TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 02/14 pp999–1008
DOI: 10.26599/TST.2022.9010051
Volume 28, Number 6, December 2023

A Variance Reducing Stochastic Proximal Method with Acceleration Techniques

Jialin Lei, Ying Zhang*, and Zhao Zhang

Abstract: We consider a fundamental problem in the field of machine learning—structural risk minimization, which can be represented as the average of a large number of smooth component functions plus a simple and convex (but possibly non-smooth) function. In this paper, we propose a novel proximal variance reducing stochastic method building on the introduced Point-SAGA. Our method achieves two proximal operator calculations by combining the fast Douglas–Rachford splitting and refers to the scheme of the FISTA algorithm in the choice of momentum factors. We show that the objective function value converges to the iteration point at the rate of $\mathcal{O}(1/k)$ when each loss function is convex and smooth. In addition, we prove that our method achieves a linear convergence rate for strongly convex and smooth loss functions. Experiments demonstrate the effectiveness of the proposed algorithm, especially when the loss function is ill-conditioned with good acceleration.

Key words: composite optimization; Variance Reduction (VR); fast Douglas–Rachford (DR) splitting; proximal operator

1 Introduction

In this paper, we consider the following convex optimization problem with a finite-sum structure, which is prevalent in machine learning[1] and statistics, such as supervised learning[2] and regularized empirical risk minimization[3],

$$\min_{x\in\mathbf{R}^d} F(x) = f(x) + h(x) \tag{1}$$

where $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ is an average of a set of convex and smooth loss functions $f_i(x)$, and $h(x)$ is a sample and convex (but possibly non-smooth) function. The goal is to find the optimal solution of x that minimizes the regularized empirical loss over the whole dataset.

When $h(x)$ is absent, traditional analysis shows that Gradient Descent (GD) yields a fast linear convergence rate but with a high per-iteration cost, and thus may not be suitable for problems with a very large n. As an alternative for large-scale problems, Stochastic Gradient Descent (SGD)[4] uses only one or a mini-batch of gradients in each iteration, and thus enjoys a significantly lower per-iteration complexity than GD. However, due to the undiminished variance of the gradient estimator, prompting SGD is shown to yield only a sub-linear convergence rate.

To effectively eliminate the variance generated by SGD, a number of Variance Reducing (VR) stochastic methods have been developed in recent years, such as SVRG[5], SAGA[6], and SDCA[7]. SVRG and SAGA are two typical algorithms among them, whose general formula of following VR stochastic gradient can be expressed as

$$\nabla f_j(x^k) - \nabla f_j(\tilde{x}) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{x}) \tag{2}$$

where $\tilde{x}$ is the saved "snapshot" of a previous x, and it is updated once every m iterations (where m is the number of internal cycles). The last two items in Formula (2)

• Jialin Lei, Ying Zhang, and Zhao Zhang are with School of Mathematical Science, Zhejiang Normal University, Jinhua 321004, China. E-mail: jialinlei@zjnu.edu.cn; znuzy@zjnu.cn; zhaozhang@zjnu.cn.

* To whom correspondence should be addressed.

Manuscript received: 2022-07-01; revised: 2022-10-13; accepted: 2022-10-30

indicate the deviations of the control variables from the unbiased estimates. The variance of this method goes to zero asymptotically along the iterative updates as $\nabla f_j(x^k)$ and $\nabla f_j(\tilde{x})$ become closer in expectation, which lead to a linear convergence rate which is much faster than that of SGD. As a result, the convergence rate can be improved from sub-linear in SGD to linear in the VR stochastic methods.

Further, for the non-smooth regularization term $h(x)$, there are also some proximal variants of stochastic optimization algorithms, such as Prox-SVRG[8], Prox-SAGA[6], and Prox-SDCA[9], which were proposed to solve Formula (1) and all apply variance reducing techniques to achieve low per-iteration and maintain a fast linear convergence rate at the same time. More recently, researchers have proposed several accelerated stochastic variance reduced methods, which include Point-SAGA[10], Catalyst[11], Katyusha[12], MIG[13], and so on. These methods can be boosted to faster convergence rates when the loss function $f_i(x)$ is ill-conditioned. In terms of oracle complexity, Prox-SAGA and Prox-SVRG both require $\mathcal{O}((n + L/\mu)\log(1/\epsilon))$ steps to achieve an ϵ-accurate solution, as compared with $\mathcal{O}((n + \sqrt{nL/\mu})\log(1/\epsilon))$ for accelerated methods (e.g., Point-SAGA, katyusha, and MIG).

In this paper, we develop an accelerated variance reducing stochastic proximal method which is called Accelerated Double Proximal operator SAGA (ADProx-SAGA). Unlike the variance reduction algorithm that contains only one proximal operator, ADProx-SAGA uses the corresponding gradient mappings, achieves proximal calculations twice by combining with fast Douglas–Rachford (DR) splitting, and with reference to the Nesterov's momentum[14] of FISTA[15] algorithm, enabling ADProx-SAGA to achieve an accelerated convergence rate for strongly convex problem, and the proposed algorithm has a good performance when the loss function is ill-conditioned with good acceleration.

2 ADProx-SAGA and Simplification

In this section, we introduce a variance reducing stochastic proximal method[16] with accelerate techniques.

2.1 ADProx-SAGA algorithm

ADProx-SAGA algorithm is introduced in Algorithm 1. It is easy to see that this algorithm maintains five sequences, x^k, y^k, z_j^k, u^k, and g_j^k, which are the iterations points. The initial point x^0 is chosen arbitrarily,

Algorithm 1 ADProx-SAGA algorithm

1: **Input:** Initial point x^0, learning date γ, and momentum factor $\beta_k = \frac{k-2}{k+1}$.
2: **for** $k = 1, 2, \ldots$ **do**
3: Uniformly randomly pick j from 1 to n.
4: Update x in the following:

$$z_j^k = x^k + \gamma(g_j^k - \frac{1}{n}\sum_{i=1}^{n} g_i^k) \tag{3}$$

$$y^{k+1} = z_j^k - \gamma g_j^{k+1} \tag{4}$$

$$u^{k+1} = y^{k+1} + \beta_{k+1}(y^{k+1} - y^k) \tag{5}$$

$$x^{k+1} = \mathrm{prox}_h^{\gamma}(u^{k+1}) \tag{6}$$

5: Selection of itertion point: Supposed that the point after proximal mapping is ω, then the update of the iteration point is as follows:

$$x^{k+1} = \begin{cases} \omega, & F(\omega) \leqslant F(x^k); \\ x^k, & F(\omega) > F(x^k). \end{cases}$$

6: Update the gradient table: Set $v_j^k = z_j^k + x^k - y^k$ and $g_j^{k+1} = \frac{1}{\gamma}(v_j^k - \mathrm{prox}_f^{\gamma}(v_j^k))$, and leave the rest of the entries unchanged ($g_j^{k+1} = g_j^k$ for $i \neq j$).
7: **end for**
8: **Output:** x^{k+1}.

g_i^0 is chosen as gradient/subgradient of $f_i(x)$ at x^0, and the algorithm adds momentum factor β_k, in addition to the parameter learning date γ.

In the k-th iteration, the loss function $f_i(x)$ is chosen randomly. The variable y^k is updated in terms of Eq. (4), u^k iterates over y^k using Eq. (5). According to the definition of z_j^k in Eq. (3) and the update of x^{k+1} in Eq. (6), z_j^k, y^{k+1}, and u^{k+1} can be treated as intermediate variables in the update from x^k to x^{k+1}. According to Eqs. (2) and (3), the main steps of Algorithm 1 can also be written as

$$y^{k+1} = x^k - \gamma(g_j^{k+1} - g_j^k + \frac{1}{n}\sum_{i=1}^{n} g_i^k),$$

where $g_j^{k+1} - g_j^k + \frac{1}{n}\sum_{i=1}^{n} g_i^k$ indicates the unbiased estimation of the gradient g_j^k, which is similar to the gradient update formula in SAGA. The gradient table is designed to reduce the computational effort by $1/n$ compared to SVRG, and g_j^{k+1} is the gradient mapping of $f_j(x)$ at $z_j^k + x^k - y^k$.

In each iteration of our algorithm, unlike Prox-SAGA which contains only one proximal operator to handle the nonsmooth term $h(x)$, we borrow the idea of the Douglas–Rachford splitting to use the proximal operator of $f_i(x)$ to calculate the gradient mapping, in addition to the proximal operator of $h(x)$, enabling it to achieve two proximal calculations, and this design can achieve

fast convergence when the loss function $f_i(x)$ is ill-conditioned.

In particular, our algorithm differs from Prox-SAGA in the definition of gradient. In Prox-SAGA, g_j^{k+1} is the gradient of $f_i(x)$ at x^k, while in our algorithm, g_j^{k+1} is the subgradient of x at the point $\text{prox}_{f_j}^{\gamma}(z_j^k + x^k - y^k)$. We have learned that Point-SAGA achieves the effect of acceleration of SAGA by involving the "future" point x^{k+1}. According to Eq. (5) and the definition of g_j^{k+1}, $\text{prox}_{f_j}^{\gamma}(z_j^k + x^k - y^k) = y^{k+1} + x^k - y^k$ can be obtained. We can see that Algorithm 1 also involves "future" points and the two proximal operators are combined by using DR splitting. In addition, on top of combining DR splitting operators, we can achieve a faster convergence compared to Prox-SAGA and Point-SAGA by adding momentum terms to the iteration points.

We wish to speed up the proximal point gradient algorithm, so we apply Nesterov's momentum at the iteration point y^{k+1}. For the choice of value of β_k, we set $\beta_k = \frac{k-2}{k+1}$ in Algorithm 1 according to the FISTA algorithm. However, FISTA algorithm is not guaranteed to be a descent algorithm, so it is necessary to verify the function value at the iteration point, and this step is essential in the later theoretical proofs.

In this paper, two cases (strongly convex and non-strongly convex) are considered for the properties of the objective function $f_i(x)$, and different values of γ are selected for these two cases. Parameters L and μ are unknown for most problems, so ADProx-SAGA will works well in practical.

In this section, we show that ADProx-SAGA is actually a combination of Point-SAGA and DR splitting operators, and is a generalization of the Point-SAGA while establishing a connection to fast Douglas–Rachford splitting.

2.2 Simplification

A well-known algorithm for solving composite optimization problem is the DR splitting algorithm[17]. In the general case where the corresponding operators are the subdifferentials of $f_i(x)$ and $h(x)$, DR splitting algorithm can be written as the following iterations:

$$\begin{aligned} x^k &= \text{prox}_h^{\gamma}(y^k), \\ z^k &= \text{prox}_f^{\gamma}(2x^k - y^k), \\ y^{k+1} &= y^k + \lambda_k(z^k - x^k), \end{aligned}$$

where $\gamma > 0$ and a typical choice for λ_k is to be set equal to 1 for all k. There is $g_j^k = \sum_{i=1}^n g_i^k / n$ in Algorithm 1 when $n = 1$, thus we have $z_j^k = x^k$ according to Eq. (3), and the main iterative process can be simplified as

$$\begin{aligned} x^k &= \text{prox}_h^{\gamma}(y^k), \\ y^{k+1} &= y^k + \text{prox}_{f_j}^{\gamma}(2x^k - y^k) - x^k. \end{aligned}$$

In this way, our algorithm is essentially the typical DR splitting algorithm. By adding the momentum term $y^{k+1} - y^k$, we can obtain a fast DR splitting method and apply this accelerate technique to our method as well.

Our algorithm is closely linked to Point-SAGA, and it is not difficult to find that the proximal acceleration about the nonsmooth term is removed when $h(x) = 0$, thus we have $x^k = y^k$ in our algorithm, and the main iteration can be simplified in the following:

$$\begin{aligned} z_j^k &= x^k + \gamma(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k), \\ y^{k+1} &= \text{prox}_{f_j}^{\gamma}(z_j^k), \\ x^{k+1} &= y^{k+1} + \beta_{k+1}(y^{k+1} - y^k). \end{aligned}$$

Obviously, this is the iteration of Point-SAGA plus Nesterov's momentum tricks. In contrast to Point-SAGA, our algorithm does a combination of the two proximal operators by DR splitting and adds momentum term to iteration point for acceleration. Point-SAGA is known to achieve an accelerated convergence rate on strongly convex smooth problems, and our algorithm has the same advantage.

3 Related Theorem

In this section, we show that the stationary point of our algorithm is the minimum point of the objective function, and furthermore, the algorithm can achieve a convergence rate of $\mathcal{O}(1/k)$ when each $f_i(x)$ in the objective function is smooth, and it can achieve linear convergence when $f_i(x)$ is further assumed to be strongly convex. Before proceeding to the practical analysis, let us present some theoretical results.

3.1 Assumption and properties

To better distinguish the objective function, we have made the following two assumptions about $f_i(x)$ in Eq. (1).

Assumption 1 For each $f_i(x)$ having an L-Lipschitz continuous gradient for any $x, y \in \mathbf{R}^d$, there is a constant L, such that

$$f_i(y) \leqslant f_i(x) + \langle \nabla f_i(x), y - x \rangle + \frac{L}{2}\|y - x\|^2,$$

where $L > 0$ and $\nabla f_i(x)$ is a gradient at x.

Assumption 2 For each $f_i(x)$ is μ-strongly convex, we also define

$$f_i(y) \geqslant f_i(x) + \langle g_i, y-x\rangle + \frac{\mu}{2}\|y-x\|^2,$$

where $\mu > 0$ and $g_i \in \partial f_i(x)$ is a set of sub-gradient of $f_i(x)$.

The assumption can be satisfied by refining $f_i(x)$ with a strongly convex regularizer. For a general convex function, the above inequality always holds with $\mu = 0$, and the following propositions are the focus of this paper.

Our analysis is based on the theoretical basis of Moreau envelope[18]. The Moreau envelope of a pretty smooth approximation function $f\colon \mathbf{R}^d \to \mathbf{R}$ with a regularization parameter $\gamma > 0$ is defined as

$$f^\gamma(x) = \inf_y \left\{ f(y) + \frac{1}{2\gamma}\|x-y\|^2 \right\} \tag{7}$$

Some properties of Moreau envelope in question are given below.

Proposition 1 (Moreau decomposition) For any $x, y \in \mathbf{R}^d$, and any convex function $f : \mathbf{R}^d \to \mathbf{R}$ with Fenchel conjugate f^*, we have

$$\operatorname{prox}_f^\gamma(x) = x - \gamma \operatorname{prox}_{f^*}^{1/\gamma}(x/\gamma).$$

Recalling the defintion of $g_f^\gamma(x)$, after combing Proposition 1, we have the following relation between the proximal operator of the conjugate f^* and $g_f^\gamma(x)$:

$$g^\gamma(x) = \frac{1}{\gamma}(x - \operatorname{prox}_f^\gamma(x)) = \operatorname{prox}_{f^*}^\gamma(x/\gamma) \tag{8}$$

Proposition 2 (Firm non-expansiveness) For any $x, y \in \mathbf{R}^d$, any convex function $f : \mathbf{R}^d \to \mathbf{R}$ with strong convexity constant $\mu \geqslant 0$, and any regularization parameter $\gamma > 0$, we have the firm non-expansiveness of $\operatorname{prox}_f^\gamma(x)$ in the following:

$$\langle \operatorname{prox}_f^\gamma(x) - \operatorname{prox}_f^\gamma(y), x-y\rangle \geqslant (1+\mu\gamma)\|\operatorname{prox}_f^\gamma(x) - \operatorname{prox}_f^\gamma(y)\|^2 \tag{9}$$

From the above result we can also deduce

$$\|2\operatorname{prox}_f^\gamma(x) - x - (2\operatorname{prox}_f^\gamma(y) - y)\| \leqslant \|x-y\| \tag{10}$$

The proofs of Propositions 1 and 2 are in Appendix A.

Proposition 3 For any $x, y \in \mathbf{R}^d$, any L-smooth function $f_L : \mathbf{R}^d \to \mathbf{R}$, and any regularization parameter $\gamma > 0$, we have

$$\langle g_f^\gamma(x) - g_f^\gamma(y), x-y\rangle \geqslant \gamma(1+\frac{1}{L\gamma})\|\langle g_f^\gamma(x) - g_f^\gamma(y)\|^2 \tag{11}$$

Proof It is known that f^* is the conjugate function of f, and L-smoothness of f_L implies $\frac{1}{L}$-strong convexity of f^*, then we apply Proposition 2 to the points x/γ and y/γ,

$$\langle \operatorname{prox}_{f^*}^{1/\gamma}(x/\gamma) - \operatorname{prox}_{f^*}^{1/\gamma}(y/\gamma), x/\gamma - y/\gamma\rangle \geqslant (1+\mu\gamma)\|\operatorname{prox}_{f^*}^{1/\gamma}(x/\gamma) - \operatorname{prox}_{f^*}^{1/\gamma}(y/\gamma)\|^2 \tag{12}$$

with the definition of the gradient $g_f^\gamma(x)$ in Proposition 1 and the firm non-expansive of the convex function $f(x)$, simplifying Formula (12) leads to Formula (11). ■

Before giving the main result, let x^* be the unique minimizer of $f_i(x)$ due to the strong convexity. In addition to the notation used in the description of Algorithm 1, there exists a set of subgradients g_j^*, and chosen $\sum_{j=1}^n g_j^* = 0$ for each $f_j(\cdot)$ at x^*, where $f_j(\cdot)$ denotes a function of the random sample j. Then we show that x^* is a minimizer of Eq. (1) if x^* exists.

Proposition 4 Suppose that y^* is the fixed point and g_i^* is the fixed set of Algorithm 1, then $x^* = \operatorname{prox}_h^\gamma(u^*)$ is the minimizer of Eq. (1), where u^* represents the optimal value of the current stage.

Proof To prove that y^* is the minimizer of Eq. (1), it is necessary to show that $0 \in \partial f(x^*) + \partial h(x^*)$ according the first-order optimality condition of subgradient. From the supposition we have $y^* = -x^* + y^* + \operatorname{prox}_{f_i}^\gamma(z_i^* + x^* - y^*)$, where z_i^* is the optimal value of the current stage. Meanwhile, since $x^* = \operatorname{prox}_h^\gamma(u^*)$, which implies

$$(z_i^* - y^*)/\gamma \in \partial f_i(x^*),\ (u^* - x^*)/\gamma \in \partial h(x^*).$$

where $i = 1, 2, \ldots, n$. Because $u^{k+1} = y^{k+1} + \beta_k(y^{k+1} - y^k)$, we can deduce that $u^* = y^*$, so we have

$$\frac{1}{n}\sum_{i=1}^n (z_i^* - y^*) + (u^* - x^*) = \frac{1}{n}\sum_{i=1}^n z_i^* - x^* = 0 \in \partial f(x^*) + \partial h(x^*).$$ ■

3.2 Main result

The proof of the main results relies on a Lyapunov function[19], which is defined in Algorithm 1 at Step $k+1$,

$$T^{k+1} = \frac{c}{n}\sum_{i=1}^n \|\gamma(g_i^{k+1} - g_i^*)\|^2 + \|x^{k+1} - x^*\|^2 \tag{13}$$

where $c > 0$ is a constant. Different c will be used later in the non-strongly convex[20] and strongly convex cases[21], and the upper bound on the expectation of the Lyapunov function is given below.

Theorem 1 Supposed that each $f : \mathbf{R}^d \to \mathbf{R}$ with strong convexity constant $\mu \geqslant 0$, while the

regularization term $h(x)$ is convex (but possibly non-smooth), over the random choice of j, the expectation of the Lyapunov function satisfies

$$\begin{aligned}E[T^{k+1}] \leqslant & \left[\frac{(1+\beta_k)^2}{2}+\left(1-\frac{1}{n}\right)c\right]\times \\ & \frac{1}{n}\sum_{i=1}^{n}\|\gamma(g_i^k-g_i^*)\|^2+ \\ & (2(1+\beta_k)^2+\frac{c}{n})E\|\gamma(g_j^{k+1}-g_j^*)\|^2+ \\ & \frac{(1-\beta_k)^2}{2}\|u^k-u^*\|^2+ \\ & \frac{(1+\beta_k)^2}{2}E\|v_j^k-v_j^*\|^2- \\ & 2(1+\beta_k)^2E\langle v_j^k-v_j^*,\gamma(g_j^{k+1}-g_j^*)\rangle\end{aligned} \tag{14}$$

Proof The first term in Formula (13) $\frac{c}{n}\sum_{i=1}^{n}\cdot\|\gamma(g_i^{k+1}-g_i^*)\|^2$ is simplified to

$$\begin{aligned}&\frac{c}{n}E\sum_{i=1}^{n}\|\gamma(g_i^{k+1}-g_i^*)\|^2= \\ &(1-\frac{1}{n})\cdot\frac{c}{n}\sum_{i=1}^{n}\|\gamma(g_i^k-g_i^*)\|^2+ \\ &\frac{c}{n}E\|\gamma(g_j^{k+1}-g_j^*)\|^2\end{aligned} \tag{15}$$

To calculate the expectation for the second term in Formula (13) $\|x^{k+1}-x^*\|^2$, recalling the definition of z_j^k and v_j^k in Algorithm 1, y^{k+1} can be expressed in the form as follows:

$$\begin{aligned}y^{k+1}=&\frac{1}{2}(y^k+\gamma(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k))+ \\ &\frac{1}{2}(2\operatorname{prox}_{f_j}^{\gamma}(v_j^k)-v_j^k).\end{aligned}$$

By the definition of u^{k+1}, u^{k+1} can be written as

$$\begin{aligned}u^{k+1}=&\frac{1-\beta_k}{2}y^k+\frac{1+\beta_k}{2}\gamma(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k)+ \\ &\frac{1+\beta_k}{2}(2\operatorname{prox}_{f_j}^{\gamma}(v_j^k)-v_j^k).\end{aligned}$$

Similarly, u^* is obtained as

$$\begin{aligned}u^*=&\frac{1-\beta_k}{2}y^*+\frac{1+\beta_k}{2}\gamma(g_j^*-\frac{1}{n}\sum_{i=1}^{n}g_i^*)+ \\ &\frac{1+\beta_k}{2}(2\operatorname{prox}_{f_j}^{\gamma}(v_j^*)-v_j^*).\end{aligned}$$

Then, the expectation about the square of $u^{k+1}-u^*$ can be described as

$$\begin{aligned}&E\|u^{k+1}-u^*\|^2= \\ &\frac{1}{4}E\|(1-\beta_k)(y^k-y^*)+ \\ &(1+\beta_k)\gamma(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*)+ \\ &(1+\beta_k)[(2\operatorname{prox}_{f_j}^{\gamma}(v_j^k)-v_j^k)- \\ &(2\operatorname{prox}_{f_j}^{\gamma}(v_j^*)-v_j^*)]\|^2\leqslant \\ &\frac{1}{2}E\|(1-\beta_k)(y^k-y^*)+ \\ &(1+\beta_k)\gamma(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*)\|^2+ \\ &\frac{1}{2}E\|(1+\beta_k)[(2\operatorname{prox}_{f_j}^{\gamma}(v_j^k)-v_j^k)- \\ &(2\operatorname{prox}_{f_j}^{\gamma}(v_j^*)-v_j^*)]\|^2\end{aligned} \tag{16}$$

The independent sample j is randomly selected, and two particularly useful expectations about g_j are $E[g_j^k]=\frac{1}{n}\sum_{i=1}^{n}g_i^k$ and $E[g_j^*]=g^*$, so we obtain $E(x^k-x^*,\gamma(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*))=0$. Formula (16) is established by using $\|a+b\|^2\leqslant 2\|a\|^2+2\|b\|^2$. For the first two terms in Formula (16), $E\|(1-\beta_k)(y^k-y^*)+(1+\beta_k)\gamma(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g)\|^2$, we have

$$\begin{aligned}&E\|(1-\beta_k)(y^k-y^*)+(1+\beta_k) \\ &\gamma(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*)\|^2= \\ &(1-\beta_k)^2E\|y^k-y^*\|^2+ \\ &(1+\beta_k)^2E\|\gamma(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*)\|^2\leqslant \\ &(1-\beta_k)^2E\|u^k-u^*\|^2+ \\ &(1+\beta_k)^2E\|\gamma(g_j^k-g_j^*)\|^2\end{aligned} \tag{17}$$

With the correct gradient direction, according to the definition of u^k and the variance formula $E[(X-E[X])^2]=E[X^2]-E[X]^2\leqslant E[X^2]$, where $E(g_j^k-g_j^*)=\frac{1}{n}\sum_{i=1}^{n}g_i^k-g^*$, Formula (17) holds.

For calculating the term $E\|(1+\beta_k)[(2\operatorname{prox}_{f_j}^{\gamma}(v_j^k)-v_j^k)-(2\operatorname{prox}_{f_j}^{\gamma}(v_j^*)-v_j^*)]\|^2$ in Formula (16), by using $g_j^{k+1}=\frac{1}{\gamma}(v_j^k-\operatorname{prox}_{f_j}^{\gamma}(v_j^k))$, we observe that $\gamma g_j^{k+1}=v_j^k-\operatorname{prox}_{f_j}^{\gamma}(v_j^k)$, so $2\operatorname{prox}_{f_j}^{\gamma}(v_j^k)-v_j^k$ can be written as $v_j^k-2\gamma g_j^{k+1}$, or $2\operatorname{prox}_{f_j}^{\gamma}(v_j^*)-v_j^*$ likewise, then we have

$$E\|(1+\beta_k)[(2\,\mathrm{prox}_{f_j}^{\gamma}(v_j^k)-v_j^k)-(2\,\mathrm{prox}_{f_j}^{\gamma}(v_j^*)-v_j^*)\|^2=(1+\beta_k)^2E\|v_j^k-2\gamma g_j^{k+1}-v_j^*+2\gamma g_j^*]\|^2=(1+\beta_k)^2[E\|v_j^k-v_j^*\|^2+4E\|\gamma(g_j^{k+1}-g_j^*)\|^2-4E\langle v_j^k-v_j^*,\gamma(g_j^{k+1}-g_j^*)\rangle] \tag{18}$$

By substituting Formula (17) and Eq. (18) into Formula (16), and combining with the Eq. (15), Formula (14) has been proved. ■

Theorem 2 (Non-strongly convex case) Suppose that each $f_i:\mathbf{R}^d\to\mathbf{R}$ is convex and L-smooth, while the regularization term h is convex. Then for Algorithm 1 with step size $\gamma\leqslant 1/L$, at Step k, we have

$$E\|\bar{g}_j^k-g_j^*\|^2\leqslant\frac{1}{k}\Big(\sum_{i=1}^{n}\|g_i^0-g_i^*\|^2+\|\frac{1}{\gamma}(u^0-u^*)\|^2\Big) \tag{19}$$

where $\bar{g}_j^k=\frac{1}{k}\sum_{t=1}^{k}g_j^t$, and the expectation of $\|\bar{g}_j^k-g_j^*\|^2$ is for all choices of index j at Step k.

Theorem 3 (Strongly convex case) Supposed that each $f_i:\mathbf{R}^d\to\mathbf{R}$ is μ-strongly convex and L-smooth, while the regularization term $h(x)$ is convex. Then for Algorithm 1 with step size $\gamma=\min\left\{\frac{1}{(1+\beta_k)^2\mu n},\frac{\sqrt{9(1+\beta_k^2)^2L^2+3(1-\beta_k)^2(1+\beta_k^2)}-3(1+\beta_k^2)L}{2(1-\beta_k)^2\mu L}\right\}$ at Step k, we have

$$E\|x^k-x^*\|^2\leqslant[1+\beta_k^2-(1+\beta_k^2)\frac{\mu\gamma}{2(\mu\gamma+1)}]^k\times\frac{(1-\beta_k)^2\mu\gamma+2(1+\beta_k^2)}{2-(1+\beta_k)^2n\mu\gamma}\times\Big\{\sum_{i=1}^{n}\|\gamma(g_i^0-g_i^*)\|^2+\|u^0-u^*\|^2\Big\} \tag{20}$$

The relevant proofs are in Appendix B.

4 Experiment

In this section, we conduct numerical experiments to examine the practical performance of the proposed method which we call ADProx-SAGA. We main focus on ℓ_2-logistic regression,

$$\min_x F(x)=\frac{1}{n}\sum_{i=1}^{n}\log\left(1+\exp\left(-b_ia_i^{\mathrm{T}}x\right)\right)+\frac{\lambda}{2}\|x\|^2 \tag{21}$$

where $a_i\in\mathbf{R}^d, b_i\in\{-1,+1\}, i=1,2,\ldots,n$. We verify the effectiveness of ADProx-SAGA to deal with composite problems, and for the ℓ_2-logistic regression model in Eq. (21), which contains smooth logistic function as well as ℓ_2 regularization term, we use different values of λ for each dataset as a way to verify the ability of ADProx-SAGA to handle ill-conditioned problem and acceleration performance in practice.

The datasets we acted on include covtype (581 012 samples and 54 features), a9a (49 749 samples and 300 features), mushrooms (abbreviated as "mus" in Fig. 1, 8124 samples and 112 features), and w7a (24 692 samples and 300 features), and λ is set to 10^{-4} and 10^{-8}. The experiments are designed as some ill-conditioned problems (with very small λ). We test Prox-SGD, Prox-

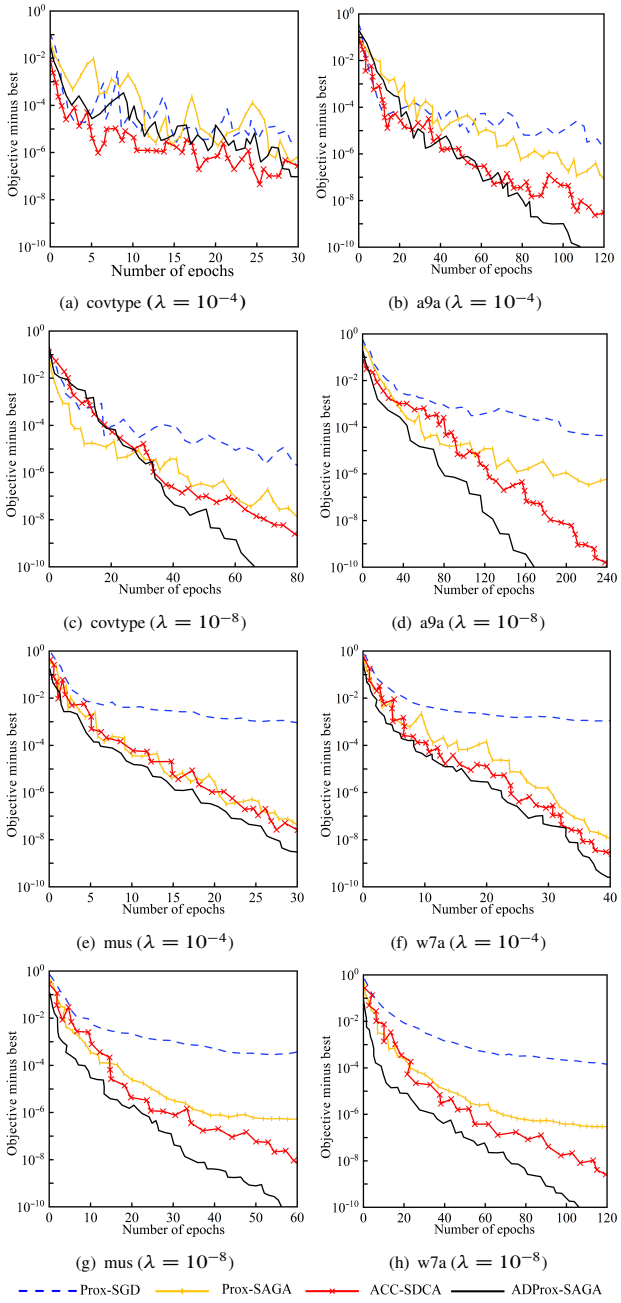


Fig. 1 Performance of algorithms.

SAGA, and ACC-SDCA with their theoretical parameter settings for comparison. The reason for choosing ACC-SDCA is that it has fewer parameters to tune, so it seems more practical.

The results are shown in Fig. 1. In fact, ADProx-SAGA has an excellent performance from the dataset. ADProx-SAGA is significantly faster than the general proximal algorithms Prox-SGD and Prox-SAGA, and has similar convergence results compared to ACC-SDCA. For ill-conditioned problem, the accelerated algorithm is significantly faster than the non-accelerated algorithm in Fig. 1 when λ gets smaller, the fast convergence of ADProx-SAGA in practice may imply that the algorithm could potentially benefit more applications. Another observation is that ADProx-SAGA does not perform well in the first several iterations of datasets covtype and a9a. For ill-conditioned problem, we all know that accelerated algorithm can yield a faster convergence rate in theory, so we conjecture that this is because the objective is locally well-conditioned around the initial point. On the contrary, non-accelerated algorithms (Prox-SVRG and Prox-SAGA) usually have better performance in this case.

However, for ill-condition problem, although ADProx-SAGA is significantly faster in terms of convergence rate, but is somewhat unstable compared to other three algorithms, it can be understood that there is a certain chance in the random selection of the initial point

In the selection of the momentum factor β_k, for the performance of ADProx-SAGA on dataset, we set $\lambda = 10^{-8}$, and choose β_k as 0.2, 0.5, 0.8, $\frac{k-2}{k+1}$, separately. In order to reflect the fairness of the results, the numerical results are obtained as the average of three experiments. Their performance on the datasets is summarized as follows.

As can be seen from Table 1, $\beta_k = 0.5$ performs best when β_k is chosed as a constant value. We conjecture that the algorithm will have more iterations when $\beta_k = 0.2$, and when $\beta_k = 0.8$, the algorithm tends to miss the optimal value point during the iteration. When making $\beta_k = \frac{k-2}{k+1}$, its value increases gradually between 0 and 1 as the number of iterations k increases, and its dynamic change rule allows it to have a more forgiving adjustment strategy compared to a fixed value, which is the reason for its relatively fewer epochs.

Table 1 Number of epochs of different β_k.

Dataset	β_k			
	0.2	0.5	0.8	$\frac{k-2}{k+1}$
covtype	149	115	127	106
a9a	259	173	223	165
mushrooms	77	68	75	56
w7a	168	136	151	107

5 Conclusion

In this paper, we propose a variance reducing stochastic proximal method ADProx-SAGA which uses the Nesterov's momentum trick. During the iteration of ADProx-SAGA, we achieve a deep fusion of Point-SAGA and fast Douglas–Rachford splitting. Theoretical results show that ADProx-SAGA achieves an accelerated linear rate for strongly convex problems, and experimental results show its good performance in practice.

Appendix

A Proofs for some properties

In this section, we prove some simple bounds of proximal operator, which are useful in the following work. Define $g_f^\gamma(x) = \frac{1}{\gamma}(x - \text{prox}_f^\gamma(x))$, so $g_f^\gamma(x)$ is the sub-gradient of $f(x)$ at point $\text{prox}_f^\gamma(x)$, so by the first-order optimality condition, we can get $\text{prox}_f^\gamma(x) + \gamma g_f^\gamma(x) = x$.

Proposition A1 (Moreau decomposition) For any $x,\ y \in \mathbf{R}^d$, and any convex function $f : \mathbf{R}^d \to \mathbf{R}$ with Fenchel conjugate f^*, we have

$$\text{prox}_f^\gamma(x) = x - \gamma\,\text{prox}_{f^*}^{1/\gamma}(x/\gamma) \tag{A1}$$

Recall the defintion of $g_f^\gamma(x)$, after combing Eq. (A1), we have the following relation between the proximal operator of the conjugate f^* and $g_f^\gamma(x)$:

$$g^\gamma(x) = \frac{1}{\gamma}(x - \text{prox}_f^\gamma(x)) = \text{prox}_{f*}^\gamma(x/\gamma) \tag{A2}$$

Proof Let $u = \text{prox}_f^\gamma(x)$ and $v = \frac{1}{\gamma}(x - u)$. By taking the derivative of the proximal operator of $f(x)$, we can get $v \in \partial f(u)$, which follows by conjugacy of $f(x)$ that $u \in \partial f^*(v)$. Thus we interpret $v = \frac{1}{\gamma}(x - u)$ as the optimality condition of the proximal operator of f^*, we have

$$v = \frac{1}{\gamma}(x - u) = \text{prox}_{f^*}^{1/\gamma}(x/\gamma).$$

Further setting $u = \text{prox}_f^\gamma(x)$, we obtain the result. ■

Lemma A1 (Lower bounds of inner product) For any $x, y \in \mathbf{R}^d$, any convex function $f : \mathbf{R}^d \to \mathbf{R}$, and any regularization parameter $\gamma > 0$, we have

$$\langle g_f^\gamma(x) - g_f^\gamma(y), x - y\rangle \geqslant \gamma\|g_f^\gamma(x) - g_f^\gamma(y)\|^2 \tag{A3}$$

Further, if $f(x)$ is μ-strongly convex, we have

$$\langle g_f^\gamma(x) - g_f^\gamma(y), x - y\rangle \geqslant \frac{\mu}{\mu\gamma + 1}\|x - y\|^2 \tag{A4}$$

Proposition A2 (Firm non-expansiveness) For

any $x, y \in \mathbf{R}^d$, any convex function $f: \mathbf{R}^d \to \mathbf{R}$ with strong convexity constant $\mu \geqslant 0$, and any regularization parameter $\gamma > 0$, we have the firm non-expansiveness of $\text{prox}_f^\gamma(x)$ in the following:

$$\langle \text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y), x - y\rangle \geqslant (1+\mu\gamma)\|\text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y)\|^2 \quad \text{(A5)}$$

From the above result we can also deduce that

$$\|2\,\text{prox}_f^\gamma(x) - x - (2\,\text{prox}_f^\gamma(y) - y)\| \leqslant \|x - y\| \quad \text{(A6)}$$

Proof It follows the strong convexity of loss function $f(x)$, we apply $\langle \nabla f(x) - \nabla f(y), x - y\rangle \geqslant \mu\|x-y\|^2$ at the (sub-) gradients $g_f^\gamma(x)$ and $g_f^\gamma(y)$, and their corresponding points $\text{prox}_f^\gamma(x)$ and $\text{prox}_f^\gamma(y)$ are as follows:

$$\langle g_f^\gamma(x) - g_f^\gamma(y), \text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y)\rangle \geqslant \mu\|\text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y)\|^2 \quad \text{(A7)}$$

Multiply both sides of Formula (A7) by γ and add $\|\text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y)\|^2$ to each afterwards, we have

$$\langle \text{prox}_f^\gamma(x) + \gamma g_f^\gamma(x) - \text{prox}_f^\gamma(y) - \gamma g_f^\gamma(y), x - y\rangle \geqslant (1+\mu\gamma)\|\text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y)\|^2,$$

According to the optimality condition $\text{prox}_f^\gamma(x) + \gamma g_f^\gamma(x) = x$, the bound is confirmed.

For Formula (A6), simply substituting $g_f^\gamma(x) = \frac{1}{\gamma}(x - \text{prox}_f^\gamma(x))$ into Formula (A3), we also can deduce $\|\text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y)\| \leqslant \|x - y\|$ from Formula (A5), then add the smaller $\|\text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y)\|$ and minus the larger $\|x - y\|$ to the left of inequality above at the same time, Formula (A6) can be obtained. ■

B Proofs of Theorems 2 and 3

Theorem A1 (Non-strongly convex case) Suppose that each $f_i: \mathbf{R}^d \to \mathbf{R}$ is convex and L-smooth, while the regularization term $h(x)$ is convex. Then for Algorithm 1 with step size $\gamma \leqslant 1/L$, at Step k, we have

$$E\|\bar{g}_j^k - g_j^*\|^2 \leqslant \frac{1}{k}\Big(\sum_{i=1}^n \|g_i^0 - g_i^*\|^2 + \|\frac{1}{\gamma}(u^0 - u^*)\|^2\Big) \quad \text{(A8)}$$

where $\bar{g}_j^k = \frac{1}{k}\sum_{t=1}^k g_j^t$, and the expectation of $\|\bar{g}_j^k - g_j^*\|^2$ for all choices of index j at Step k.

Proof

We adjust the upper bound of $E[T^{k+1}]$ and recall the definition of $v_j^k = z_j^k + x^k - y^k$, then we learn about $v_j^* = z_j^* + x^* - y^*$. Combining the definition of z_j^k in Algorithm 1 and deduced z_j^*, so v_j^k can be formulated as $v_j^k = 2x^k - y^k + \gamma(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k)$.

We bound the expectation of as

$$\begin{aligned}
&E\|v_j^k - v_j^*\|^2 = \\
&E\|2x^k - y^k - (2x^* - y^*) + \\
&\gamma(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k - g_j^* + g^*)\|^2 = \\
&\|2x^k - y^k - (2x^* - y^*)\|^2 + \\
&E\|\gamma(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k - g_j^* + g^*)\|^2 \leqslant \\
&\|u^k - u^*\|^2 + \\
&E\|\gamma(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k - g_j^* + g^*)\|^2 \leqslant \\
&\|u^k - u^*\|^2 + E\|\gamma(g_j^k - g_j^*)\|^2 \qquad \text{(A9)}
\end{aligned}$$

In the first inequality of Formula (A9), since $x^k = \text{prox}_h^\gamma(u^k)$, we have $2x^k - y^k = 2\,\text{prox}_f^\gamma(u^k) - y^k$. Applying Formula (A6) , we can get the first upper bound. The right-hand side of the second inequatily of Formula (A9), $\|u^k - u^*\|^2 + E\|\gamma(g_j^k - g_j^*)\|^2$, can also be obtained by using the variance formula $E[(X - E[X])^2] = E[X^2] - E[X]^2 \leqslant E[X^2]$, where $E(g_j^k - g_j^*) = \frac{1}{n}\sum_{i=1}^n g_i^k - g^*$.

According to the definition of g_j^{k+1}, which is the gradient mapping at v_j^k, we further bound the inner product $-\langle v_j^k - v_j^*, \gamma(g_j^{k+1} - g_j^*)\rangle$ by Formula (11),

$$\begin{aligned}
&-E\langle v_j^k - v_j^*, \gamma(g_j^{k+1} - g_j^*)\rangle \leqslant \\
&-(1 + \frac{1}{L\gamma})E\|\gamma(g_j^{k+1} - g_j^*)\|^2 \qquad \text{(A10)}
\end{aligned}$$

Substituting Formulas (A9) and (A10) into Formula (14), then simplifying for $E[T^{k+1}]$ yields

$$\begin{aligned}
&E[T^{k+1}] \leqslant (1+\beta_k^2)T^k + [(1+\beta_k)^2 - \frac{c}{n}]\times \\
&\frac{1}{n}\sum_{i=1}^n \|\gamma(g_i^k - g_i^*)\|^2 + (\frac{c}{n} - \frac{2(1+\beta_k)^2}{L\gamma} + 1)\times \\
&E\|\gamma(g_j^{k+1} - g_j^*)\|^2 - E\|\gamma(g_j^{k+1} - g_j^*)\|^2 \qquad \text{(A11)}
\end{aligned}$$

In particular, we chose $c = n(1+\beta_k)^2$ and $\gamma \leqslant 1/L$ to ensure that parameters $[(1+\beta_k)^2 - \frac{c}{n}]$ and $[\frac{c}{n} - \frac{2(1+\beta_k)^2}{L\gamma} + 1]$ are non-positive, so we can get

$$E[T^{k+1}] \leqslant (1+\beta_k^2)T^k - E\|\gamma(g_j^{k+1} - g_j^*)\|^2 \quad \text{(A12)}$$

Taking the expectation for both sides of Formula (A12) from 0 to k, we have

$$\begin{aligned}
\sum_{t=1}^k E\|\gamma(g_j^t - g_j^*)\|^2 \leqslant &[(1+\beta_k^2)T^0 + \\
&\beta_k^2(T^1 + T^2 + \cdots + T^{k-1})] - E[T^k],
\end{aligned}$$

where $\bar{g}_j^k = \frac{1}{k}\sum_{t=1}^{k} g_j^t$. Discarding the non-negative term $E[T^k]$ and simplifying with Jensen's inequality $\sum_{t=1}^{k} E\|(g_j^t - g_j^*)\|^2 \geqslant kE\|\bar{g}_j^k - g_j^*\|^2$ and $\beta_k = 0$, then we have

$$E\|\bar{g}_j^k - g_j^*\|^2 \leqslant \frac{1}{\gamma^2 \cdot k} T^0.$$

Substituting $c = n(1+\beta_k)^2$ into T^0, Formula (A8) can be proved. ■

Theorem A2 (Strongly convex case) Supposed that each $f_i : \mathbf{R}^d \to \mathbf{R}$ is μ-strongly convex and L-smooth, while the regularization term $h(x)$ is convex. Then for Algorithm 1 with step size

$$\gamma = \min\left\{\frac{1}{(1+\beta_k)^2\mu n}, \frac{\sqrt{9(1+\beta_k^2)^2L^2+3(1-\beta_k)^2(1+\beta_k^2)}-3(1+\beta_k^2)L}{2(1-\beta_k)^2\mu L}\right\},$$

at Step k, we have

$$E\|x^k - x^*\|^2 \leqslant [1+\beta_k^2-(1+\beta_k^2)\frac{\mu\gamma}{2(\mu\gamma+1)}]^k \times \frac{(1-\beta_k)^2\mu\gamma + 2(1+\beta_k^2)}{2-(1+\beta_k)^2 n\mu\gamma} \times \left\{\sum_{i=1}^{n}\|\gamma(g_i^0 - g_i^*)\|^2 + \|u^0 - u^*\|^2\right\} \quad \text{(A13)}$$

Proof We present an upper bound for $E[T^{k+1}]$ which is different from Theorem 1. Recalling g^{k+1} and g_j^* are the gradient mappings of f_j at v_j^k and v_j^*, respectively, applying Formula (A4) in Lemma A1, the inner product holds,

$$-\frac{1}{2}\langle\gamma(g_j^{k+1} - g_j^*), v_j^k - v_j^*\rangle \leqslant -\frac{\mu\gamma}{2(1+\mu\gamma)}\|v_j^k - v_j^*\|^2 \quad \text{(A14)}$$

Substituting Formula (A14) into Formula (14), and combining with the upper bound of $E\|v_j^k - v_j^*\|^2$ given by Formula (A9), we have

$$E[T^{k+1}] \leqslant \Big[1+\beta_k^2-(1+\beta_k)^2\frac{\mu\gamma}{2(\mu\gamma+1)}\Big]T^k + \frac{1}{2}\Big(\frac{(1+\beta_k)^2\mu\gamma}{\mu\gamma+1}c - \frac{2c}{n} + \frac{(1-\beta_k)^2\mu\gamma+2(1+\beta_k^2)}{\mu\gamma+1}\Big)\times \frac{1}{n}\sum_{i=1}^{n}\|\gamma(g_i^k - g_i^*)\|^2 + \Big[\frac{1}{2}\Big(1-\frac{3}{L\gamma}\Big)(1+\beta_k)^2 + \frac{c}{n}\Big]\times E\|\gamma(g_j^{k+1} - g_j^*)\|^2 \quad \text{(A15)}$$

In particular, we chose $c = \frac{(1-\beta_k)^2\mu\gamma+2(1+\beta_k^2)}{2/n-(1+\beta_k)^2\mu\gamma}$ and $\gamma = \min\left\{\frac{1}{(1+\beta_k)^2\mu n}, \frac{\sqrt{9(1+\beta_k^2)^2L^2+3(1-\beta_k)^2(1+\beta_k^2)}-3(1+\beta_k^2)L}{2(1-\beta_k)^2\mu L}\right\}$ to ensure that the parameters of the terms $\Big[\frac{1}{2}\Big(\frac{(1+\beta_k)^2\mu\gamma}{\mu\gamma+1}c - \frac{2c}{n} + \frac{(1-\beta_k)^2\mu\gamma+2(1+\beta_k^2)}{\mu\gamma+1}\Big)\Big]$ and $\Big[\frac{1}{2}\Big(1-\frac{3}{L\gamma}\Big)\Big(1+\beta_k)^2 + \frac{c}{n}\Big]$ in Formula (A15) are positive, discarding the non-negative term and taking the expectation for the rest terms of Formula (A15), so we can get

$$E[T^{k+1}] \leqslant (1+\beta_k^2-(1+\beta_k)^2\frac{\mu\gamma}{2(\mu\gamma+1)})E[T^k].$$

Further summing from 0 to k yields

$$E[T^k] \leqslant (1+\beta_k^2-(1+\beta_k)^2\frac{\mu\gamma}{2(\mu\gamma+1)})^k \times T^0.$$

We learn about that $x_k - x^* = \mathrm{prox}_h^\gamma(u^k) - \mathrm{prox}_h^\gamma(u^*)$ since $x^k = \mathrm{prox}_h^\gamma(u^k)$ and $x^* = \mathrm{prox}_h^\gamma(u^*)$. Due to the firm non-expansiveness, we have

$$E\left\|x^k - x^*\right\|^2 \leqslant E\left\|u^k - u^*\right\|^2 \leqslant E\left[T^k\right] \leqslant \left(1+\beta_k^2-(1+\beta_k)^2\frac{\mu\gamma}{2(\mu\gamma+1)}\right)^k \times T^0 \quad \text{(A16)}$$

Substituting $c = \dfrac{(1-\beta_k)^2\mu\gamma+2(1+\beta_k^2)}{2/n-(1+\beta_k)^2\mu\gamma}$ into T^0, Formula (A13) can be proved. ■

References

[1] Z. Yuan, Y. Lu, and Y. Xue, DroidDetector: Android malware characterization and detection using deep learning, *Tsinghua Science and Technology*, vol. 21, no. 1, pp. 114–123, 2016.

[2] Y. Sun, Z. Dou, Y. Li, and S. Wang, Improving semantic part features for person re-identification with supervised non-local similarity, *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 636–646, 2020.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[4] H. Robbins and S. Monro, A stochastic approximation method, *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.

[5] R. Johnson and T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in *Proc. 26th Int. Conf. Neural Information Processing Systems*, Lake Tahoe, NV, USA, 2013, pp. 315–323.

[6] A. Dedazio, F. Bach, and S. Lacoste-Julien, SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives, in *Proc. 27th Int. Conf. Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 1646–1654.

[7] O. Shamir and T. Zhang, Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes, in *Proc. 30th Int. Conf. Machine Learning*, Atlanta, GA, USA, 2013, pp. 71–79.

[8] L. Xiao and T. Zhang, A proximal stochastic gradient method with progressive variance reduction, *SIAM J. Optim.*, vol. 24, no. 4, pp. 2057–2075, 2014.

[9] S. Shalev-Shwartz and T. Zhang, Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization, in *Proc. 31st Int. Conf. Machine Learning*, Beijing, China, 2014, pp. I-64–I-72.

[10] A. Defazio, A simple practical accelerated method for finite sums, in *Proc. 30th Int. Conf. Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 676–684.

[11] H. Lin, J. Mairal, and Z. Harchaoui, Catalyst acceleration for first-order convex optimization: From theory to practice, *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 7854–7907, 2017.

[12] Z. Allen-Zhu, Katyusha: The first direct acceleration of stochastic gradient methods, *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 8194–8244, 2017.

[13] K. Zhou, F. Shang, and J. Cheng, A simple stochastic variance reduced algorithm with fast convergence rates, in *Proc. 35th Int. Conf. Machine Learning*, Stockholm, Sweden, 2018, pp. 5980–5989.

[14] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. New York, NY, USA: Springer, 2004.

[15] A. Chambolle and C. H. Dossal, On the convergence of the iterates of "FISTA", *J. Optim. Theory Appl.*, vol. 166, no. 3, p. 25, 2015.

[16] J. Liu, L. Xu, S. Shen, and Q. Ling, An accelerated variance reducing stochastic method with Douglas-Rachford splitting, *Mach. Learn.*, vol. 108, no. 5, pp. 859–878, 2019.

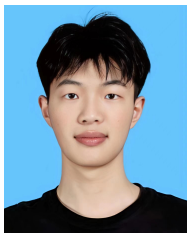
[17] P. Panagiotis, L. Stella, and A. Bemporad, Douglas-Rachford splitting: Complexity estimates and accelerated variants, in *Proc. 53rd IEEE Conf. Decision and Control*, Los Angeles, CA, USA, 2014, pp. 4234–4239.

[18] C. Lemaréchal and C. Sagastizábal, Practical aspects of the Moreau–Yosida regularization: Theoretical preliminaries, *SIAM J. Optim.*, vol. 7, no. 2, pp. 367–385, 1997.

[19] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams, Variance reduced stochastic gradient descent with neighbors, in *Proc. 28th Int. Conf. Neural Information Processing Systems*, Montreal, Canada, 2015, pp. 2305–2313.

[20] H. Luo, X. Bai, G. Lim, and J. Peng, New global algorithms for quadratic programming with a few negative eigenvalues based on alternative direction method and convex relaxation, *Math. Prog. Comp.*, vol. 11, no. 1, pp. 119–171, 2019.

[21] H. Luo, X. Ding, J. Peng, R. Jiang, and D. Li, Complexity results and effective algorithms for worst-case linear optimization under uncertainties, *Informs J. Comput.*, vol. 33, no. 1, pp. 180–197, 2021.



Jialin Lei received the BEng degree from Henan Institute of Science and Technology, China in 2020. He is currently a master student at Zhejiang Normal University, China. His research interests include machine learning and optimization algorithm.

Zhao Zhang received the PhD degree in applied mathematics from Xinjiang University, China in 2003. She is currently a distinguished professor at Zhejiang Normal University, China. Her research interests include approximation algorithm, combinatorial optimization, and machine learning. She has published more than 180 academic papers.

Ying Zhang received the PhD degree in operations research and cybernetics from Shanghai University, China in 2009. She is currently an associate professor at Zhejiang Normal University, China. Her research interests include optimal control, global optimization, and machine learning. She has published some papers in international journals, such as *Applied Mathematical Modelling*, *Journal of Computational and Applied Mathematics*, and *Communications in Nonlinear Science and Numerical Simulation*.