Machine Learning
https://doi.org/10.1007/s10994-019-05785-3

An accelerated variance reducing stochastic method with Douglas-Rachford splitting

Jingchang Liu[1] · Linli Xu[1] · Shuheng Shen[1] · Qing Ling[2]

Received: 21 April 2018 / Accepted: 9 January 2019

Abstract
We consider the problem of minimizing the regularized empirical risk function which is represented as the average of a large number of convex loss functions plus a possibly non-smooth convex regularization term. In this paper, we propose a fast variance reducing (VR) stochastic method called Prox2-SAGA. Different from traditional VR stochastic methods, Prox2-SAGA replaces the stochastic gradient of the loss function with the corresponding gradient mapping. In addition, Prox2-SAGA also computes the gradient mapping of the regularization term. These two gradient mappings constitute a Douglas-Rachford splitting step. For strongly convex and smooth loss functions, we prove that Prox2-SAGA can achieve a linear convergence rate comparable to other accelerated VR stochastic methods. In addition, Prox2-SAGA is more practical as it involves only the stepsize to tune. When each loss function is smooth but non-strongly convex, we prove a convergence rate of $\mathcal{O}(1/k)$ for the proposed Prox2-SAGA method, where k is the number of iterations. Moreover, experiments show that Prox2-SAGA is valid for non-smooth loss functions, and for strongly convex and smooth loss functions, Prox2-SAGA is prominently faster when loss functions are ill-conditioned.

Keywords Variance reduction (VR) · Acceleration · Douglas-Rachford splitting · Proximal operator · Gradient mapping

Editors: Masashi Sugiyama, Yung-Kyun Noh.

✉ Linli Xu
linlixu@ustc.edu.cn

Jingchang Liu
xdjcl@mail.ustc.edu.cn

Shuheng Shen
vaip@mail.ustc.edu.cn

Qing Ling
lingqing556@mail.sysu.edu.cn

[1] University of Science and Technology of China, Hefei, China

[2] Sun Yat-Sen University, Guangzhou, China

Published online: 01 May 2019

1 Introduction

In many artificial intelligence and machine learning applications, one needs to solve the following generic optimization problem in the form of regularized empirical risk minimization (Hastie et al. 2009)

$$\min_{x\in\mathbb{R}^d} f(x) + h(x). \tag{1}$$

Given n samples, f is the average of a set of convex loss functions

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{2}$$

where $f_i : \mathbb{R}^d \to \mathbb{R}$ denotes the empirical loss of the i-th sample with regard to the parameters x, and h is the regularization term, which is convex but possibly non-smooth. The goal is to find the optimal solution of x that minimizes the regularized empirical loss over the whole dataset.

Numerous efforts have been devoted to solve this problem (Bottou et al. 2016; Johnson and Zhang 2013; Defazio et al. 2014; Shalev-Shwartz and Zhang 2013). When h is absent, stochastic gradient descent (SGD) (Robbins and Monro 1951) is a standard and effective method to solve (1), especially when the number of samples is very large. Specifically, stochastic gradient is utilized in SGD to update x in each step instead of calculating the full gradient, which yields lower per iteration cost. However, as a side effect, a rather large variance introduced by the stochastic gradient will slow down the convergence (Bottou et al. 2016).

To address the issue, a number of variance reducing (VR) stochastic methods have been developed in recent years, such as SVRG (Johnson and Zhang 2013), SAGA (Defazio et al. 2014) and SDCA (Shamir and Zhang 2013). As a key feature of the VR stochastic methods, the variance of the stochastic gradient goes to zero asymptotically along the iterative updates. Therefore, unlike SGD which needs a decaying step size to guarantee convergence, the step size can be fixed for these methods. As a result, the convergence rate can be improved from sub-linear in SGD to linear in the VR stochastic methods. Further, for the problem with the non-smooth regularization term h, a proximal operator of h is introduced at the end of each iteration of the VR stochastic methods, for example, Prox-SVRG (Xiao and Zhang 2014), Prox-SAGA (Defazio et al. 2014) and Prox-SDCA (Shalev-Shwartz and Zhang 2014). In addition, acceleration techniques such as Acc-SDCA (Shalev-Shwartz and Zhang 2014), Catalyst (Lin et al. 2015, 2017) and Katyusha (Allen-Zhu 2017), can boost these methods to faster convergence rates when the loss function is ill-conditioned. However, existing accelerated VR stochastic methods often involve mulitple parameters to tune, which brings difficulties to their implementations.

In this paper, we develop a simple accelerated VR stochastic method, named as Prox2-SAGA, to solve (1). Similar to most non-accelerated algorithms, Prox2-SAGA only has one parameter, the step size, to tune, and is hence easy to implement. Different from most stochastic algorithms which utilize the gradients of f_i, Prox2-SAGA uses the corresponding gradient mappings, through applying the proximal operator on each f_i. It is the proximal operator that enables Prox2-SAGA to achieve the accelerated rate when the loss functions are ill-conditioned. Prox2-SAGA can be regarded as the generalization of Point-SAGA (Defazio 2016) which considers a special case when the non-smooth regularizer h is absent. To handle h, Prox2-SAGA employs another proximal operator. The two proximal operators in one

iteration of Prox2-SAGA essentially constitute a Douglas-Rachford splitting step. Our main contributions are listed below:

- We design Prox2-SAGA, a fast and simple VR stochastic method, to solve (1) with Douglas-Rachford splitting.
- When loss functions f_i are convex and smooth, we prove that Prox2-SAGA can achieve a $\mathcal{O}(1/k)$ convergence rate, where k is the number of iterations. Further when f_i's are strongly convex, we prove that Prox2-SAGA converges with an accelerated linear rate.
- Experiments are conducted to demonstrate the efficacy of the proposed algorithm, no matter whether the loss functions f_i are smooth or not, in particular when the loss functions f_i are ill-conditioned.

2 Definitions and assumptions

In this section, we introduce definitions and assumptions used in this paper.

2.1 Definitions

For a function f, the proximal operator at point x with step size $\gamma > 0$ is defined as

$$\text{prox}_f^\gamma(x) = \underset{y\in\mathbb{R}^d}{\text{argmin}}\Big(f(y) + \frac{1}{2\gamma}\|y - x\|^2\Big). \tag{3}$$

For many functions f of interest, the proximal operator prox_f^γ has a closed form solution or can be computed efficiently (Parikh and Boyd 2014).

Further, we define

$$\phi_f^\gamma(x) = \frac{1}{\gamma}(x - \text{prox}_f^\gamma(x)) \tag{4}$$

as the gradient mapping of f at point x with $\gamma > 0$. According to the definition of the proximal operator in (3), $\phi_f^\gamma(x)$ is a subgradient of f at $\text{prox}_f^\gamma(x)$.

The subdifferential is introduced to facilitate the analysis of non-smoothness. The subdifferential $\partial f(x)$ of f at x is the set of all subgradient

$$\partial f(x) = \big\{g \mid g^T(y - x) \le f(y) - f(x), \forall y \in \text{dom}\, f\big\}.$$

Besides, the conjugate of a function f is defined as

$$f^*(y) = \sup_{x\in\text{dom}\, f}(y^T x - f(x)).$$

2.2 Assumptions

In this paper, we may assume that each f_i is μ-strongly convex, namely, for any $x, y \in \mathbb{R}^d$ and any subgradient g_i of f_i at x, it holds that

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

where $\mu > 0$. The assumption can be easily satisfied by refining f_i with a strongly-convex regularizer. For a general convex function, the above inequality always holds with $\mu = 0$.

We may also assume that each f_i is L-smooth, namely, for any $x, y \in \mathbb{R}^d$, it holds that

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

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

3 Related work

This section gives an overview of VR stochastic methods. In particular, we emphasize the accelertion techniques for ill-conditioned problems.

3.1 Variance reducing stochastic methods

To effectively reduce the variance of stochastic gradient in stochastic optimization, several statistical VR methods, such as importance sampling and stratified sampling (Owen 2013; Ross 2013), have been introduced. Although utilizing the internal structure of dataset to proceed importance sampling or stratified sampling, as considered in Zhao and Zhang (2014), (2015) and Needell et al. (2014), works quite well, it cannot asymptotically reduce the variance to zero.

Meanwhile, some other methods which employ control variates (Owen 2013, Chapter 8.9) have been considered in Johnson and Zhang (2013), Defazio et al. (2014), Shamir and Zhang (2013), Xiao and Zhang (2014) and Schmidt et al. (2017). SAGA (Defazio et al. 2014) and SVRG (Johnson and Zhang 2013) are two typical algorithms among them, which utilize the following VR stochastic gradient

$$\nabla f_j(x^k) - \nabla f_j(\tilde{x}) + \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{x}), \tag{5}$$

where $\tilde{x}$ is the saved "snapshot" of a previous x, to replace $\nabla f_j(x^k)$ in SGD. In SAGA and SVRG, $\nabla f_j(\tilde{x})$ can be regarded as the control variate of $\nabla f_j(x^k)$. The variance of the VR stochastic gradient goes to zero asymptotically along the iterative updates as $\nabla f_j(x^k)$ and $\nabla f_j(\tilde{x})$ become closer in expectation. This leads to a much faster convergence rate than that of SGD.

3.2 Acceleration for ill-conditioned problems

For an L-smooth and μ-strongly convex function, L/μ is known as its condition number and we call a function ill-conditioned when L/μ is too large. Many gradient-based methods may perform poorly in handling ill-conditioned functions. Fortunately, the convergence rate can be boosted by some acceleration techniques. Specifically, for (1) where each f_i is L-smooth and

μ-strongly convex while h is convex but possibly non-smooth, most VR stochastic methods such as Prox-SDCA, Prox-SAGA and Prox-SVRG require $\mathcal{O}\left((n + L/\mu)\log(1/\epsilon)\right)$ steps to achieve an ϵ-accurate solution. Nevertheless, if we apply some acceleration techniques, the numbers of iterations needed are $\mathcal{O}((n + \sqrt{nL/\mu})\log(L/\mu)\log(1/\epsilon))$ in Catalyst (Woodworth and Srebro 2016) and $\mathcal{O}((n + \sqrt{nL/\mu})\log(1/\epsilon))$ in Acc-SDCA (Shalev-Shwartz and Zhang 2014) and Katyusha (Allen-Zhu 2017). As a result, these accelerated methods will be significantly faster than the non-accelerated ones when $L/\mu \gg n$. In this paper, we shall show that our algorithm can also achieve an accelerated rate $\mathcal{O}((n + \sqrt{nL/\mu})\log(1/\epsilon))$.

4 Algorithm

The proposed algorithm is outlined in Algorithm 1. It maintains four sequences, x^k, y^k, g_j^k and z_j^k, where j stands for the j-th loss function. The starting point x^0 is set arbitrarily. Each g_i^0 can be chosen as any gradient/subgradient of f_i at x^0. The algorithm has only one parameter, the step size γ. In the k-th iteration, a loss function f_j is randomly chosen. Each g_j is updated from g_j^k to g_j^{k+1} (see (9)) and x is updated from x^k to x^{k+1} (see (11)), while z_j^k and y^k can be regarded as the intermediate variables for the updates of x^k and g_j^k. According to the definition of z_j^k in (8) and update of y^{k+1} in (10), the main steps can be written as

$$y^{k+1} = x^k - \gamma\left(g_j^{k+1} - g_j^k + \frac{1}{n}\sum_{i=1}^{n} g_i^k\right), \tag{6}$$

$$x^{k+1} = \mathrm{prox}_h^{\gamma}(y^k), \tag{7}$$

where g_j^{k+1} is the gradient mapping of f_j at $z_j^k + x^k - y^k$.

In every iteration of our algorithm, we make use of the proximal operator of f_i to calculate the gradient mapping, in addition to the proximal operator of h. This setting enables the proposed algorithm to achieve the accelerated rate when the loss functions f_i's are ill-conditioned. The main iteration steps in our algorithm are similar to those in Prox-SAGA, which, however, contains only one proximal operator to handle the non-smoothness of h. In this sense, we name our algorithm as Prox2-SAGA.

To be specific, the main difference between Prox2-SAGA and Prox-SAGA is the definition of g_j. In Prox2-SAGA, g_j^{k+1} is a subgradient of f_j at point $\mathrm{prox}_{f_j}^{\gamma}(z_j^k + x^k - y^k)$, while in Prox-SAGA g_j^{k+1} is the gradient of f_j at x^k. From (9) and (10), it holds that $\mathrm{prox}_{f_j}^{\gamma}(z_j^k + x^k - y^k) = y^{k+1} + x^k - y^k$. That is to say, $\mathrm{prox}_{f_j}^{\gamma}(z_j^k + x^k - y^k)$ involves the "future" point y^{k+1}, which is analogous to the update in Point-SAGA (Defazio 2016). Therefore, compared to Prox-SAGA, our algorithm would achieve a faster convergence rate.

Algorithm 1 Prox2-SAGA

Input: $x^0 \in \mathbb{R}^d$, g_i^0 $(i = 1, 2, \ldots, n)$, step size $\gamma > 0$.
for $k = 0, 1, \ldots$ **do**
  Uniformly randomly pick j from 1 to n.
  Calculate g_j^{k+1}:

$$z_j^k = x^k + \gamma\Big(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k\Big), \tag{8}$$

$$g_j^{k+1} = \frac{1}{\gamma}\big((z_j^k + x^k - y^k) - \mathrm{prox}_{f_j}^{\gamma}(z_j^k + x^k - y^k)\big). \tag{9}$$

  Update x:

$$y^{k+1} = z_j^k - \gamma g_j^{k+1}, \tag{10}$$

$$x^{k+1} = \mathrm{prox}_h^{\gamma}(y^{k+1}). \tag{11}$$

  Update g_i $(i = 1, 2, \ldots, n)$ in the table:

$$g_i^{k+1} = \begin{cases} g_j^{k+1}, & \text{if } i = j, \\ g_i^k, & \text{otherwise.} \end{cases} \tag{12}$$

end for
Output: x^{k+1}.

Like Prox-SAGA, we maintain a table of g_i and update one element of the table in each iteration. The sum of gradient mappings $\sum_{i=1}^n g_i/n$ used in calculating z_j^k can be cached and updated efficiently at each iteration by $\sum_{i=1}^n g_i^{k+1}/n = \sum_{i=1}^n g_i^k/n + (g_j^{k+1} - g_j^k)/n$. Besides, for linearly parameterized models where $f_i(x)$ can be represented as the more structured form $\psi_i(a_i^T x)$, following the routine of SAGA, we just need to store a single real value instead of a full vector for each g_i. Linear regression and binary classification with logistic or hinge losses both fall in this regime.

5 Connection with other methods

In this section, we show that Prox2-SAGA is essentially a Douglas-Rachford splitting algorithm, and is a generalization of Point-SAGA. Further, we also establish the relations between Prox2-SAGA and Prox-SDCA.

5.1 Connection with Douglas-Rachford splitting

When $n = 1$, since $g_j^k = \sum_{i=1}^n g_i^k/n$ in Prox2-SAGA, the main iterations can be simplified to

$$y^{k+1} = -x^k + y^k + \mathrm{prox}_f^{\gamma}(2x^k - y^k),$$
$$x^{k+1} = \mathrm{prox}_h^{\gamma}(y^{k+1}).$$

These are the iterations of Douglas-Rachford splitting to minimize the composite cost function $f(x) + h(x)$ (Eckstein and Bertsekas 1992; Bauschke and Combettes 2017). In this sense, Prox2-SAGA is essentially a Douglas-Rachford splitting method, but aiming at solving the regularized empirical risk minimization problem when the number of samples n is larger than 1.

5.2 Generalization of point-SAGA

When $h = 0$, we have $x^k = y^k$ for Prox2-SAGA, and the main iterations can be simplified to

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

These are the iterations of Point-SAGA. Compared to Point-SAGA, Prox2-SAGA employs another proximal operator of h and uses Douglas-Rachford splitting to combine two proximal operators. Point-SAGA has been proven to have a $\mathcal{O}(1/k)$ convergence rate for non-smooth but strongly convex problems, and achieve an accelerated rate when each f_i is smooth and strongly convex. Some convergence properties can also be inherited by Prox2-SAGA.

5.3 Relation to Prox-SDCA

Different from other VR stochastic methods such as Prox-SAGA and Prox-SVRG, Prox-SDCA considers the dual problem of (1). In this section, we show that Prox-SDCA is connected to Prox2-SAGA in the sense that it also involves calculating of gradient mappings and proximal operators. However, they are essentially different since Prox2-SAGA handles functions in the primal domain, while Prox-SDCA works in the dual domain.

The Prox-SDCA algorithm has been considered in Shalev-Shwartz and Zhang (2014). In order to unify notations, we work with $f_i(x)$ rather than the more structured $\psi_i(a_i x)$. Then the dual objective to maximize is:

$$D(\alpha) = \frac{1}{n}\sum_{i=1}^n -f_i^*(-\alpha_i) - h^*\Big(\frac{1}{n}\sum_{i=1}^n \alpha_i\Big),$$

where f_i^*, h^* are the conjugate functions of f_i and h, respectively; α_i's are d-dimension dual variables.

Adopting Option I of Prox-SDCA in Figure 1 of Shalev-Shwartz and Zhang (2012), for the selected index j in step k, we can represent the update of α_j as

$$\alpha_j^{k+1} = \alpha_j^k + \underset{\Delta\alpha_j \in \mathbb{R}^d}{\mathrm{argmin}}\Big\{ f_j^*(-\alpha_j^k - \Delta\alpha_j) + \frac{n}{2}\big\|x^k + \frac{1}{n}\Delta\alpha_j\big\|^2\Big\},$$

which is equivalent to

$$\alpha_j^{k+1} = \underset{y}{\mathrm{argmin}}\Big\{ f_j^*(-y) + \frac{1}{2n}\big\|y - \alpha_j^k + nx^k\big\|^2\Big\}. \tag{13}$$

Obviously, this update involves the calculation of the proximal operator of f_j^*. The relation between the proximal operator of a function and its convex conjugate can be established by the extended Moreau decomposition (Parikh and Boyd 2014):

$$\text{prox}_{f_j^*}^{1/\gamma}(u/\gamma) = (u - \text{prox}_{f_j}^{\gamma}(u))/\gamma,$$

which shows that $\text{prox}_{f_j^*}^{1/\gamma}(u/\gamma)$ is identical to the gradient mapping of f_j at u. Therefore, the update of the dual variable α_j in (13) implies the calculation of the gradient mapping of f_j. It is the gradient mapping that allows Prox-SDCA to directly solve the problems with non-smooth loss functions. Next, we consider its update of the primal variable x.

The update of x in Prox-SDCA can be represented as

$$\begin{aligned} v^{k+1} &= v^k + \frac{1}{n}\Delta\alpha_j, \\ x^{k+1} &= \nabla h^*(v^{k+1}), \end{aligned} \tag{14}$$

where v is an auxiliary variable. As we optimize the strongly convex function, we can consider that there is an L_2 regularization in $h(x)$. We represent h as: $h(x) = \frac{\lambda_2}{2}\|x\|^2 + \lambda_1 r(x)$, where $r(x)$ is the non-smooth part. Then, the conjugate function of h is

$$h^*(v) = \max_{x\in\mathbb{R}^d}\left\{v^T x - \frac{\lambda_2}{2}\|x\|^2 - \lambda_1 r(x)\right\}.$$

Therefore, it follows that

$$\begin{aligned} \nabla h^*\left(v^{k+1}\right) &= \underset{x\in\mathbb{R}^d}{\text{argmax}}\left\{(v^{k+1})^T x - \frac{\lambda_2}{2}\|x\|^2 - \lambda_1 r(x)\right\} \\ &= \underset{x\in\mathbb{R}^d}{\text{argmin}}\left\{r(x) + \frac{1}{2}\frac{\lambda_2}{\lambda_1}\left\|x - \frac{v^{k+1}}{\lambda_2}\right\|^2\right\}, \end{aligned}$$

which is the proximal operator of r at $\frac{v^{k+1}}{\lambda_2}$. Thus, the update of x in (14) can be regarded as applying a proximal operator with step size $\frac{\lambda_1}{\lambda_2}$ on non-smooth r.

In conclusion, similar to Prox2-SAGA, Prox-SDCA involves computing the gradient mappings and proximal operators. Both (12) and (13) can be regarded as the calculation of the gradient mapping of f_i; the main difference is that in (13) the gradient mapping is calculated through the conjugate function of f_i, while in (12) the gradient mapping is calculated in the primal domain and by a rather straightforward way. Likewise, both (11) and (14) are the calculation of the proximal operator, except that (11) is more intuitive as it does not involve conjugate functions. Moreover, the gradient mapping in Prox2-SAGA involves the "future" point. Therefore, although both Prox2-SAGA and Prox-SDCA are able to converge linearly when each loss function is smooth and strongly convex, Prox2-SAGA would be faster than Prox-SDCA, as evidenced by the experiments.

6 Theory

In this section, we show that Prox2-SAGA converges to the optimal solution of (1) at a rate of $\mathcal{O}(1/k)$ when each f_i is smooth, and achieves an accelerated linear rate when each

f_i is further assumed to be strongly convex. We begin with several useful propositions and lemmas.

6.1 Preliminaries

Our analysis is built upon the theory of Moreau envelope (Lemaréchal and Sagastizábal 1997). The Moreau envelope of a continuous function $f : \mathbb{R}^d \to \mathbb{R}$ with a regularization parameter $\gamma > 0$ is defined as

$$f^\gamma(x) = \inf_y \left\{ f(y) + \frac{1}{2\gamma} \|x - y\|^2 \right\}. \tag{15}$$

The following proposition demonstrates the basic properties of Moreau envelope (Lemaréchal and Sagastizábal 1997).

Proposition 1 (Properties of Moreau envelope) *Given a convex continuous function $f : \mathbb{R}^d \to \mathbb{R}$ and a regularization parameter $\gamma > 0$, we consider its Moreau envelope f^γ defined in (15). Then*

1. *f^γ is continuously differentiable even when f is non-differentiable, and*

$$\nabla f^\gamma(x) = \frac{1}{\gamma}(x - \text{prox}_f^\gamma(x)). \tag{16}$$

   Moreover, f^γ is $\frac{1}{\gamma}$-smooth.
2. *If f is μ-strongly convex, then f^γ is $\frac{\mu}{\mu\gamma+1}$-strongly convex.*

From the definition of the gradient mapping $\phi_f^\gamma(x) = \frac{1}{\gamma}(x - \text{prox}_f^\gamma(x))$ in (4), we observe that $\nabla f^\gamma(x) = \phi_f^\gamma(x)$. According to the fact that f^γ is $\frac{1}{\gamma}$-smooth when f is convex and $\frac{\mu}{\mu\gamma+1}$-strongly convex when f is μ-strongly convex, we have the following lemma (Nesterov 2013).

Lemma 1 (Lower bounds of inner product) *For any $x, y \in \mathbb{R}^d$, any convex function $f : \mathbb{R}^d \to \mathbb{R}$ and any regularization parameter $\gamma > 0$, we have*

$$\left\langle \phi_f^\gamma(x) - \phi_f^\gamma(y), x - y \right\rangle \geq \gamma \|\phi_f^\gamma(x) - \phi_f^\gamma(y)\|^2. \tag{17}$$

Further, if f is strongly convex with constant $\mu > 0$, we have

$$\left\langle \phi_f^\gamma(x) - \phi_f^\gamma(y), x - y \right\rangle \geq \frac{\mu}{\mu\gamma + 1} \|x - y\|^2. \tag{18}$$

A direct corollary of Lemma 1 gives the following nonexpansiveness results, which are useful in the analysis.

Corollary 1 (Nonexpansiveness) *For any $x, y \in \mathbb{R}^d$, any convex function $f : \mathbb{R}^d \to \mathbb{R}$ and any regularization parameter $\gamma > 0$, we have the firm nonexpansiveness of $\text{prox}_f^\gamma(x)$, given by*

$$\|\text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y)\|^2 \leq \langle \text{prox}_f^\gamma(x) - \text{prox}_f^\gamma(y), x - y \rangle,$$

and the nonexpansiveness of $2\text{prox}_f^\gamma(x) - x$, given by

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

Proof The two inequalities follow from substituting $\phi_f^\gamma(x) = \frac{1}{\gamma}(x - \text{prox}_f^\gamma(x))$ into (17) in Lemma 1 and reorganizing terms. □

Lemma 1 gives lower bounds for the inner product $\langle \phi_f^\gamma(x) - \phi_f^\gamma(y), x - y \rangle$ when f is convex or strongly convex, no matter whether f is smooth or not. When f is convex and smooth, we can deduce another lower bound for the inner product.

Lemma 2 (Another lower bound of inner product) *For any $x, y \in \mathbb{R}^d$, any L-smooth function $f : \mathbb{R}^d \to \mathbb{R}$ and any regularization parameter $\gamma > 0$, we have*

$$\left\langle \phi_f^\gamma(x) - \phi_f^\gamma(y), x - y \right\rangle \geq \gamma \left(1 + \frac{1}{L\gamma}\right) \|\phi_f^\gamma(x) - \phi_f^\gamma(y)\|^2. \tag{19}$$

Proof Denote f^* as the conjugate function of f. Note that L-smoothness of f implies $\frac{1}{L}$-strong convexity of f^*. According to (17) and (18) in Lemma 1, we have

$$\left\langle \phi_{f^*}^\gamma(x) - \phi_{f^*}^\gamma(y), x - y \right\rangle \geq \frac{\frac{1}{L}}{2\frac{\gamma}{L} + 1} \|x - y\|^2 + \frac{\gamma(\frac{\gamma}{L} + 1)}{2\frac{\gamma}{L} + 1} \|\phi_{f^*}^\gamma(x) - \phi_{f^*}^\gamma(y)\|^2. \tag{20}$$

Recalling the extended Moreau decomposition (Parikh and Boyd 2014)

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

we have

$$\phi_{f^*}^\gamma(x) = \frac{1}{\gamma}\big(x - \text{prox}_{f^*}^\gamma(x)\big) = \text{prox}_f^{1/\gamma}(x/\gamma) = \frac{1}{\gamma}\big(x - \phi_f^{1/\gamma}(x/\gamma)\big). \tag{21}$$

Plugging $\phi_{f^*}^\gamma(x) = \frac{1}{\gamma}\big(x - \phi_f^{1/\gamma}(x/\gamma)\big)$ into (20) and simplifying the terms lead to (19). □

For the purpose of analysis, it is convenient to plug (9) into (10) to express Algorithm 1 in the form of

$$\begin{cases} y^{k+1} = -x^k + y^k + \text{prox}_{f_j}^\gamma(u_j^k), \\ g_j^{k+1} = \frac{1}{\gamma}\big(u_j^k - \text{prox}_{f_j}^\gamma(u_j^k)\big). \end{cases} \tag{22}$$

Here we define

$$u_j^k = z_j^k + x^k - y^k, \tag{23}$$

while $z_j^k = x^k + \gamma\Big(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k\Big)$, $x^k = \text{prox}_h^\gamma(y^k)$, as defined in (8) and (11), respectively. In these definitions, $j \in \{1, 2, \ldots, n\}$.

Before giving the main results, we show that the fixed point of the Prox2-SAGA iteration, if it exists, is exactly a minimizer of (1).

Proposition 2 *Suppose that $(y^\infty, \{g_i^\infty\}_{i=1,\ldots,n})$ is the fixed point of the Prox2-SAGA iteration (22). Then $x^\infty = \text{prox}_h^\gamma(y^\infty)$ is a minimizer of (1).*

Proof Define $z_j^\infty = x^\infty + \gamma\big(g_j^\infty - \frac{1}{n}\sum_{i=1}^n g_i^\infty\big)$. Since $(y^\infty, \{g_i^\infty\}_{i=1,\ldots,n})$ is the fixed point of (22), $y^\infty = -x^\infty + y^\infty + \text{prox}_{f_i}^\gamma(z_i^\infty + x^\infty - y^\infty)$, which implies

$$(z_i^\infty - y^\infty)/\gamma \in \partial f_i(x^\infty), \ \ i = 1, \ldots, n. \tag{24}$$

Meanwhile, because $x^\infty = \text{prox}_h^\gamma(y^\infty)$, we have

$$(y^\infty - x^\infty)/\gamma \in \partial h(x^\infty). \tag{25}$$

Observing that

$$\frac{1}{n}\sum_{i=1}^n (z_i^\infty - y^\infty) + (y^\infty - x^\infty) = \frac{1}{n}\sum_{i=1}^n z_i^\infty - x^\infty = 0,$$

from (24) and (25), we have $0 \in \partial f(x^\infty) + \partial h(x^\infty)$, meaning that x^∞ is a minimizer of (1). □

Denote x^* as a minimizer of (1). According to the first-order optimality condition of (1), there exist a set of subgradients g_j^*, one for each loss function f_j at x^*, and a subgradient $\partial h(x^*)$ for the regularization function h at x^*, such that $0 \in \frac{1}{n}\sum_{i=1}^n g_i^* + \partial h(x^*)$. Define

$$g^* = \frac{1}{n}\sum_{i=1}^n g_i^*,\ y^* = z_j^* - \gamma g_j^*,\ z_j^* = x^* + \gamma\left(g_j^* - g^*\right),\ u_j^* = z_j^* - x^* - y^*. \tag{26}$$

It is not difficult to verify from these definitions and the properties of the proximal operator that

$$g_j^* = \frac{1}{\gamma}\left(u_j^* - \text{prox}_{f_j}^\gamma(u_j^*)\right),\ x^* = \text{prox}_h^\gamma(y^*). \tag{27}$$

Throughout the analysis, all expectations are taken with respect to the choice of j at iteration k unless stated otherwise. Two particularly useful expectations are

$$\mathbb{E}[g_j^k] = \frac{1}{n}\sum_{i=1}^n g_i^k,\ \mathbb{E}[g_j^*] = g^*. \tag{28}$$

6.2 Main results

The proofs of the main results rely on a Lyapunov function, which at time $k+1$ is defined as

$$T^{k+1} = \frac{c}{n}\sum_{i=1}^n \left\|\gamma(g_i^{k+1} - g_i^*)\right\|^2 + \left\|y^{k+1} - y^*\right\|^2, \tag{29}$$

where $c > 0$ is a constant. We shall choose c as different values in the proofs of Theorems 1 and 2. The following lemma gives an upper bound for the expecation of the Lyapunov function.

Lemma 3 (Expectation of Lyapunov function) *Assume that each loss functions f_i is convex and L-smooth, while the regularization function h is convex. Then for Prox2-SAGA, at any time $k > 0$, the expectation of the Lyapunov function defined in (29) satisfies*

$$\begin{aligned}\mathbb{E}[T^{k+1}] \le{}& \left(\frac{1}{2} + \left(1 - \frac{1}{n}\right)c\right)\frac{1}{n}\sum_{i=1}^n \left\|\gamma(g_i^k - g_i^*)\right\|^2 + \left(2 + \frac{c}{n}\right)\mathbb{E}\left\|\gamma(g_j^{k+1} - g_j^*)\right\|^2 \\ &+ \frac{1}{2}\|y^k - y^*\|^2 + \frac{1}{2}\mathbb{E}\left\|u_j^k - u_j^*\right\|^2 - 2\mathbb{E}\langle u_j^k - u_j^*, \gamma(g_j^{k+1} - g_j^*)\rangle.\end{aligned} \tag{30}$$

Proof Taking expectation over the first term of T^{k+1}, we have

$$\frac{c}{n}\mathbb{E}\sum_{i=1}^{n}\left\|\gamma(g_i^{k+1}-g_i^*)\right\|^2=\left(1-\frac{1}{n}\right)\frac{c}{n}\sum_{i=1}^{n}\left\|\gamma(g_i^k-g_i^*)\right\|^2+\frac{c}{n}\mathbb{E}\left\|\gamma(g_j^{k+1}-g_j^*)\right\|^2. \tag{31}$$

To calculate the expectation for the second term of T^{k+1}, recall the definition of z_j^k in (8) and u_j^k in (23), we start by rewriting the iteration of y^{k+1} in (22) as

$$y^{k+1}=\frac{1}{2}\Big(y^k+\gamma\big(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k\big)\Big)+\frac{1}{2}\big(2\mathrm{prox}_{f_j}^{\gamma}(u_j^k)-u_j^k\big). \tag{32}$$

Rewriting y^* in the same way that $y^*=\frac{1}{2}\big(y^*+\gamma(g_j^*-g^*)\big)+\frac{1}{2}\big(2\mathrm{prox}_{f_j}^{\gamma}(u_j^*)-u_j^*\big)$, then by Young's inequality, we have

$$\begin{aligned}\mathbb{E}\left\|y^{k+1}-y^*\right\|^2&=\frac{1}{4}\mathbb{E}\Big\|y^k-y^*+\gamma\big(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*\big)\\&\quad+\big(2\mathrm{prox}_{f_j}^{\gamma}(u_j^k)-u_j^k\big)-\big(2\mathrm{prox}_{f_j}^{\gamma}(u_j^*)-u_j^*\big)\Big\|^2\\&\le\frac{1}{2}\mathbb{E}\Big\|y^k-y^*+\gamma\big(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*\big)\Big\|^2\\&\quad+\frac{1}{2}\mathbb{E}\Big\|\big(2\mathrm{prox}_{f_j}^{\gamma}(u_j^k)-u_j^k\big)-\big(2\mathrm{prox}_{f_j}^{\gamma}(u_j^*)-u_j^*\big)\Big\|^2.\end{aligned} \tag{33}$$

Because y^k-y^* is independent with the selection of j, and $\mathbb{E}(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*)=0$ according to (28), we have $\mathbb{E}\big\langle y^k-y^*,\gamma(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*)\big\rangle=0$. Then, for the first term at the right-hand side of (33), it holds

$$\begin{aligned}&\mathbb{E}\Big\|y^k-y^*+\gamma\big(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*\big)\Big\|^2\\&=\|y^k-y^*\|^2+\mathbb{E}\|\gamma\big(g_j^k-\frac{1}{n}\sum_{i=1}^{n}g_i^k-g_j^*+g^*\big)\|^2\\&\le\|y^k-y^*\|^2+\mathbb{E}\|\gamma\big(g_j^k-g_j^*\big)\|^2.\end{aligned} \tag{34}$$

The inequality comes from the variance formula $\mathbb{E}(X-\mathbb{E}X)^2\le\mathbb{E}X^2$ applied to $\mathbb{E}\left\|\gamma\big(g_j^k-g_j^*-\frac{1}{n}\sum_{i=1}^{n}g_i^k+g^*\big)\right\|^2$, since $\mathbb{E}(g_j^k-g_j^*)=\frac{1}{n}\sum_{i=1}^{n}g_i^k-g^*$.

We further manipulate the second term at the right-hand side of (33). Observe that $\gamma g_j^{k+1}=u_j^k-\mathrm{prox}_{f_j}^{\gamma}(u_j^k)$ by (9) and $\gamma g_j^*=u_j^*-\mathrm{prox}_{f_j}^{\gamma}(u_j^*)$ by (27). Then we have

$$\begin{aligned}&\mathbb{E}\Big\|\big(2\mathrm{prox}_{f_j}^{\gamma}(u_j^k)-u_j^k\big)-\big(2\mathrm{prox}_{f_j}^{\gamma}(u_j^*)-u_j^*\big)\Big\|^2\\&=\mathbb{E}\left\|u_j^k-2\gamma g_j^{k+1}-u_j^*+2\gamma g_j^*\right\|^2\\&=\mathbb{E}\left\|u_j^k-u_j^*\right\|^2+4\mathbb{E}\left\|\gamma(g_j^{k+1}-g_j^*)\right\|^2-4\mathbb{E}\big\langle u_j^k-u_j^*,\gamma(g_j^{k+1}-g_j^*)\big\rangle.\end{aligned} \tag{35}$$

Substituting (34) and (35) into (33) and combining with (31), we obtain the upper bound given by (30). □

Theorem 1 (Non-strongly convex case) *Assume that each loss function f_i is convex and L-smooth, while the regularization function h is convex. Then for Prox2-SAGA with step size $\gamma \leq 1/L$, at any time $k > 0$ it holds*

$$\mathbb{E}\|\bar{g}_j^k - g_j^*\|^2 \leq \frac{1}{k}\Big(\sum_{i=1}^n \|g_i^0 - g_i^*\|^2 + \|\frac{1}{\gamma}(y^0 - y^*)\|^2\Big),$$

where $\bar{g}_j^k = \frac{1}{k}\sum_{t=1}^k g_j^t$. Here the expectation is taken over all choices of index j up to time k.

Proof We further manipulate the upper bound of $\mathbb{E}[T^{k+1}]$ given by (30). Recalling the definitions of $u_j^k = z_j^k + x^k - y^k$ in (23) and $u_j^* = z_j^* + x^* - y^*$ in (26) as well as the definitions of z_j^k in (8) and z_j^* in (26), we bound $\mathbb{E}\|u_j^k - u_j^*\|^2$ as

$$\begin{aligned}
\mathbb{E}\|u_j^k - u_j^*\|^2 &= \mathbb{E}\Big\|2x^k - y^k - (2x^* - y^*) + \gamma\big(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k - g_j^* + g^*\big)\Big\|^2 \\
&= \|2x^k - y^k - (2x^* - y^*)\|^2 + \mathbb{E}\Big\|\gamma\big(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k - g_j^* + g^*\big)\Big\|^2 \\
&\leq \|y^k - y^*\|^2 + \mathbb{E}\Big\|\gamma\big(g_j^k - \frac{1}{n}\sum_{i=1}^n g_i^k - g_j^* + g^*\big)\Big\|^2 \\
&\leq \|y^k - y^*\|^2 + \mathbb{E}\|\gamma\big(g_j^k - g_j^*\big)\|^2. \qquad (36)
\end{aligned}$$

The first inequality is due to the nonexpansiveness of $2\mathrm{prox}_h^\gamma(y) - y$ as stated in Corollary 1, since $x^k = \mathrm{prox}_h^\gamma(y^k)$ by (11) and $x^* = \mathrm{prox}_h^\gamma(y^*)$ by (27). The second inequality comes from the variance formula $\mathbb{E}(X - \mathbb{E}X)^2 \leq \mathbb{E}X^2$ applied to $\mathbb{E}\|\gamma\big(g_j^k - g_j^* - \frac{1}{n}\sum_{i=1}^n g_i^k + g^*\big)\|^2$, since $\mathbb{E}(g_j^k - g_j^*) = \frac{1}{n}\sum_{i=1}^n g_i^k - g^*$.

According to the definitions of g_j^{k+1} by (9) and g_j^* by (27), g_j^{k+1} is the gradient mapping at u_j^k, while g_j^* is the gradient mapping at u_j^*, we further apply Lemma 2 to bound $-\gamma\langle u_j^k - u_j^*, g_j^{k+1} - g_j^*\rangle$ to deduce

$$-\mathbb{E}\langle u_j^k - u_j^*, \gamma(g_j^{k+1} - g_j^*)\rangle \leq -\big(1 + \frac{1}{L\gamma}\big)\mathbb{E}\|\gamma(g_j^{k+1} - g_j^*)\|^2. \qquad (37)$$

Plugging (36) and (37) into (30) and reorganizing terms, we obtain

$$\begin{aligned}
\mathbb{E}[T^{k+1}] \leq T^k &+ \Big(1 - \frac{c}{n}\Big)\frac{1}{n}\sum_{i=1}^n \|\gamma(g_i^k - g_i^*)\|^2 \\
&+ \Big(\frac{c}{n} - \frac{2}{L\gamma} + 1\Big)\mathbb{E}\|\gamma(g_j^{k+1} - g_j^*)\|^2 - \mathbb{E}\|\gamma(g_j^{k+1} - g_j^*)\|^2.
\end{aligned}$$

In particular, we set $c = n$ and $\gamma \leq 1/L$ to ensure that $1 - \frac{c}{n}$ and $\frac{c}{n} - \frac{2}{L\gamma} + 1$ are both non-positive, such that

$$\mathbb{E}[T^{k+1}] \leq T^k - \mathbb{E}\|\gamma(g_j^{k+1} - g_j^*)\|^2. \qquad (38)$$

Taking expectation on both sides of (38) and summing up over times from 0 to k, we have

$$\sum_{t=1}^{k}\mathbb{E}\left\|\gamma(g_j^t-g_j^*)\right\|^2\leq T^0-\mathbb{E}[T^k].$$

Using Jensen's inequality $\sum_{t=1}^{k}\mathbb{E}\left\|(g_j^t-g_j^*)\right\|^2\geq k\mathbb{E}\left\|\bar{g}_j^k-g_j^*\right\|^2$ where $\bar{g}_j^k=\frac{1}{k}\sum_{t=1}^{k}g_j^t$, and throwing away the non-positive term $-\mathbb{E}[T^k]$, we further have

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

Substituting $c=n$ into T^0 completes the proof. □

Theorem 2 (Strongly convex case) *Assume that each loss functions f_i is μ-strongly convex convex and L-smooth, while the regularization function h is convex. Then for Prox2-SAGA with stepsize $\gamma=\min\left\{\frac{1}{\mu n},\frac{\sqrt{9L^2+3\mu L}-3L}{2\mu L}\right\}$, for any time $k>0$ it holds*

$$\mathbb{E}\left\|x^k-x^*\right\|^2\leq\left(1-\frac{\mu\gamma}{2\mu\gamma+2}\right)^k\cdot\frac{\mu\gamma-2}{2-n\mu\gamma}\left\{\sum_{i=1}^{n}\left\|\gamma(g_i^0-g_i^*)\right\|^2+\|y^0-y^*\|^2\right\}.\tag{39}$$

Here the expectation is taken over all choices of index j up to k.

Proof We elaborate on the upper bound of $\mathbb{E}[T^{k+1}]$ given by (30) in a different way than that in the proof of Theorem 1. Since f_j is μ-strongly convex as well as g_j^{k+1} and g_j^* is the gradient mapping of f_j at u_j^k and u_j^*, respectively, from (18) in Lemma 1, it holds that

$$-\frac{1}{2}\langle u_j^k-u_j^*,\gamma(g_j^{k+1}-g_j^*)\rangle\leq-\frac{\mu\gamma}{2(1+\mu\gamma)}\left\|u_j^k-u_j^*\right\|^2.\tag{40}$$

Plugging (40) and (37) into (30) in Lemma 3 and recalling the upper bound for $\mathbb{E}\|u_j^k-u_j^*\|^2$ given by (36) in the proof of Theorem 1, we obtain

$$\begin{aligned}\mathbb{E}[T^{k+1}]\leq&\left(1-\frac{\mu\gamma}{2\mu\gamma+2}\right)T^k+\frac{1}{2}\left(\frac{\mu\gamma}{\mu\gamma+1}c-\frac{2c}{n}+\frac{\mu\gamma+2}{\mu\gamma+1}\right)\frac{1}{n}\sum_{i=1}^{n}\left\|\gamma(g_i^k-g_i^*)\right\|^2\\&+\frac{1}{2}\left(1-\frac{3}{L\gamma}+\frac{2c}{n}\right)\mathbb{E}\left\|\gamma(g_j^{k+1}-g_j^*)\right\|^2.\end{aligned}\tag{41}$$

We choose proper values for c and γ to ensure that the coefficients of the last two terms at the right-hand side of (41) are non-positive. Here we take

$$c=\frac{\mu\gamma+2}{2/n-\mu\gamma},\quad\gamma=\min\left\{\frac{1}{\mu n},\frac{\sqrt{9L^2+3\mu L}-3L}{2\mu L}\right\}.\tag{42}$$

Dropping these two non-positive terms and then taking expectation for (41) with respect to all the previous steps give

$$\mathbb{E}[T^{k+1}]\leq\left(1-\frac{\mu\gamma}{2\mu\gamma+2}\right)\mathbb{E}[T^k].$$

Further chaining over k yields

$$\mathbb{E}[T^k] \leq \left(1 - \frac{\mu\gamma}{2\mu\gamma + 2}\right)^k \cdot T^0.$$

Due to the firm nonexpansiveness of $x^k = \text{prox}_h^\gamma(y^k)$, we have

$$\mathbb{E}\|x^k - x^*\|^2 \leq \mathbb{E}\|y^k - y^*\|^2 \leq \mathbb{E}[T^k] \leq \left(1 - \frac{\mu\gamma}{2\mu\gamma + 2}\right)^k \cdot T^0.$$

Substituting $c = \frac{\mu\gamma+2}{2/n-\mu\gamma}$ into T^0 completes the proof. □

Remark 1 Under the step size rule $\gamma = \min\left\{\frac{1}{\mu n}, \frac{\sqrt{9L^2+3\mu L}-3L}{2\mu L}\right\}$, to achieve an ϵ-accurate solution x^k such that $\mathbb{E}\left\|x^k - x^*\right\|^2 \leq \epsilon$, the number of required steps is $\mathcal{O}(n + L/\mu)\log(1/\epsilon)$, which is consistent with existing VR stochastic algorithms. Nevertheless, when f_i is ill-conditioned, namely, $L/\mu \gg n$, we can use a different step size rule

$$\gamma = \min\left\{\frac{1}{\mu n}, \frac{6L + \sqrt{36L^2 - 6(n-2)\mu L}}{2(n-2)\mu L}\right\},$$

under which the number of required steps to achieve an ϵ-accurate solution is $\mathcal{O}(n + \sqrt{nL/\mu})\log(1/\epsilon)$. This accelerated rate is consistent with the fastest accelerated methods such as Acc-SDCA and Katyusha.

7 Experiments

In this section, we conduct numerical experiments to validate the effectiveness and the theoretical properties of the proposed Prox2-SAGA algorithm. In the experiments, we focus on sparse SVMs:

$$\min_x F(x) = \frac{1}{n}\sum_{i=1}^n \max\left\{0, 1 - b_i a_i^T x\right\} + \lambda_1 \|x\|_1 + \frac{\lambda_2}{2}\|x\|^2 \tag{43}$$

and $\ell_1\ell_2$-Logistic Regression (LR):

$$\min_x F(x) = \frac{1}{n}\sum_{i=1}^n \log\left(1 + \exp\left(-b_i a_i^T x\right)\right) + \lambda_1 \|x\|_1 + \frac{\lambda_2}{2}\|x\|^2, \tag{44}$$

where $a_i \in \mathbb{R}^d$, $b_i \in \{-1, +1\}$ and $\lambda_1, \lambda_2 \geq 0$. The first problem involves the non-smooth hinge loss, from which we verify the effectiveness of Prox2-SAGA to handle non-smooth loss functions. The $\ell_1\ell_2$-logistic regression contains smooth logistic functions, from which we verify the acceleration effect of Prox2-SAGA and the performance of Prox2-SAGA for non-strongly convex problems.

We employ datasets from LIBSVM (Chang et al. 2011) which are summarized in Table 1. By referring to the previous works, we set the values of λ_1 and λ_2. Some values of λ_1 and λ_2 are also listed in Table 1. Prox-SAGA (Defazio et al. 2014), Prox-SDCA (Shalev-Shwartz and Zhang 2014), Prox-SGD (Duchi and Singer 2009; Langford et al. 2009) and Acc-SDCA (Shalev-Shwartz and Zhang 2014) are included in the experiments for comparison. In the k-th iteration, the step size of Prox-SGD is set as $\gamma^k = \gamma^0/(1 + \gamma^0\eta k)$ with $\gamma^0, \eta > 0$,

Table 1 Summary of the datasets and models used in the experiments

Dataset	n	d	Model	λ_1	λ_2
svmguide3	1243	21	SVM	10^{-3}	10^{-3}
rcv1	20242	47236	SVM	10^{-5}	10^{-5}
covtype	581012	54	SVM	10^{-5}	10^{-5}
ijcnn1	49990	22	SVM	10^{-4}	10^{-5}
mushrooms	8124	112	LR	10^{-4}	
w7a	24692	300	LR	5×10^{-5}	

and we take the fixed step size for other algorithms. We tune the step size and the other parameters for different algorithms so that they can achieve the best performance. To make a fair comparison, the initial value of x is set to zero in all algorithms. Denote the number of samples as n, we measure the *objective gap* at x as $f(x) - f(x^*) + g(x) - g(x^*)$ and the *epoch* as the evaluation of n component gradients to evaluate the performance of algorithms.

7.1 Sparse SVMs

We first compare the performance of the proposed Prox2-SAGA with Prox-SGD, Prox-SAGA and Prox-SDCA for solving (43). For the non-smooth hinge loss $f_i(x) = \max\{0, 1 - b_i a_i x\}$, we take its subgradient $g_i = -\mathbb{1}\{b_i a_i^T x \leq 1\} b_i a_i$, and the proximal operator has a closed-form expression:

$$\text{prox}_{f_i}^{\gamma}(x) = x - \gamma b_i u a_i,$$

where

$$u = \begin{cases} -1, & \text{if } s \geq 1 \\ 0, & \text{if } s \leq 0 \\ -s, & \text{otherwise} \end{cases}, \qquad s = \frac{1 - b_i \cdot a_i^T x}{y \|a_i\|^2}.$$

Note that only Prox-SDCA and Prox-SGD can be theoretically guaranteed to converge to the minimizer of (1).

Experiments are conducted on four datasets, and the results are shown in Fig. 1. It can be seen that Prox2-SAGA works well with non-smooth loss functions. In contrast, the performance of the Prox-SGD algorithm is poor on all the datasets. Meanwhile, although Prox-SAGA may perform well in the beginning, it is possible to get stuck in the later iterations, which is particularly evident on the rcv1 dataset.

7.2 $\ell_1\ell_2$-logistic regression

In the investigation here, we compare the performance of Prox2-SAGA with Prox-SGD, Prox-SAGA, Prox-SDCA and Acc-SDCA for solving (44). Prox2-SAGA and Acc-SDCA are the accelerated methods for Prox-SAGA and Prox-SDCA, respectively. For the log loss $f_i(x) = \log(1 + \exp(b_i a_i^T x))$, the proximal operator can be computer efficiently by several Newton iterations. That is to say, we start from an initial point $c^0 \in \mathbb{R}$, and do the following iterations until convergence

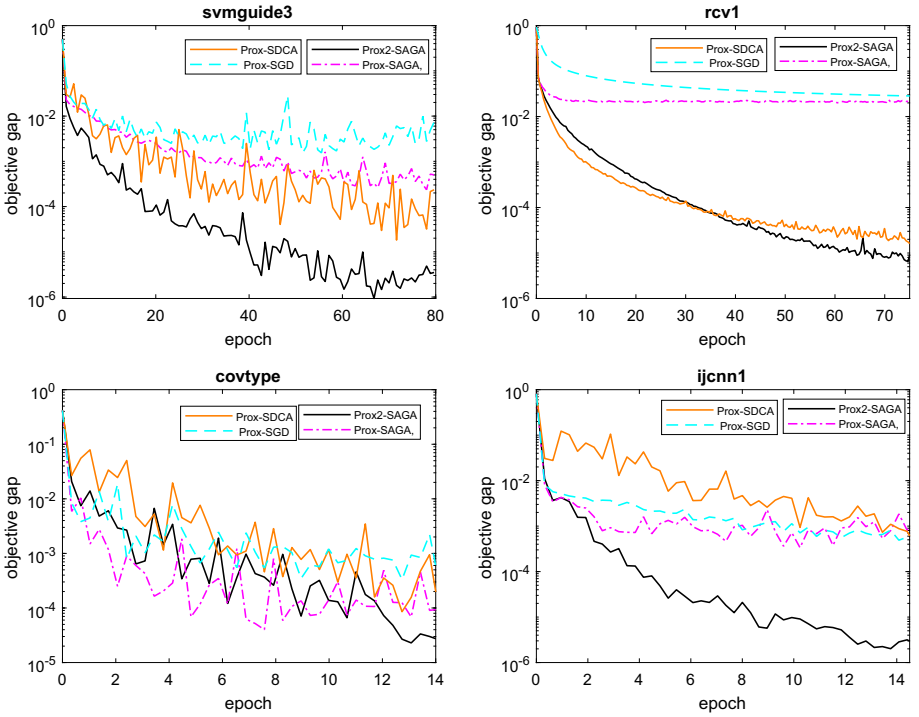

Fig. 1 Comparison of several algorithms with sparse SVMs

$$s^k = -\frac{b_i}{1+\exp(b_i c^k)},$$
$$c^{k+1} = c^k - \frac{\gamma\|a_i\|^2 s^k + c^k - a_i^T x}{\gamma\|a_i\|^2 \exp(b_i c^k) s_k^2 + 1}.$$

Then the proximal operator is

$$\text{prox}_{f_i}^{\gamma}(x) = x - (a_i^T x - c^k)a_i/\|a_i\|^2.$$

Note that the Prox-SDCA and Acc-SDCA (Shalev-Shwartz and Zhang 2014) algorithms also need to employ such Newton iterations in practice. In order to understand the impacts of condition number on these algorithms, we set three different values of λ_2 for each dataset, which are marked in Fig. 2. Note that $\lambda_2 = 0$ corresponds to the non-strongly convex case, where Prox-SDCA and Acc-SDCA are not suitable. We use Acc-SDCA for comparison rather than other accelerated algorithms, since Acc-SDCA has less parameters to tune and is more practical.

Experiments are conducted on the datasets of mushrooms and w7a, and the results are shown in Fig. 2. One can observe that for relatively large λ_2, most VR stochastic methods perform similarly. On the other hand, when λ_2 gets smaller, the accelerated methods are significantly faster than the non-accelerated methods in most cases. This shows that as the accelerated algorithms, Prox2-SAGA and Acc-SDCA can resist the ill conditions well. Furthermore, Prox2-SAGA is more stable and performs better than Acc-SDCA according to Fig. 2.

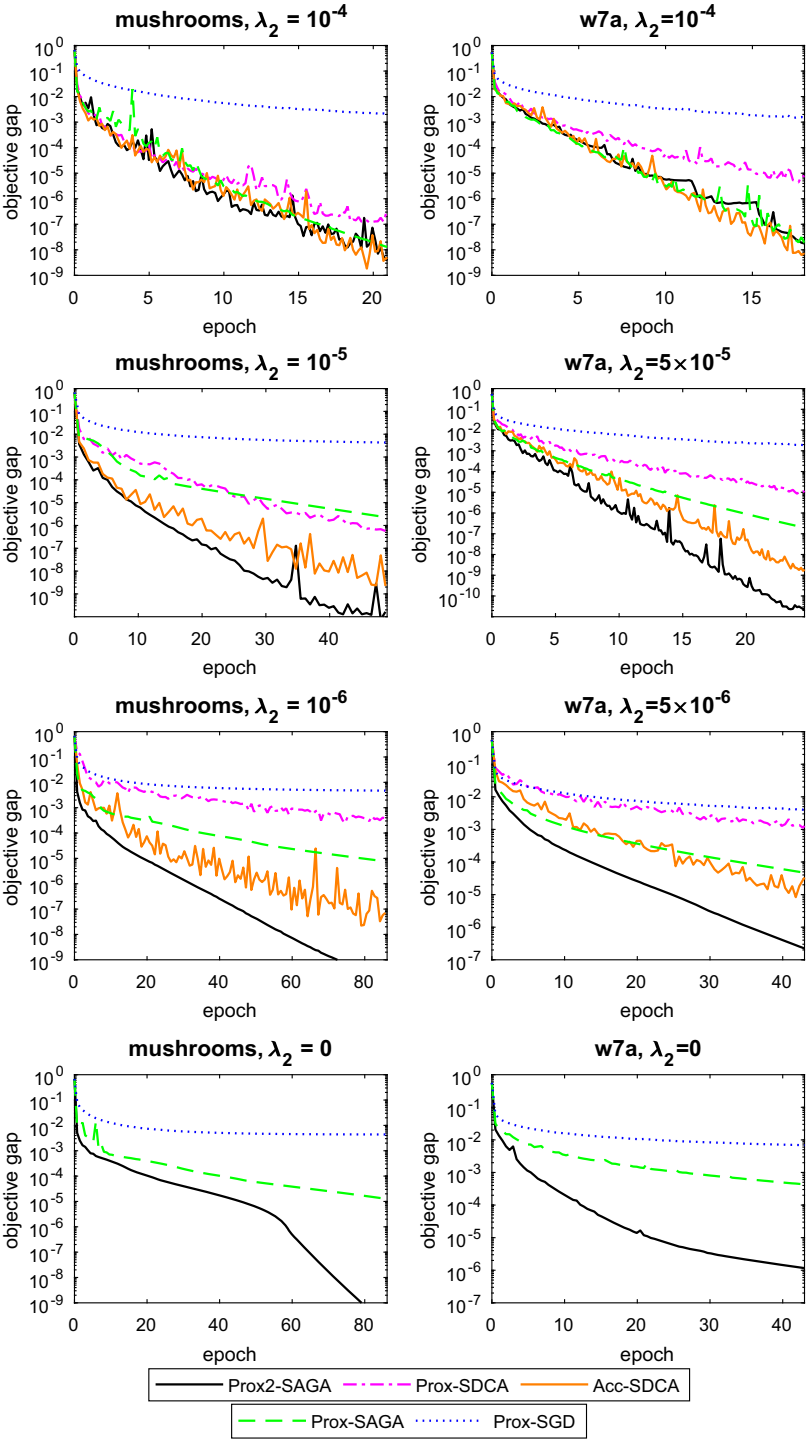

Fig. 2 Comparison of several algorithms with $\ell_1\ell_2$-Logistic Regression

8 Conclusion

In this paper, we propose a novel VR stochastic algorithm, Prox2-SAGA, to solve the regularized empirical risk minimization problem. At every iteration of Prox2-SAGA, we use two proximal operators, one on a randomly chosen loss function and the other on the regularization function. Accelerated convergence rate can be achieved when each loss function is strongly convex and smooth. Experimental results demonstrate its superiority over the other VR stochastic methods.

Acknowledgements Research supported by the National Natural Science Foundation of China (No. 61673364, No. 91746301) and the Fundamental Research Funds for the Central Universities (WK2150110008).

References

Allen-Zhu, Z. (2017). Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th annual ACM SIGACT symposium on theory of computing, ACM* (pp. 1200–1205).

Bauschke, H. H., & Combettes, P. L. (2017). *Convex analysis and monotone operator theory in Hilbert spaces*. Berlin: Springer.

Bottou, L., Curtis, F. E., & Nocedal J. (2016). Optimization methods for large-scale machine learning. arXiv:1606.4838.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27.

Defazio, A. (2016). A simple practical accelerated method for finite sums. In *Advances in neural information processing systems* (pp. 676–684).

Defazio, A., Bach, F., & Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems* (pp. 1646–1654).

Duchi, J., & Singer, Y. (2009). Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, *10*, 2899–2934.

Eckstein, J., & Bertsekas, D. P. (1992). On the douglasrachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, *55*(1–3), 293–318.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (2nd ed.). Berlin: Springer.

Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems* (pp. 315–323).

Langford, J., Li, L., & Zhang, T. (2009). Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, *10*, 777–801.

Lemaréchal, C., & Sagastizábal, C. (1997). Practical aspects of the Moreau-Yosida regularization: Theoretical preliminaries. *SIAM Journal on Optimization*, *7*(2), 367–385.

Lin, H., Mairal, J., Harchaoui, Z. (2015). A universal catalyst for first-order optimization. In *Advances in neural information processing systems* (pp. 3384–3392).

Lin, H., Mairal, J., Harchaoui, Z. (2017). Catalyst acceleration for first-order convex optimization: From theory to practice. arXiv:1712.5654.

Needell, D., Ward, R., & Srebro, N. (2014). Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in neural information processing systems* (pp. 1017–1025).

Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course* (Vol. 87). Berlin: Springer.

Owen, A. B. (2013) Monte Carlo theory, methods and examples.

Parikh, N., & Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, *1*(3), 127–239. https://doi.org/10.1561/2400000003.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, *22*, 400–407.

Ross, S. (2013). Chapter 9 - variance reduction techniques. In S. Ross (Ed.), *Simulation* (5th ed., pp. 153–231). Cambridge: Academic Press.

Schmidt, M., Le Roux, N., & Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, *162*(1–2), 83–112.

Shalev-Shwartz, S., & Zhang, T. (2012). Proximal stochastic dual coordinate ascent. arXiv:1211.2717.

Shalev-Shwartz, S., & Zhang, T. (2013). Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, *14(Feb)*, 567–599.

Shalev-Shwartz, S., & Zhang, T. (2014). Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International conference on machine learning* (pp. 64–72).

Shamir, O., & Zhang, T. (2013). Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning* (pp. 71–79).

Woodworth, B. E., & Srebro, N. (2016). Tight complexity bounds for optimizing composite objectives. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 29, pp. 3639–3647). New York: Curran Associates, Inc.,

Xiao, L., & Zhang, T. (2014). A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, *24*(4), 2057–2075.

Zhao, P., & Zhang, T. (2014) Accelerating minibatch stochastic gradient descent using stratified sampling. arXiv:1405.3080.

Zhao, P., & Zhang, T. (2015). Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32nd international conference on machine learning (ICML-15)* (pp. 1–9).