Control and Optimization in Applied Mathematics (COAM)
Vol. 8, No. 1, Winter-Spring 2023 (19-32), ©2016 Payame Noor University, Iran

DOI. 10.30473/coam.2023.64060.1205 (Cited this article)

Research Article

A Proximal Method of Stochastic Gradient for Convex Optimization

Zeinab Saeidian[1,*], **Maryam Mahmoudoghli**[2]

[1]Department of Mathematics, University of Kashan, Kashan, Iran.
[2] K.N. Toosi University of Technology, Tehran, Iran.

Received: May 23, 2022; **Accepted:** April 08, 2023.

Abstract. The Proximal Stochastic Average Gradient (Prox-SAG+) is a primary method used for solving optimization problems that contain the sum of two convex functions. This kind of problem usually arises in machine learning, which utilizes a large amount of data to create component functions from a dataset. A proximal operation is applied to obtain the optimal value due to its appropriate properties. The Prox-SAG+ algorithm is faster than some other methods and has a simpler algorithm than previous ones. Moreover, using this specific operator can help to reassure that the achieved result is optimal. Additionally, it has been proven that the proposed method has an approximately geometric rate of convergence. Implementing the proposed operator makes the method more practical than other algorithms found in the literature. Numerical analysis also confirms the efficiency of the proposed scheme.

Keywords. Proximal stochastic average gradient, Convergence property, Training examples, Machine learning.

MSC. 25C90; 50M78; 65K05; 90C30.

* Corresponding author
saeidian@kashanu.ac.ir, m.mamoudoghli@gmail.com
https://mathco.journals.pnu.ac.ir

1 Introduction

In this paper, we deal with the optimization problem to compute an approximated minimizer of the function which is the summation of the finite number of component functions. This problem arises in many applications such as machine learning and Data Mining. This minimization problem is as follows

$$\min P(x), \tag{1}$$

in which $P(x) = F(x)+R(x)$. The function $F(x)$ is the average of many smooth component functions such as

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

and the function $R(x)$ can be non-differentiable. A large number of training examples makes the problem more practical. This problem is also known as regularized empirical risk minimization [1]. In such problems, we have training examples $(a_1, b_1), \ldots, (a_n, b_n)$, where each of $a_i \in \mathbb{R}^d$ is a feature vector and $b_i \in \mathbb{R}$ is the desired response.

Now, let us mention some methods which have been proposed by some researchers. One of the methods used for solving (1) is the Proximal Full Gradient (Prox-FG) [7]. Now, let us mention some methods which have been proposed by some researchers. One of the methods used for solving (1) is the Proximal Full Gradient (Prox-FG) (see Equation (6), [15]). In this method, in each iteration $k = 1, 2, \ldots,$ an i_k is chosen randomly from $\{1, \ldots, n\}$. Shwartz and Zhang [11, 12] proposed an effective function $f_i(x) = \phi_i(a_i^T x)$, for solving the problem (1) which is choosing Fenchel conjugate functions of ϕ_i and R . The Fenchel conjugate function is

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

where f^* is a closed and convex function and

$$\mathrm{dom}(f) := \{x \in \mathbb{R}^d | f(x) < +\infty\}.$$

The inner product which is used in the previous equality is a vector space V over the field F, which is a map

$$\langle \cdot, \cdot \rangle : V \times V \to F.$$

Assuming $R(x)$ is μ-strongly convex, they indicated that a proximal stochastic dual coordinate ascent (Prox-SDCA) method has the same complexity as the other methods. Le Roux et al., set $R(x) \equiv 0$ and offered a new Stochastic Average Gradient (SAG) method [4]. Another scheme that was proposed by Johnson and Zhang, is called Stochastic Variance-Reduced Gradient (SVRG) [3]. The SVRG method uses a multi-stage plan to gradually reduce the variance generated through the stochastic gradient. Later, the various reduction in SVRG was extended, so the method was developed to a Proximal SVRG (Prox-SVRG) [15]. Also, in this method, uniform sampling of the component functions was applied. Then, Li and Li proposed another method that was termed Prox-SVRG+ [5]. Although this algorithm is based on variance reduction, it does not have the geometric convergence in expectation.

Recently, a method that is called Prox-GEN [17] has been proposed, in which the regulator can be non-smooth and non-convex. It uses a unified framework for stochastic proximal gradient descent and shows that the whole family has the same convergence rate. For more detail, refer to [13, 14, 16].

Let us consider two following assumptions that are necessary to use as the primary rules [15].

Assumption 1. Suppose that $R(x)$ is a lower semi-continuous and also convex function with a closed domain $\mathrm{dom}(R) := \{x \in \mathbb{R}^d | R(x) < +\infty\}$. All $f_i(x)$ for $i = 1,\ldots,n$, are differentiable on a supposed open set with $\mathrm{dom}(R)$, and their gradients are Lipschitz continuous. For Lipschitz continuity, there exists L_i, such that for all $x, y \in \mathrm{dom}(R)$ we get

$$\| \nabla f_i(x) - \nabla f_i(y) \| \leqslant L_i \| x - y \| . \tag{3}$$

Assumption 2. Suppose that $P(x)$, the cost function in (1), is strongly convex, then there exists $\tau > 0$ such that for all $x \in \mathrm{dom}(R)$, $y \in \mathbb{R}^d$ and ∂P as a partial derivative of P satisfy. We obtain

$$P(y) \geqslant P(x) + \zeta^T (y - x) + \frac{\tau}{2} \| y - x\|^2, \quad \forall \zeta \in \partial P(x). \tag{4}$$

In this paper, we propose a proximal method of the SAG approach which makes it more practical. Applying a proximal operator in this method implies that our method executes easier than the original SAG method in the case of $R(x) \neq 0$. The mentioned operator can help us to achieve the optimal value readily.

The rest of this paper is as follows. In Section 2, we describe some essential definitions. Then, in Section 3, we explain the proximal method. Section 4 is devoted to explaining the new algorithm. In Section 5, the convergence properties are analyzed. Finally, in Section 6, we illustrate the numerical experiments.

2 Some Basic Definition

We need to describe a special operator. Suppose that $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed convex function where

$$\mathrm{epi}\ h = \{(x, u) \in \mathbb{R}^n \times \mathbb{R} | h(x) \leqslant u\},$$

is a nonempty closed convex set. Also, the proximal operator [8] $\mathrm{prox}_h : \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$\mathrm{prox}_h(v) = \arg\min_x (h(x) + (1/2) \| x - v \|_2^2), \tag{5}$$

where $\| \cdot \|_2$ shows the Euclidean norm. The function is strongly convex and it is not everywhere infinite, then it has a unique optimal (minimizer) for every $v \in \mathbb{R}^n$.

We have the scaled function λh, where

$$\text{prox}_{\lambda h}(v) = \arg\min_x (h(x) + (1/2\lambda) \| x - v \|_2^2), \qquad \lambda > 0.$$

Figure 1 demonstrates that applying this operator causes the red points to converge to the minimum of the function, even when some points, such as the blue points, are either within or outside the function's domain.

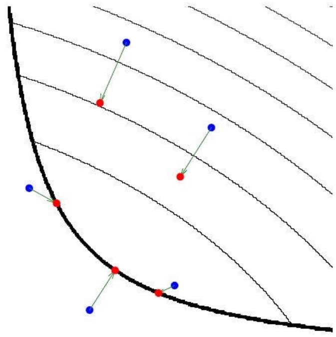

Figure 1: Evaluating a proximal operator at different points [8].

The definition of the mentioned operator indicates that $\text{prox}_h(v)$ is a point that compromises between being close to v and minimizing h. The proximal operator can be interpreted as a gradient step for function h. Hence, we obtain

$$\text{prox}_{\lambda h}(v) \approx v - \lambda \nabla h(v),$$

once λ is small and h is differentiable. As a result, we can see a connection between proximal operation and gradient methods [8].

3 Proximal Method

In this section, we will describe one of the popular methods used to solve problem (1). Additionally, we will provide some details regarding this method.

The proposed method used to solve the problem (1) is the proximal gradient method. An initial point is given as the input of the algorithm. The update rule for the proximal method is

$$x_k = \arg\min_{x \in \mathbb{R}^d} \{\nabla F(x_{k-1})^T x + \frac{1}{2\gamma_k} \| x - x_{k-1} \|^2 + R(x)\}, \quad k = 1, 2, \ldots, \tag{6}$$

where γ_k is the step size at the k-th iteration and $R(x) = \lambda_1 \| x \|_1$, $R(x) = \lambda_2/2 \| x \|_2^2$, or the sum of these two forms $R(x) = \lambda_1 \| x \|_1 + \lambda_2/2 \| x \|_2^2$, in which λ_1 and λ_2 are nonnegative regularization parameters. The loss function is logistic loss as $f_i(x) = \log(1 + \exp(-b_i a_i^T x))$ and it can be added to any of the regularization terms. Throughout this paper, to simplify, we use $\| \cdot \|$ instead of $\| \cdot \|_2$, where it shows the Euclidean norm.

As in the Proximal SVRG in [15], the operator is used as a gradient step, so we can apply it to the SAG method.

So, in proximal gradient, we have

$$x_k = \operatorname{prox}_{\gamma_k R}(x_{k-1} - \gamma_k \nabla F(x_{k-1})), \tag{7}$$

where n, the number of component functions, can be very large. Hence, the Prox-FG method would be too expensive. The alternative method which has a lower cost is Prox-SG which uses some training examples. Choosing these limited number of training examples that leads to a limited number of component functions is through a random process. So, (7) is written as

$$x_k = \operatorname{prox}_{\gamma_k R}(x_{k-1} - \gamma_k \nabla f_{i_k}(x_{k-1})), \tag{8}$$

where i_k is chosen randomly among the set of $\{1, \ldots, n\}$.

Also, we have

$$E\nabla f_{i_k}(x_{k-1}) = \nabla F(x_{k-1}). \tag{9}$$

4 Prox-SAG+ Method

This section is allocated to explain how we get the idea of the new method. Also, the proposed algorithm is described in detail.

In the SAG method [10] such as the previous methods for solving the problem (1), the initial point x_0 is given, and the update rule is defined as follow.

$$x_{k+1} = x_k - \frac{\gamma_k}{n} \sum_{i=1}^{n} y_i^k, \tag{10}$$

where i_k is drawn randomly and y_i^k computed by

$$y_i^k = \begin{cases} \nabla f_i(x^{k-1}), & \text{if } \ i = i_k, \\ y_i^{k-1}, & \text{otherwise.} \end{cases} \tag{11}$$

Our method (Prox-SAG+) has been done even in the case where $R(x) \neq 0$. In the case of $R(x) \neq 0$, algorithms are complicated, especially for implementing test problems. We propose a proximal method that is equipped with the proximal operator

$$x_k = \operatorname{prox}_{\gamma_k R}(x_{k-1} - \gamma_k v_k), \tag{12}$$

$$v_k = v_{k-1} - y_i^k + \nabla f_{i_k}(x), \tag{13}$$

where is obtained from [11].

Now, let us introduce the Prox-SAG+ algorithm.

Algorithm 2 Prox-SAG+ algorithm

$v = 0, y_i = 0$ **for** $i = 1, 2, \dots, n$
for $k = 1, 2, \dots$ do
  Sample i from $\{1, 2, \dots, n\}$
  $v_k = v_{k-1} - y_i + \nabla f_i(x)$
  $y_i = \nabla f_i(x)$
  $x_k = \mathrm{prox}_{\gamma_k R}(x_{k-1} - \gamma_k v_k)$
end for

5 Convergence Analysis

To analyze the convergence of the new method, we express some lemmas. These lemmas are used to prove the principle theorem.

The following lemmas are similar to the ones in [15].

Lemma 1. $P(x)$ is considered as satisfied in (1) and (2). Let Assumption 1 holds, $x^* = \arg\min_x P(x)$ and $L_s = max_i L_i/n$. So

$$\frac{1}{n}\sum_{i=1}^{n} \| \nabla f_i(x) - \nabla f_i(x^*)\|^2 \leqslant 2L_S[P(x) - P(x^*)].$$

Proof. Consider the following function

$$\psi_i(x) = f_i(x) - f_i(x^*) - \nabla f_i(x^*)^T(x - x^*).$$

It is easy to check $\nabla\psi_i(x^*) = 0$, so $\min_x \psi_i(x) = \psi_i(x^*) = 0$. As $\nabla\psi_i(x)$ is Lipschitz continuous with constant L_i, and from (Theorem 2.1.5, [6]), we have

$$\frac{1}{2L_i} \| \nabla\psi_i(x)\|^2 \leqslant \psi_i(x) - \min_y \psi_i(y) = \psi_i(x) - \psi_i(x^*) = \psi_i(x).$$

This implies that

$$\| \nabla f_i(x) - \nabla f_i(x^*)\|^2 \leqslant 2L_i[f_i(x) - f_i(x^*) - \nabla f_i(x^*)^T(x - x^*)].$$

Now by multiplying the last inequality by $1/n$, in addition to summing over $i = 1, \dots, n$, it is obtained that

$$\frac{1}{n}\sum_{i=1}^{n} \| \nabla f_i(x) - \nabla f_i(x^*)\|^2 \leqslant 2L_S[F(x) - F(x^*) - \nabla F(x^*)(x - x^*)].$$

As x^* is the optimal point,

$$x^* = \arg\min_x P(x) = \arg\min_x \{F(x) + R(x)\},$$

there exists $\zeta^* \in \partial R(x^*)$, which ∂R is a partial derivative, that $\nabla F(x^*) + \zeta^* = 0$. Then

$$F(x) - F(x^*) - \nabla F(x^*)(x - x^*) = F(x) - F(x^*) + \zeta^*(x - x^*)$$

$$\begin{aligned}&\leqslant F(x)-F(x^*)+R(x)-R(x^*)\\&=P(x)-P(x^*).\end{aligned}$$

In the previous inequality, the convexity of $R(x)$ is supposed (from Assumption 1). So we have

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x)-\nabla f_i(x^*)\|^2\leqslant 2L_S[P(x)-P(x^*)].$$

□

Corollary 1. Let v_k be as defined in (13) (Corollary 3, [15]), suppose that at the point of x_{k-1} we have $Ev_k\leqslant\nabla F(x_{k-1})$ then

$$E\|v_k-\nabla F(x_{k-1})\|^2\leqslant 4L_S[P(x_{k-1})-P(x^*)]+M.$$

Proof. Conditioned on x_{k-1}, it is taken expectation concerning i_k to gain

$$E\left[\nabla f_{i_k}(x_{k-1})\right]=\sum_{i=1}^{n}\frac{1}{n}\nabla f_{i_k}(x_{k-1})=\nabla F(x_{k-1}).$$

Now, the following inequality can be achieved

$$\begin{aligned}E\|v_k-\nabla F(x_{k-1})\|^2&=E\|v_{k-1}-y_i^k+\nabla f_{i_k}(x_{k-1})-\nabla F(x_{k-1})\|^2\\&\leqslant 2E\|v_{k-1}-y_i^k\|^2+2\mathrm{E}\|\nabla f_{i_k}(x_{k-1})-\nabla F(x_{k-1})\|^2\\&\leqslant 2E\|v_{k-1}-y_i^k\|^2+2E\|\nabla f_{i_k}(x_{k-1})\|^2-2\|\nabla F(x_{k-1})\|^2\\&\leqslant 2E\|v_{k-1}-y_i^k\|^2+2\mathrm{E}\|\nabla f_{i_k}(x_{k-1})\|^2\\&=2\|v_{k-1}-y_i^k\|^2+2\mathrm{E}\|\nabla f_{i_k}(x_{k-1})+\nabla f_{i_k}(x^*)-\nabla f_{i_k}(x^*)\|^2\\&\leqslant\ 2\|v_{k-1}-y_i^k\|^2+4\mathrm{E}\|\nabla f_{i_k}(x_{k-1})-\nabla f_{i_k}(x^*)\|^2+4\|\nabla f_{i_k}(x^*)\|^2\\&\leqslant \mathrm{M}\ +\ 4\mathrm{E}\|\nabla f_{i_k}(x_{k-1})-\nabla f_{i_k}(x^*)\|^2\\&\leqslant 4L_S[P(x_{k-1})-P(x^*)]+M.\end{aligned}$$

In the first and fourth inequality, we used $\|\alpha+\beta\|^2\leqslant 2\|\alpha\|^2+2\|\beta\|^2$ and finally, the second equality is achieved from Lemma 1 and, the fact that for any random vector $x_i\in\mathbb{R}^d$, we have $E\|\xi-E\xi\|^2=E\|\xi\|^2-\|E\xi\|^2$. □

We need two more lemmas to use and complete proving the convergence theorem (Section 31, [9]).

Lemma 2. Consider R being a closed convex function on $\mathbb{R}^d$ and also $x,y\in\mathrm{dom}(R)$. So

$$\|\mathrm{prox}_R(x)-\mathrm{prox}_R(y)\|\leqslant\|x-y\|.$$

To obtain a lower bound we use the next lemma (Lemma 3, [2]).

Lemma 3. Consider $P(x) = F(x) + R(x)$, in which $\nabla F(x)$ is Lipschitz continuous with its parameter L. Also, $F(x)$ and $R(x)$ has strong convexity parameters τ_F and τ_R. For any $x \in \mathsf{dom}(R)$ and arbitrary $v \in \mathbb{R}^d$, we define

$$x_+ = \mathsf{pro}\,x_{\gamma R}(x - \gamma v),$$
$$d = \frac{1}{\gamma}(x - x_+),$$
$$\Delta = v - \nabla F(x),$$

where $0 < \gamma \leqslant 1/L$ is a step size. Now for any $y \in \mathbb{R}^d$, we have

$$P(y) \geqslant P(x_+) + d^T(y - x) + \frac{\gamma}{2}\parallel d\parallel^2 + \frac{\tau_F}{2}\parallel y - x\parallel^2 + \frac{\tau_R}{2}\parallel y - x_+\parallel^2 + \Delta^T(x_+ - y).$$

Now, consider the following convergence theorem.

Theorem 1. Let Assumptions 1 and 2 are satisfied, and $x^* = \arg\min_x P(x)$ and $L_S = \max_i L_i/n$. Furthermore, suppose that $0 < \gamma < 1/(4L_S)$ and

$$\rho = 1 - \frac{2\gamma - \frac{2}{\tau}}{8\gamma^2 L_S} < 1. \tag{14}$$

Then, the Prox-SAG+ method has the geometric convergence in expectation

$$EP(x_k) - P(x^*) \leqslant \rho^k[P(x_0) - P(x^*)]. \tag{15}$$

Proof. The stochastic gradient mapping is defined for convergence

$$d_k = \frac{1}{\gamma}(x_{k-1} - x_k) = \frac{1}{\gamma}(x_{k-1} - \mathsf{pro}\,x_{\gamma R}(x_{k-1} - \gamma v_k)),$$

then the proximal gradient step (12) can be rewritten as

$$x_k = x_{k-1} - \gamma d_k.$$

To complete the proof of Theorem 1, we need to know the distance between x_k and x^*.

$$\begin{aligned}\parallel x_k - x^*\parallel^2 &= \parallel x_{k-1} - \gamma d_k - x^*\parallel^2 \\ &= \parallel x_{k-1} - x^*\parallel^2 - 2\gamma d_k^T(x_{k-1} - x^*) + \gamma^2 \parallel d_k\parallel^2.\end{aligned}$$

Using Lemma 3 with $x = x_{k-1}$, $v = v_k$, $x_+ = x_k$, $d = d_k$ and $y = x^*$, we have

$$-d_k^T(x_{k-1} - x^*) + \frac{\gamma}{2}\parallel d_k\parallel^2 \leqslant P(x^*) - P(x_k) - \frac{\tau_F}{2}\parallel x_{k-1} - x^*\parallel^2 - \frac{\tau_R}{2}\parallel x_k - x^*\parallel^2 + \Delta_k^T(x_k - x^*),$$

in which $\Delta_k = v_k - \nabla F(x_{k-1})$. By using the assumption in theorem 1 we get

$$\eta < 1/(4L_S) < 1/L$$

since $L_S \geqslant (1/n)\sum_{i=1}^n L_i \geqslant L$. As a result,

$$
\begin{aligned}
\| x_k - x^*\|^2 &\leqslant \| x_{k-1} - x^*\|^2 - \gamma\tau_F \| x_{k-1} - x^*\|^2 - \gamma\tau_R \| x_k - x^*\|^2 \\
&\quad - 2\gamma[P(x_k) - P(x^*)] - 2\gamma\Delta_k^T(x_k - x^*) \\
&\leqslant \| x_{k-1} - x^*\|^2 - 2\gamma[P(x_k) - P(x^*)] - 2\gamma\Delta_k^T(x_k - x^*). \qquad (16)
\end{aligned}
$$

Then, we find an upper bound $-2\gamma\Delta_k^T(x_k - x^*)$. In addition, we mention the proximal full gradient updates as

$$\tilde{x}_k = \mathrm{prox}_{\gamma R}(x_{k-1} - \gamma\nabla F(x_{k-1})),$$

which is independent of the random variable i_k. So,

$$
\begin{aligned}
-2\gamma\Delta_k^T(x_k - x^*) &= -2\gamma\Delta_k^T(x_k - \tilde{x}_k) - 2\gamma\Delta_k^T(\tilde{x}_k - x^*) \\
&\leqslant 2\gamma \| \Delta_k \| \| x_k - \tilde{x}_k \| - 2\gamma\Delta_k^T(\tilde{x}_k - x^*) \\
&\leqslant 2\gamma \| \Delta_k \| \| x_{k-1} - \gamma v_k) - (x_{k-1} - \gamma\nabla F(x_{k-1})) \| - 2\gamma\Delta_k^T(\tilde{x}_k - x^*) \\
&= 2\gamma^2 \| \Delta_k\|^2 - 2\gamma\Delta_k^T(\tilde{x}_k - x^*),
\end{aligned}
$$

where the Cauchy-Schwarz inequality was used in the first inequality, and in the second inequality, Lemma 2 was used. Combining with (16), the following result was obtained

$$
\begin{aligned}
\| x_k - x^*\|^2 &\leqslant \| x_{k-1} - x^*\|^2 - 2\gamma[P(x_k) - P(x^*)] + 2\gamma^2 \| \Delta_k\|^2 - 2\gamma\Delta_k^T(\tilde{x}_k - x^*). \\
E \| x_k - x^*\|^2 &\leqslant \| x_{k-1} - x^*\|^2 - 2\gamma[EP(x_k) - P(x^*)] + 2\gamma^2 E \| \Delta_k\|^2 - 2\gamma E[\Delta_k^T(\tilde{x}_k - x_*)] \\
&\leqslant \| x_{k-1} - x^*\|^2 - 2\gamma[EP(x_k) - P(x^*)] + 2\gamma^2(4L_S[P(x_{k-1}) - P(x^*) + M].
\end{aligned}
$$

Now, by taking expectations again over the last inequality we obtain the desired result

$$
\begin{aligned}
&E \| x_{k-1} - x^*\|^2 - 2\gamma[EP(x_k) - P(x^*)] + 2\gamma^2 \times 4L_S[EP(x_{k-1}) - P(x^*)] + 2\gamma^2 M \\
&\leqslant \| x_0 - x^*\|^2 - 2\eta[P(x_0) - P(x^*)] + 8\gamma^2 L_S[P(x_0) - P(x^*)] + 2\gamma^2 M,
\end{aligned}
$$

and then by using the last inequality, we have

$$8\gamma^2 L_S[EP(x_{k-1}) - P(x^*)] \leqslant \| x_0 - x^*\|^2 - 2\gamma[P(x_0) - P(x^*)] + 8\gamma^2 L_S[P(x_0) - P(x^*)].$$

In addition, we have $\| x_0 - x^*\|^2 \leqslant \frac{2}{\tau}[P(x_0) - P(x^*)]$. Therefore,

$$
\begin{aligned}
8\gamma^2 L_S[EP(x_{k-1}) - P(x^*)] &\leqslant (\frac{2}{\tau} - 2\gamma + 8\gamma^2 L_S)[P(x_0) - P(x^*)] \\
EP(x_{k-1}) - P(x^*) &\leqslant \frac{(\frac{2}{\tau} - 2\gamma + 8\gamma^2 L_S)}{8\gamma^2 L_S}[P(x_0) - P(x^*)] \\
&= \left(1 - \frac{2\gamma - \frac{2}{\tau}}{8\gamma^2 L_S}\right)[P(x_0) - P(x^*)].
\end{aligned}
$$

Now, $[1 - \frac{2\gamma - \frac{2}{\tau}}{8\gamma^2 L_S}]$ can be defined as ρ. Therefore,

$$EP(x_{k-1}) - P(x^*) \leqslant \rho^{k-1}[P(x_0) - P(x^*)].$$

We have successfully proven Theorem 1. □

6 Numerical Experiments

In this section, the numerical results of the proposed method are described. To get the desired result, we used the regularized logistic regression problem for binary classification. We are given training examples of $(a_1, b_1), \ldots, (a_n, b_n)$, where $a_i \in \mathbb{R}^d$ and $b_i \in \{-1, +1\}$ and $b_i \in \{0, 1\}$. The aim is to find the optimal point $x \in \mathbb{R}^d$ as a predictor by solving

$$\min_{x\in\mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-b_i a_i^T x)) + \frac{\lambda_2}{2} \| x \|_2^2 + \lambda_1 \| x\|_1,$$

where λ_1 and λ_2 are regularization parameters. The component functions can be one of the following forms

$$f_i(x) = \log(1 + \exp(-b_i a_i^T x)) + \frac{\lambda_2}{2} \| x \|_2^2, R(x) = \lambda_1 \| x\|_1,$$

or

$$f_i(x) = \log(1 + \exp(-b_i a_i^T x)),\ \ R(x) = \frac{\lambda_2}{2} \| x \|_2^2 + \lambda_1 \| x\|_1. \tag{17}$$

We use MATLAB software for the implementation of all the considered algorithms (MATLAB v9.9.0 R2020b environment on a PC with CPU Intel Core i5 8500, 3.00 GHz, and 16GB RAM).

We compared the Prox-SAG+ algorithm with the following algorithms:

- Prox-SVRG: Algorithm Prox-SVRG in [15]
- SAG: Algorithm 1 in [10]
- Prox-SG: Eq. (8) in [15]
- Prox-FG: Algorithm 3.3 in [7]

Moreover, we investigate a stochastic dataset and two other datasets called (Machine Predictive Maintenance Classification) [18] and (Phishing Website Detector) [19]. Moreover, Cross-validation is used for generating and assessing the data. In this technique, the data divides into two groups 75 percent and 25 percent of the dataset. Then 75 percent of the data is trained for the algorithm and the rest of the data is examined. In each stage, an error is counted. λ_1 and λ_2 are chosen by the user. During the execution of the code, a stochastic dataset is normalized. So we have $\| a_i\|_2 = 1$ for each $i = 1, \ldots, n$, which causes us to get the same upper bound on the Lipschitz constants $L = L_i = \ \| a_i \|_2^2 /4$. In the implementation, we used (17) and uniform sampling of the component functions.

In Figure 2, we chose $\lambda_1, \lambda_2 = 10^{-4}$ and $m = 2n$. In addition, $\gamma = 0.1/L$ is our step size. We consider a dataset of 100 elements ($n = 100$). As we can see in Figure 2, in Prox-SAG+ after a few iterations the gap between $P(x_k)$ and P^* becomes zero, i.e. $P(x_k) - P^* = 0$. For other methods after some more iterations, the objective gap $P(x_k) - P^*$ decreases.

Figures 3 and 4 illustrate that the objective gap $P(x_k) - P(x^*)$ is lower for Prox-SAG+ than all other methods. Hence, it has better performance in comparison to other examined methods.

Table 1: Datasets and regularization parameters

Data sets	n	d	source	λ_2	λ_1
Random	100	6		10^{-4}	10^{-4}
Pre	10000	6	[18]	10^{-4}	10^{-4}
Phishing	11054	31	[19]	10^{-4}	10^{-4}

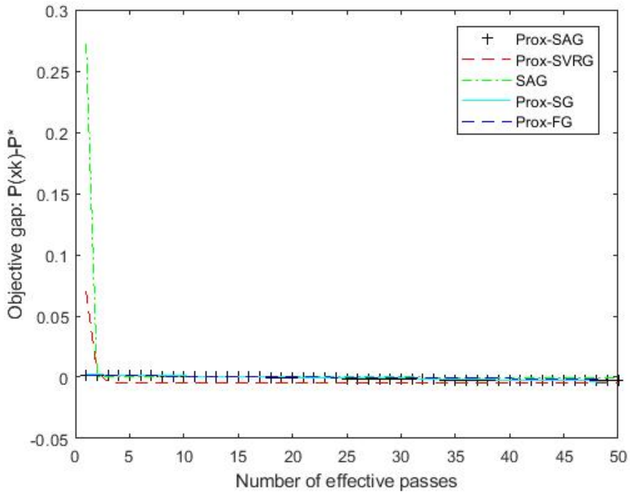

Figure 2: Comparing the objective gap of some methods with Prox-SAG+.

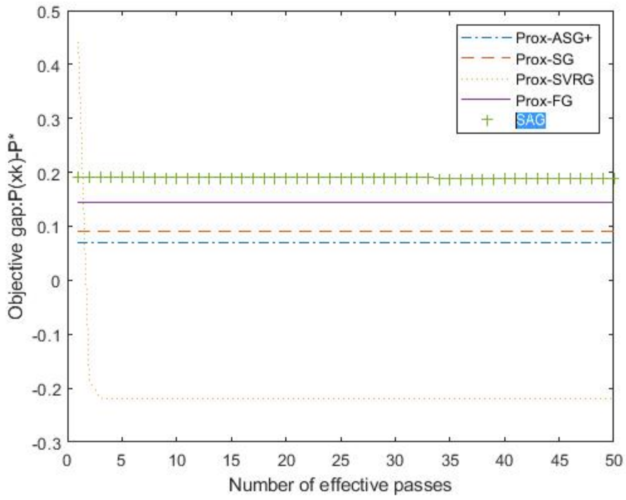

Figure 3: Comparing the objective gap of some methods with Prox-SAG+ on the Pre dataset.

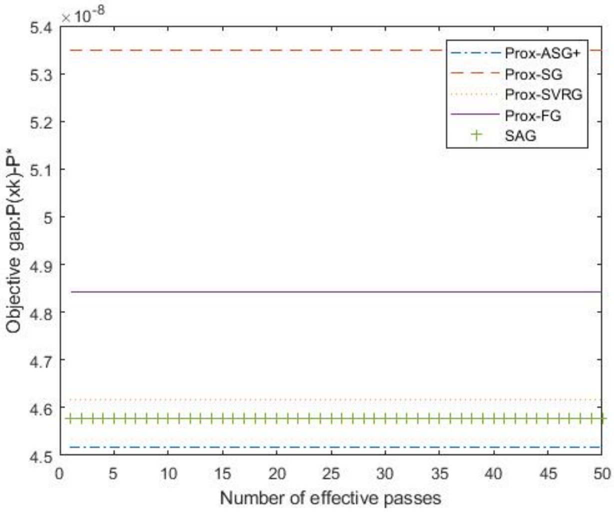

Figure 4: Comparing some methods with Prox-SAG+ on the Phishing dataset.

7 Conclusion

We proposed a new proximal stochastic method that uses a proximal operator to improve the SAG method. Additionally, the algorithm is simpler when the regularization function is not zero $\left(R(x) \neq 0\right)$. Furthermore, it outperforms Prox-SVRG since it does not require a multi-stage approach. Using the multi-stage approach makes the method more complicated than the newly proposed one. Prox-SAG+ has a geometric convergence in expectation, allowing it to solve optimization problems in machine learning effectively.

Declarations

Availability of supporting data
All data generated or analyzed during this study are included in this published paper.

Funding
This study received no funds, grants, or other financial support.

Competing interests
The authors declare no competing interests that are relevant to the content of this

paper.

Authors' contributions

The main manuscript text is collectively written by all authors.

References

[1] Hastie, T., Tibshirani, R., Friedman, J. (2009). "The elements of statistical learning", Data Mining, Second edition, Springer Series in Statistics, Springer, New York.

[2] Hu, C., Kwok, J.T., Pan, W. (2009). "Accelerated gradient methods for stochastic optimization and online learning", In Advances in Neural Information Processing Systems, 22, 781-789.

[3] Johnson, R., Zhang, T. (2013). "Accelerating stochastic gradient descent using predictive variance reduction", In Advances in Neural Information Processing Systems, 26, 315-323.

[4] Le Roux, N., Schmidt, M., Bach, F. (2012). "A stochastic gradient method with an exponential convergence rate for finite training sets", In Advances in Neural Information Processing Systems, 25, 2672-2680.

[5] Li, Z., Li, J. (2018). "A simple proximal stochastic gradient method for nonsmooth nonconvex optimization", 32nd Conference on Neural Information Processing Systems, Canada.

[6] Nesterov, Y. (2004). "Introductory lectures on convex optimization: A basic course", Kluwer, Boston.

[7] Nesterov, Y. (2013). "Gradient methods for minimizing composite functions", Mathematical Programming, Series B, 140, 125-161.

[8] Parikh, N., Boyd, S. (2013). "Proximal algorithms", Foundations and Trends R in Optimization, 1(3), 131-223.

[9] Rockafellar, R.T. (1970). "Convex analysis", Princeton University Press.

[10] Schmidt, M., Le Roux, N., Bach F. (2017). "Minimizing finite sums with the stochastic average gradient", Mathematical Programming, (1-2), Series A, 113.

[11] Shalev-Shwartz, S., Zhang, T. (2012). "Proximal stochastic dual coordinate ascent", arXiv: 1211.2772, November.

[12] Shalev-Shwartz, S., Zhang, T. (2013). "Stochastic dual coordinate ascent methods for regularized loss minimization", Journal of Machine Learning Research, 14, 567-599.

[13] Wojtowytsch, S. (2023). "Stochastic gradient descent with noise of machine learning type Part I: Discrete time analysis", Journal of Nonlinear Science 33, 45.

[14] Wojtowytsch, S. (2021). "Stochastic gradient descent with noise of machine learning type. Part II: Continuous time analysis, arXiv:2106.02588.

[15] Xiao, L., Zhang, T. (2014). "A proximal stochastic gradient method with progressive variance reduction", SIAM Journal on Optimization, 24(4), 2057-2075.

[16] Yuan, W., Hu, F. & Lu, L. (2022). "A new non-adaptive optimization method: Stochastic gradient descent with momentum and difference", Applied Intelligence, 52(4), 3939–3953.

[17] Yun, J., Lozano, A.C., Yang, E. (2020). "A general family of stochastic proximal gradient methods for deep learning", arXiv: 2007.07484 (cs).

[18] https://www.kaggle.com/datasets/shivamb/machine-predictive-maintenance-classification.

[19] https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector.

How to Cite this Article:

Saeidian, Z., Mahmoudoghli, M.,(2023). "A proximal method of stochastic gradient for convex optimization", Control and Optimization in Applied Mathematics, 8(1): 19-32. doi: 10.30473/coam.2023.64060.1205