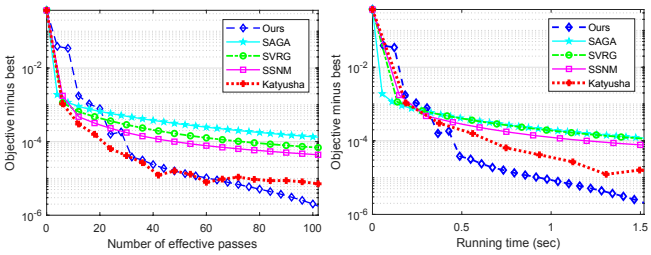


Figure 2: Comparison of all the algorithms for solving logistic regression problems for the ill-conditioned case ($\lambda = 10^{-8}$) on a9a.

6 Conclusions

In this paper, we proposed a novel accelerated incremental gradient descent algorithm with the proposed momentum acceleration technique. Unlike the existing accelerated algorithms such as point-SAGA and SSNM, our algorithm is a direct accelerated method, and requires significantly less memory than SSNM for various large-scale linear models. Moreover, we provided the convergence property of our algorithm for solving strongly convex problems, which shows that our algorithm attains the best-known oracle complexity, and an improved convergence rate for the case of $n \geq L/\mu$. Our algorithm can be extended to non-convex setting and constrained composite convex minimization setting, especially, the problems of the general ADMM form as in [Liu *et al.*, 2017].

Acknowledgments

This work was supported by Project supported Foundation for the 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), the National Natural Science Foundation of China (Nos. 61876220, 61876221, 61836009, U1701267, 61871310, 61573267, 61502369 and 61473215), the Science Foundation of Xidian University (Nos. 10251180018 and 10251180019), 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 (No. B07048), and the Key Special Project of China High Resolution Earth Observation System-Young Scholar Innovation Fund.

References

[Allen-Zhu and Hazan, 2016] Zeyuan Allen-Zhu and Elad Hazan. Optimal black-box reductions between optimization objectives. In *NIPS*, 2016.

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

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

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

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

[Defazio *et al.*, 2014b] Aaron Defazio, Tiberio S. Caetano, and Justin Domke. Finito: A faster, permutable incremental gradient method for big data problems. In *ICML*, pages 1125–1133, 2014.

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

[Frostig *et al.*, 2015] Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML*, pages 2540–2548, 2015.

[Hien *et al.*, 2019] Le Thi Khanh Hien, Cuong V. Nguyen, Huan Xu, Canyi Lu, and Jiashi Feng. Accelerated stochastic mirror descent algorithms for composite non-strongly convex optimization. *J. Optimiz. Theory App.*, 2019.

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

[Kim *et al.*, 2009] Seyoung Kim, Kyung-Ah Sohn, and Eric P. Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25:i204–i212, 2009.

[Lin *et al.*, 2015] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *NIPS*, pages 3366–3374, 2015.

[Liu *et al.*, 2017] Yuanyuan Liu, Fanhua Shang, and James Cheng. Accelerated variance reduced stochastic ADMM. In *AAAI*, pages 2287–2293, 2017.

[Mahdavi *et al.*, 2013] Mehrdad Mahdavi, Lijun Zhang, and Rong Jin. Mixed optimization for smooth functions. In *NIPS*, pages 674–682, 2013.

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

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

[Roux *et al.*, 2012] Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, pages 2672–2680, 2012.

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

[Shang *et al.*, 2017] Fanhua Shang, Yuanyuan Liu, James Cheng, and Jiacheng Zhuo. Fast stochastic variance reduced gradient method with momentum acceleration for machine learning. *arXiv:1703.07948*, 2017.

[Shang *et al.*, 2018a] Fanhua Shang, Licheng Jiao, Kaiwen Zhou, James Cheng, Yan Ren, and Yufei Jin. ASVRG: Accelerated proximal SVRG. In *P. Mach. Learn. Res.*, pages 815–830, 2018.

[Shang *et al.*, 2018b] Fanhua Shang, Yuanyuan Liu, James Cheng, K. W. Ng, and Yuichi Yoshida. Guaranteed sufficient decrease for stochastic variance reduced gradient optimization. In *AISTATS*, pages 1027–1036, 2018.

[Shang *et al.*, 2019] Fanhua Shang, Kaiwen Zhou, Hongying Liu, James Cheng, Ivor Tsang, Lijun Zhang, Dacheng Tao, and Licheng Jiao. VR-SGD: A simple stochastic variance reduction method for machine learning. *IEEE Trans. Knowl. Data Eng.*, 2019.

[Tibshirani and Taylor, 2011] Ryan J. Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *Annals of Statistics*, 39(3):1335–1371, 2011.

[Woodworth and Srebro, 2016] Blake Woodworth and Nathan Srebro. Tight complexity bounds for optimizing composite objectives. In *NIPS*, 2016.

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

[Zhang *et al.*, 2013] Lijun Zhang, Mehrdad Mahdavi, and Rong Jin. Linear convergence with condition number independent access of full gradients. In *NIPS*, pages 980–988, 2013.

[Zhou *et al.*, 2018] Kaiwen Zhou, Fanhua Shang, and James Cheng. A simple stochastic variance reduced algorithm with fast convergence rates. In *ICML*, pages 5975–5984, 2018.

[Zhou *et al.*, 2019] Kaiwen Zhou, Qinghua Ding, Fanhua Shang, James Cheng, Danli Li, and Zhi-Quan Luo. Direct acceleration of SAGA using sampled negative momentum. In *AISTATS*, 2019.