IMPROVED SVRG FOR FINITE SUM STRUCTURE OPTIMIZATION WITH APPLICATION TO BINARY CLASSIFICATION

GUANGMEI SHAO¹, WEI XUE¹,*, GAOHANG YU² AND XIAO ZHENG¹

¹School of Computer Science and Technology, Anhui University of Technology
Maanshan 243032, China
²School of Sciences, Hangzhou Dianzi University, Hangzhou 310018, China

(Communicated by Panos M. Pardalos)

ABSTRACT. This paper looks at a stochastic variance reduced gradient (SVRG) method for minimizing the sum of a finite number of smooth convex functions, which has been involved widely in the field of machine learning and data mining. Inspired by the excellent performance of two-point stepsize gradient method in batch learning, in this paper we present an improved SVRG algorithm, named stochastic two-point stepsize gradient method. Under some mild conditions, the proposed method achieves a linear convergence rate \(O(\rho^k)\) for smooth and strongly convex functions, where \(\rho \in (0.68, 1)\). Simulation experiments on several benchmark data sets are reported to demonstrate the performance of the proposed method.

1. Introduction. Many learning tasks in machine learning and data mining involve computing a minimizer of a finite sum of functions measuring misfit over a large number of data points. One classical example is logistic regression

\[
\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(1 + \exp(-y_i \mathbf{x}_i^T \mathbf{w})\right),
\]

where \(\mathbf{w} \in \mathbb{R}^d\) is the optimization variable (usually called weight vector in learning problems), and \((\mathbf{x}_i \in \mathbb{R}^d, y_i \in \{\pm 1\})_{i=1}^{n}\) are the samples associated with a binary classification problem. Another important example is least squares regression

\[
\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \mathbf{w})^2,
\]

where \((\mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R})_{i=1}^{n}\) are the samples associated with a regression problem. Let \(\{f_1, \ldots, f_n\}\) be a sequence of functions from \(\mathbb{R}^d\) to \(\mathbb{R}\), in this paper we mainly consider the problem of optimizing a finite sample average

\[
\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{w}),
\]

2020 Mathematics Subject Classification. Primary: 65K05; Secondary: 68W27.

Key words and phrases. Machine leaning, online learning, stochastic optimization, variance reduction, two-point stepsize.

* Corresponding author: Wei Xue (E-mail: cswxue@ahut.edu.cn).
Algorithm 1 Stochastic Gradient Descent (SGD)

Require: initial point $w_0$, update frequency $T$, learning rate sequence $\{\eta_t\}$

for $t = 0, 1, \ldots, T - 1$ do
    Randomly pick $i_t \in \{1, \ldots, n\}$
    $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t)$
end for

where each $f_i(w)$ is smooth and convex. Additionally, we assume that the objective function $F(w)$ is strongly convex, which often arises due to the use of a strongly convex regularization such as the squared $l_2$-norm, $\|\cdot\|^2$, resulting in problems of the following form

$$\min_{w \in \mathbb{R}^d} \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} l_i(w),$$

where $l_i$ is a data-fitting function, and $\lambda > 0$ is the regularization parameter. This kind of problems can be put in the framework of Eq. (1) by setting $f_i(w) = \frac{\lambda}{2} \|w\|^2 + l_i(w)$, and the resulting objective function $F$ will be strongly convex on condition that each function $l_i$ is convex.

To solve Eq. (1), a standard method is full gradient descent, which uses iterations of the form

$$w_{k+1} = w_k - \eta_k \nabla F(w_k) = w_k - \frac{\eta_k}{n} \sum_{i=1}^{n} \nabla f_i(w_k),$$

where $\eta_k > 0$ is the so-called learning rate (a.k.a. stepsize). But the full gradient descent method requires evaluation of $n$ derivatives at each step, which is too expensive, and it would be unappealing when the number of data points $n$ (a.k.a. training samples) is large. Recently, stochastic gradient descent (SGD) method has received extensive attention due to its potential for solving such problems arising in machine learning and other applications. The main advantage of SGD is that each step only requires a single derivative, and thus its iteration cost is independent of $n$. Suppose we have the most up-to-date weight vector $w_t$ at the $t$-th round \footnote{We get used to using $k$ to represent iteration index in batch optimization/learning. Here, to make a clear distinction, we prefer $t$ in online/stochastic optimization/learning.}. Whenever the sample pair $(x_t, y_t)$ is available, we can evaluate the gradient of each component function and then compute $w_{t+1}$ based on the information. In brief, the basic SGD method for solving Eq. (1) takes the following update form

$$w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),$$

where $\eta_t > 0$, and $i_t$, the index of $t$-th sample, is randomly chosen among the set $\{1, \ldots, n\}$. The pseudo code of basic SGD is given in Algorithm 1. Essentially, SGD belongs to stochastic approximation, which can date back to the work of \cite{16}. Here, $\nabla f_{i_t}$ is referred to as stochastic gradient, and it is usually assumed that $\nabla f_{i_t}$ is an unbiased estimator of $\nabla F$, i.e.,

$$\mathbb{E}[\nabla f_{i_t}(w_t)] = \nabla F(w_t).$$

Hence, the expectation $\mathbb{E}[w_{t+1}|w_t]$ equals Eq. (2). However, the major disadvantage of SGD is that the randomness may introduce variance, which is caused by...
Algorithm 2 Stochastic Variance Reduced Gradient (SVRG)

Require: initial point $\tilde{w}_0$, update frequency $T$, learning rate $\eta$

for $k = 0, 1, \ldots$ do
    Compute full gradient $\bar{g}_k = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{w}_k)$
    $w_0 = \tilde{w}_k$
    for $t = 0, 1, \ldots, T - 1$ do
        Randomly pick $i_t \in \{1, \ldots, n\}$
        $w_{t+1} = w_t - \eta(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + \bar{g}_k)$
    end for
    Option I: $\tilde{w}_{k+1} = w_T$
    Option II: $\tilde{w}_{k+1} = w_t$ for randomly chosen $t \in \{1, \ldots, T - 1\}$
end for

the fact that stochastic gradient $\nabla f_{i_t}(w_t)$ equals the full gradient $\nabla F(w_t)$ in expectation but each $\nabla f_{i_t}(w_t)$ is different. Particularly, if it has a large variance, the convergence speed will be slow.

1.1. Background of SVRG. More recently, the convergence speed of SGD has been further improved by using variance reduction technique [7, 9, 18, 19, 21]. These research results show that SGD can converge much faster if one makes a better choice of the stochastic gradient $\nabla f_{i_t}(w_t)$ so that its variance $\mathbb{E}[\nabla f_{i_t}(w_t) - \nabla F(w_t)]$ reduces as $t$ increases. One particular approach to reducing the variance is the stochastic variance reduced gradient (SVRG) method [7] described as follows.

Keep a snapshot $\tilde{w}$ after every $T$ stochastic update steps and maintain the average gradient $\nabla F(\tilde{w}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{w})$. Let $\nu_t \triangleq \nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}) + \nabla F(\tilde{w})$ be the stochastic gradient, then the corresponding update rule of SVRG can be illustrated as

$$w_{t+1} = w_t - \eta_t \nu_t. \quad (4)$$

Note that the expectation of $\nabla f_{i_t}(\tilde{w}) - \nabla F(\tilde{w})$ over $i_t$ is 0, and the expectation of $\nabla f_{i_t}(w_t)$ over $i_t$ is $\nabla F(w_t)$. We thus have

$$\mathbb{E}[w_{t+1}|w_t] = w_t - \eta_t \nabla F(w_t).$$

In SGD, we need to assume the sample gradient is an unbiased estimator of the full gradient, resulting in $\mathbb{E}[w_{t+1}|w_t] = w_t - \eta_t \nabla F(w_t)$. However, we can obtain this result in SVRG without this assumption. That is to say, SVRG can explicitly reduce the variance. Moreover, unlike SGD, the learning rate in SVRG does not have to decay, resulting in faster convergence. The entire process of SVRG is described in Algorithm 2.

1.2. Motivation and organization. In SVRG, the learning rate $\eta$ is defined beforehand. According to [7], the choice of $\eta$ depends on the Lipschitz constant of $F$, which is usually difficult to estimate in practice. As argued by [17], one important issue regarding to a stochastic optimization algorithm is how to choose an appropriate learning rate while running the algorithm.

This paper follows the trend of improving SVRG, which is aimed at using the two-point stepsize gradient method to automatically compute the learning rate for SVRG with very little additional computation and memory cost. We prove that the improved SVRG algorithm converges linearly for smooth and strongly convex
learning problems. We also utilize this improved algorithm for teaching binary classification learning problems based on the $l_2$-regularized logistic regression.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the two-point stepsize gradient method for unconstrained optimization in batch learning. The main contribution of this paper, the adaptation of two-point stepsize for SVRG, is also presented in Section 2 along with the convergence analysis. In Section 3, we provide numerical experiments to illustrate the efficiency of the proposed method. Finally, we conclude this paper in Section 4.

2. Algorithm. In this section, to introduce our algorithm, we first review the two-point stepsize briefly. Firstly, we recall some definitions frequently used in convex analysis.

Definition 2.1. [Strong convexity [2]] A function $f : \mathbb{R}^d \to \mathbb{R}$ is strongly convex, if there exists a $\mu > 0$ such that $\nabla^2 f(w) \succeq \mu I$ for all $w \in \mathbb{R}^d$, where $I$ is the identity matrix.

Remark 1. For $w, v \in \mathbb{R}^d$, we have

$$f(w) = f(v) + \nabla f(v)^T (w - v) + \frac{1}{2} (w - v)^T \nabla^2 f(u) (w - v)$$

for some $u$ on the line segment $[v, w]$. By the strong convexity definition, the last term on the righthand side is at least $\mu \|w - v\|^2 / 2$. Further, the following inequality

$$f(w) \geq f(v) + \nabla f(v)^T (w - v) + \frac{\mu}{2} \|w - v\|^2$$

holds for all $v$ and $w$ in $\mathbb{R}^d$.

Definition 2.2. [Lipschitzness [2]] A function $f : \mathbb{R}^d \to \mathbb{R}$ is Lipschitz with constant $\rho$, if $\|f(v) - f(w)\| \leq \rho \|v - w\|$.

Definition 2.3. [Co-coercivity property of gradient [11]] If function $f : \mathbb{R}^d \to \mathbb{R}$ is convex, and the corresponding gradient is $L$-Lipschitz, then for all $v, w \in \mathbb{R}^d$, we have $\|\nabla f(v) - \nabla f(w)\|^2 \leq L (\nabla f(v) - \nabla f(w))^T (v - w)$.

2.1. Preparation: Two-point stepsize. Consider the following unconstrained optimization problem

$$\min f(w), w \in \mathbb{R}^d,$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a continuous differentiable function. The negative gradient method for solving Eq. (5) usually takes the negative gradient as its search direction and uses iterations of the form

$$w_{k+1} = w_k - \eta_k g_k,$$

where $g_k = \nabla f(w_k)$. Note that the matrix $B_k \triangleq \frac{1}{\eta_k} I$ can be regarded as an approximation to the Hessian matrix of $f$ at $w_k$. Barzilai and Borwein [1] chose $\eta_k$ such that $B_k$ has a certain quasi-Newton property, yielding the so-called long stepsize

$$\eta_k^\text{long} = \frac{\mathbf{s}_k^T \mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}},$$

where $\mathbf{s}_{k-1} = w_k - w_{k-1}$, and $\mathbf{y}_{k-1} = g_k - g_{k-1}$. An alternative approach is to approximate the inverse Hessian matrix by the matrix $H_k \triangleq \eta_k I$, resulting in the
so-called short stepsize
\[ \eta_k^{\text{short}} = \frac{s_k^{T} y_{k-1}}{y_{k-1}^{T} y_{k-1}} \]  

In [15], Raydan has shown that the two-point stepsize gradient method with either stepsize formula is globally convergent in the case of strictly convex quadratic. Dai and Fletcher[5] have shown that Barzilai and Borwein’s method with either Eq. (7) or Eq. (8) is likely to be asymptotically R-superlinearly convergent in the 3-dimensional case. The negative gradient method Eq. (6) with Eq. (7) or Eq. (8) has achieved a lot of attention since it performs much more better than the classical steepest descent method. It is worth mentioning that there has been some alternate use of the two-point stepsize formulas Eq. (7) and Eq. (8), such as the alternative stepsize [24] and the adaptive stepsize [6]. They are given by
\[ \eta_k = \begin{cases} 
\eta_k^{\text{long}} & \text{if } k \text{ is even,} \\
\eta_k^{\text{short}} & \text{if } k \text{ is odd,} 
\end{cases} \quad \eta_k = \begin{cases} 
\frac{\eta_k^{\text{short}}}{\eta_k^{\text{long}}} & \text{if } \frac{\eta_k^{\text{short}}}{\eta_k^{\text{long}}} \leq \kappa, \\
\eta_k^{\text{long}} & \text{otherwise,}
\end{cases} \]

respectively, where \( \kappa > 0 \) is a constant. Some recent developments of the two-point stepsize gradient method can be found in [3, 4, 8, 20, 22, 23, 25].

2.2. Algorithm. Note that the two-point stepsize formulas both Eq. (7) and Eq. (8) do not need any parameter nd they can be computed while running the algorithm. As a result, we are motivated to propose an improved SVRG that compute the learning rate automatically.

The improved stochastic variance reduction algorithm, stochastic two-point stepsize gradient method, for solving the finite sum structure minimization problem Eq. (1) is outlined in Algorithm 3.

Algorithm 3 Stochastic Two-point Stepsize Gradient (STSG)

Require: initial point \( \tilde{w}_0 \), initial learning rate \( \eta_0 \), update frequency \( T \), threshold parameter \( \epsilon \in (0, 1) \)

for \( k = 0, 1, \ldots \) do

Compute full gradient \( \nabla F(\tilde{w}_k) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{w}_k) \)

if \( k > 0 \) then

\[ \eta_k^{(a)} = \frac{\|\tilde{w}_k - \tilde{w}_{k-1}\|^2}{(\tilde{w}_k - \tilde{w}_{k-1})^T (\tilde{w}_k - \tilde{w}_{k-1})} \]
\[ \eta_k^{(b)} = \frac{(\tilde{w}_k - \tilde{w}_{k-1})^T (\tilde{g}_k - \tilde{g}_{k-1})}{\|\tilde{g}_k - \tilde{g}_{k-1}\|^2} \]

\[ \eta_k = \begin{cases} 
\eta_k^{(b)}/T, & \text{if } (\tilde{w}_k - \tilde{w}_{k-1})^T (\tilde{g}_k - \tilde{g}_{k-1}) < \epsilon \\
\eta_k^{(a)}/T, & \text{otherwise}.
\end{cases} \]

end if

\( \tilde{w}_0 = \tilde{w}_k \)

for \( t = 0, 1, \ldots, T - 1 \) do

Select \( i_t \) randomly among the set \( \{1, \ldots, n\} \)

Compute \( \nabla f_{i_t}(\tilde{w}_t) \) and \( \nabla f_{i_t}(\tilde{w}_k) \) based on the \( i_t \)-th sample

\( w_{t+1} = w_t - \eta_k (\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + \nabla F(\tilde{w}_k)) \cdots (\star) \)

end for

\( \tilde{w}_{k+1} = w_T \)

end for

For this algorithm, we have some remarks.
Remark 2. Our learning rate strategy can be naturally incorporated to other SVRG variants, such as [10, 12, 13].

Remark 3. If we set \( \eta_k = \eta \) in Algorithm 3, then STSG reduces to the original SVRG method.

2.3. Convergence analysis. In this section, we give analysis of the convergence property of Algorithm 3. Our results will be based on the following assumptions.

Assumption 1. Assume that Eq. (3) holds for any \( \omega \), and the average function \( F \) is \( \mu \)-strongly convex.

Assumption 2. Assume that each component function \( f_i \) is continuously differentiable and the gradient function \( \nabla f_i \) has Lipschitz constant \( L_i \). Additionally, we denote \( \sup L \) the supremum of the support of \( L_i \), i.e., the smallest \( \bar{L} \) such that \( L_i \leq \bar{L} \).

Assumption 3. Assume that \( \omega_* \) is the optimal solution of Eq. (1), i.e., \( \omega_* = \arg \min_{\omega \in \mathbb{R}^d} F(\omega) \).

The following lemmas are the key to the analysis of our method.

Lemma 2.4. For \( k, T > 0 \), we have \( \frac{1}{TT} \leq \eta_k \leq \frac{1}{T\mu} \).

Proof. Using the strong convexity of \( F \), we obtain

\[
F(\tilde{w}_k) \geq F(\tilde{w}_{k-1}) + \tilde{g}_{k-1}^\top(\tilde{w}_k - \tilde{w}_{k-1}) + \frac{\mu}{2}\|\tilde{w}_k - \tilde{w}_{k-1}\|^2
\]

and

\[
F(\tilde{w}_{k-1}) \geq F(\tilde{w}_k) + \tilde{g}_{k-1}^\top(\tilde{w}_{k-1} - \tilde{w}_k) + \frac{\mu}{2}\|\tilde{w}_{k-1} - \tilde{w}_k\|^2,
\]

\( \forall \tilde{w}_k, \tilde{w}_{k-1} \). From the above two inequalities, we have

\[
\mu\|\tilde{w}_k - \tilde{w}_{k-1}\|^2 \leq (\tilde{g}_k - \tilde{g}_{k-1})^\top(\tilde{w}_k - \tilde{w}_{k-1}) \quad (9)
\]

It follows that \( \eta_k^{(a)} \) can be bounded as

\[
\eta_k^{(a)} = \frac{\|\tilde{w}_k - \tilde{w}_{k-1}\|^2}{(\tilde{w}_k - \tilde{w}_{k-1})^\top(\tilde{g}_k - \tilde{g}_{k-1})} \leq \frac{\|\tilde{w}_k - \tilde{w}_{k-1}\|^2}{\mu\|\tilde{w}_k - \tilde{w}_{k-1}\|^2} = \frac{1}{\mu}.
\]

In addition, since the gradient of each component function \( f_i(\omega) \) is \( L_i \)-Lipschitz, it is easy to show that \( \eta_k^{(a)} \geq 1/\bar{L} \). Moreover, by using Definition 2.3 and Assumption 2, we get

\[
\|\tilde{g}_k - \tilde{g}_{k-1}\|^2 \leq \bar{L}(\tilde{g}_k - \tilde{g}_{k-1})^\top(\tilde{w}_k - \tilde{w}_{k-1}),
\]

which implies that \( \eta_k^{(b)} \geq 1/\bar{L} > 0 \). By using the inequation \( \|a\|^2\|b\|^2 \geq \|ab\|^2 \), we have

\[
\frac{\eta_k^{(a)}}{\eta_k^{(b)}} = \frac{\|\tilde{w}_k - \tilde{w}_{k-1}\|^2\|\tilde{g}_k - \tilde{g}_{k-1}\|^2}{(\tilde{w}_k - \tilde{w}_{k-1})^\top(\tilde{g}_k - \tilde{g}_{k-1})\|^2} \geq 1.
\]
Suppose that $\tilde{w}_k \neq \tilde{w}_{k-1}$, then combined with Eq. (9) and the definitions of $\eta_k^{(a)}$ and $\eta_k^{(b)}$, we have $\eta_k^{(a)} > 0$ and $\eta_k^{(b)} > 0$, which means $\eta_k^{(a)} \geq \eta_k^{(b)}$. It follows that

$$\frac{1}{L} \leq \eta_k^{(b)} \leq \eta_k^{(a)} \leq \frac{1}{\mu}. \quad (10)$$

Lemma 2.4 then follows immediately by dividing Eq. (10) by $T$.

**Lemma 2.5.** Let $\nu_i \triangleq \nabla f_i(\tilde{w}_1) - \nabla f_i(\tilde{w}_k) + \nabla F(\tilde{w}_k)$ in (▲), then we have $E\|\nu_i\|^2 \leq 2\bar{L}(w_t - w_*)^T\nabla F(w_t) + 8\bar{L}^2\|\tilde{w}_k - w_*\|^2$.

**Proof.**

$$E\|\nu_i\|^2 = E\|\nabla f_i(w_t) - \nabla f_i(\tilde{w}_k) + \nabla F(\tilde{w}_k)\|^2$$

(using $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$)

$$\leq 2E\|\nabla f_i(w_t) - \nabla f_i(w_*)\|^2 + 2E\|\nabla f_i(\tilde{w}_k) - \nabla f_i(w_*) - \nabla F(\tilde{w}_k)\|^2$$

(using $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ and $\nabla F(w_*) = 0$)

$$\leq 2E\|\nabla f_i(w_t) - \nabla f_i(w_*)\|^2 + 4E\|\nabla f_i(\tilde{w}_k) - \nabla f_i(w_*)\|^2$$

+ $4\|\nabla F(\tilde{w}_k) - \nabla F(w_*)\|^2$

(using the Lipschitzness of $\nabla f_i$)

$$\leq 2L_i E[(w_t - w_*)^T(\nabla f_i(w_t) - \nabla f_i(w_*))] + 4L_i^2\|\tilde{w}_k - w_*\|^2$$

+ $4\tilde{L}_i^2\|\tilde{w}_k - w_*\|^2$

(noting that $E[\nabla f_i(w_t)] = \nabla F(w_t), \nabla F(w_*) = 0$, and $\tilde{L}$ is the supremum of the support of $L_i$)

$$\leq 2\bar{L}(w_t - w_*)^T\nabla F(w_t) + 8\bar{L}^2\|\tilde{w}_k - w_*\|^2.$$

This finishes the proof of Lemma 2.5.

**Lemma 2.6.** Under Assumption 3, we have

$$E\|w_{t+1} - w_*\|^2 \leq (1 - 2\mu\eta_k(1 - \bar{L}\eta_k))\|w_t - w_*\|^2$$

+ $8\bar{L}^2\eta_k^2\|\tilde{w}_k - w_*\|^2$.

**Proof.**

$$E\|w_{t+1} - w_*\|^2 = E\|w_t - \eta_k\nu_t - w_*\|^2$$

$$= \|w_t - w_*\|^2 - 2\eta_k E[(w_t - w_*)^T\nu_t] + \eta_k^2 E\|\nu_t\|^2$$

(noting that $E[\nu_t] = \nabla F(w_t)$)

$$= \|w_t - w_*\|^2 - 2\eta_k(w_t - w_*)^T\nabla F(w_t) + \eta_k^2 E\|\nu_t\|^2$$

(using Lemma 2.5)

$$\leq \|w_t - w_*\|^2 - 2\eta_k(w_t - w_*)^T\nabla F(w_t)$$

+ $2\tilde{L}\eta_k^2(w_t - w_*)^T\nabla F(w_t) + 8\bar{L}^2\eta_k^2\|\tilde{w}_k - w_*\|^2$

$$= \|w_t - w_*\|^2 - 2\eta_k(1 - \bar{L}\eta_k)(w_t - w_*)^T\nabla F(w_t)$$

+ $8\bar{L}^2\eta_k^2\|\tilde{w}_k - w_*\|^2$

$$\leq (1 - 2\mu\eta_k(1 - \bar{L}\eta_k))\|w_t - w_*\|^2 + 8\bar{L}^2\eta_k^2\|\tilde{w}_k - w_*\|^2.$
This finishes the proof of Lemma 2.6 \footnote{In the last inequality, we used the strong convexity of \( F \) and \( \nabla F(w_*) = 0 \). By using the strong convexity of \( F \) at \( w_t \) and \( w_* \), we have
\[
F(w_t) \geq F(w_*) + \nabla F(w_*)^\top (w_t - w_*) + \frac{\mu}{2} \| w_t - w_* \|^2, \tag{11}
\]
and
\[
F(w_*) \geq F(w_t) + \nabla F(w_t)^\top (w_* - w_t) + \frac{\mu}{2} \| w_t - w_* \|^2. \tag{12}
\]
Eq. (11) plusing Eq. (12) yields
\[
\mu \| w_t - w_* \|^2 \leq (\nabla F(w_t) - \nabla F(w_*)^\top (w_t - w_*)
\]
(\text{using } \nabla F(w_*) = 0)
\[
= \nabla F(w_*)^\top (w_t - w_*).
\tag{13}
\]
Combining Eq. (13) with the fourth equality gives the last inequality.}

By applying the above inequality over \( t \) recursively, we obtain the following corollary.

**Corollary 1.** Let \( \delta_k \triangleq (1 - 2\mu \eta_k (1 - \bar{L} \eta_k))^T \frac{4L^2 \eta_k}{\mu (1 - \bar{L} \eta_k)} \), then we have
\[
E \| \tilde{w}_{k+1} - w_* \|^2 \leq \delta_k \| \tilde{w}_k - w_* \|^2.
\]

**Proof.** Noting that \( \tilde{w}_k = w_0 \) and \( \tilde{w}_{k+1} = w_T \) and using Lemma 2.6, we have
\[
E \| \tilde{w}_{k+1} - w_* \|^2 \leq (1 - 2\mu \eta_k (1 - \bar{L} \eta_k))^T \| \tilde{w}_k - w_* \|^2
\]
\[
+ 8\bar{L}^2 \eta_k^2 \{ 1 + [(1 - 2\mu \eta_k (1 - \bar{L} \eta_k))]^2
\]
\[
+ \cdots
\]
\[
+ [(1 - 2\mu \eta_k (1 - \bar{L} \eta_k))]^{T-1} \| \tilde{w}_k - w_* \|^2
\]
\[
< \left[ (1 - 2\mu \eta_k (1 - \bar{L} \eta_k))^T + \frac{4L^2 \eta_k}{\mu (1 - \bar{L} \eta_k)} \right] \| \tilde{w}_k - w_* \|^2.
\]

This finishes the proof of Corollary 1. \( \square \)

The following theorem establishes the convergence property of STSG.

**Theorem 2.7.** Consider the STSG method in Algorithm 3 under Assumptions 1 and 2. Let \( \theta \triangleq (1 -\frac{\mu}{\bar{L}})/2 \), and if
\[
T > \max \left\{ \frac{2}{\log(1 - 2\theta) + 2\mu/\bar{L}}, \frac{4L}{\theta \mu^2} + \frac{\bar{L}}{\mu} \right\},
\]
then we have
\[
E \| \tilde{w}_k - w_* \|^2 \leq (1 - \theta)^k \| \tilde{w}_0 - w_* \|^2.
\]
Table 1. Details of data sets.

| Data set | # of training samples | # of test samples | dimension |
|----------|-----------------------|-------------------|-----------|
| a1a      | 1605                  | 30956             | 123       |
| a9a      | 32561                 | 16281             | 123       |
| w1a      | 2477                  | 47272             | 300       |
| w8a      | 49749                 | 14951             | 300       |

Proof. According to Lemma 2.4, $\delta_k$ defined in Corollary 1 can be bounded as

$$
\delta_k = (1 - 2\mu\eta_k(1 - \bar{L}\eta_k))^T + \frac{4\bar{L}^2\eta_k}{\mu(1 - \bar{L}\eta_k)}
\leq [1 - \frac{2\mu}{T\bar{L}}(1 - \frac{\bar{L}}{T\mu})]^T + \frac{4\bar{L}^2}{T\mu^2[1 - (L/\mu)T]} \\
\leq \exp \left\{ - \frac{2\mu}{L} \left(1 - \frac{\bar{L}}{T\mu}\right) \right\} + \frac{4\bar{L}^2}{T\mu^2 - L\mu} \\
= \exp \left\{ - \frac{2\mu}{L} + \frac{2}{T} \right\} + \frac{4\bar{L}^2}{T\mu^2 - L\mu}.
$$

Because $T > \max \left\{ \frac{2}{\log(1 - 2\theta) + 2\mu/L} \frac{4L}{2\theta^2 \mu^2} + \frac{L}{\mu} \right\}$, so we obtain

$$
\delta_k < \exp \left\{ - \frac{2\mu}{L} + \log(1 - 2\theta) + \frac{2\mu}{L} \right\} + \frac{4\bar{L}^2}{4\bar{L}^2/\theta + L\mu - L\mu} \\
= 1 - \theta.
$$

Additionally, noting that $\theta = (1 - e^{-\mu})/2 \in (0,0.32)$, and from Corollary 1, it follows that

$$
E\|\bar{w}_k - w_*\|^2 \leq (1 - \theta)^k \|\bar{w}_0 - w_*\|^2,
$$

which means that the STSG method converges linearly in expectation. \(\square\)

3. Numerical experiments. In this section, we present some numerical experiments to verify the effectiveness of the stochastic two-point stepsize gradient method (STSG). We firstly compare it with SGD and SVRG on the $l_2$-regularized logistic regression with regularization parameter $\lambda > 0$

$$
\min_w F(w) = \frac{1}{n} \sum_{i=1}^{n} (1 + \exp(-y_i x_i^T w)) + \frac{\lambda}{2} \|w\|^2. \tag{14}
$$

For both SVRG and STSG, we set $T = 2n$ as suggested in [7]. For the baseline method SGD, we choose a conventional learning rate, that is, $\eta_t = 1/(t + 1)$. See Chapter 6 in [14] for more information on the choice of $\eta_t$. Table 1 provides details of the data sets used in our experiments, which can be downloaded from the website http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets. In the second experiment, we compare STSG with the well-known batch learning model SVM solved by the LIBSVM tool (a library for support vector machines). All experiments are implemented in PyCharm on a PC with Windows 7 operating system.
In the first experiment, we firstly analyze the impact of the algorithm parameter $\epsilon$ values in STSG on the objective loss and the quality of classification. The resulting
Figure 4. Evolutions of objective loss and test classification accuracy w.r.t. epochs for Eq. (14) on data set “w8a”.

Figure 5. Evolutions of objective loss and test classification accuracy w.r.t. epochs for Eq. (14) on data set “a1a”.

Figure 6. Evolutions of objective loss and test classification accuracy w.r.t. epochs for Eq. (14) on data set “w1a”.

analysis is shown in Figures 1 and 2. In all subfigures, the x-axis denotes the number of epochs $k$. In Figures 1 (a) and 2 (a), the y-axis denotes the objective loss, and
in Figures 1 (b) and 2 (b), the y-axis denotes test classification accuracy. From the resulting analysis, we can see that STSG is not too sensitive to \( \epsilon \) values. Overall, \( \epsilon = 0.1 \) seems to be preferable. Hence, we set \( \epsilon = 0.1 \) for the following experiments.

We next show the comparison results of SGD, SVRG and STSG. Experimental results are shown in Figures 3\textendash}6. From Figures 3 and 4, we see that the performance of SGD is worst. We can also see that the SVRG method is sensitive to the choice of learning rate \( \eta_0 \), and its numerical performance is not stable. However, the proposed STSG method is not sensitive to the choice of \( \eta_0 \). In addition, the objective loss and test classification accuracy of STSG tend to be stable after only a few epochs. Overall, the proposed method STSG can give a comparable or even better numerical performance than SVRG. Figures 3\textendash}6 further show that STSG is very sensitive to the initial stepsize \( \eta_0 \). Different \( \eta_0 \) has different numerical performance. Overall, STSG possesses better performance when \( \eta_0 \) is smaller. More experiments are reported in Figures 7 and 8 to verify this characteristic.

In the second experiment, we compare STSG with the well-known SVM model. SVM is a batch processing optimization method, which uses all samples at each iteration to train the model parameter, while STSG randomly employs a sample per iteration. That is, the amount of the sample information used by STSG is far
less than SVM. Comparison results reported in Figure 9 show that the proposed approach STSG can obtain a comparable performance to SVM.

![Comparison results between SVM and STSG on test classification accuracy on four test data sets.](image)

**Figure 9.** Comparison results between SVM and STSG on test classification accuracy on four test data sets.

In short, the preliminary experiment on binary classification indicates that STSG provides a valid approach for solving finite sum structure minimization problems. It can obtain a comparable performance to the batch learning method such as SVM and process the best performance followed by SVRG and SGD.

4. **Conclusions.** We proposed in this paper an improved Stochastic Variance Reduction Gradient (SVRG) algorithm, named Stochastic Two-point Stepsize Gradient (STSG) method, for solving finite sum structure optimization problems in an online setting from the viewpoint of optimization. We tried to use the well-known two-point stepsize gradient strategy to automatically compute the learning rate for SVRG. Moreover, we established a linear convergence rate in expectation for STSG under some mild conditions when the objective is smooth and strongly convex. Comparison results further demonstrated the effectiveness of the proposed method.

**Acknowledgments.** The authors thank the editor and the reviewers for their numerous helpful comments, which greatly improved the presentation of this paper. W. Xue would like to thank Dr. C. Tan from The Chinese University of Hong Kong for the help of the proof. This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61806004, U1636220, 11661007, 61602482 and 61472423), the Natural Science Foundation of Zhejiang Province, China (Grant No. LD19A010002), and the Major Technologies R & D Special Program of Anhui Province, China (Grant No. 16030901060).

**REFERENCES**

[1] J. Barzilai and J. M. Borwein, Two-point step size gradient methods, *IMA J. Numer. Anal.*, 8 (1988), 141–148.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, 2004.

[3] Y. Chen, L. Qi and Q. Wang, Computing extreme eigenvalues of large scale hankel tensors, *J. Sci. Comput.*, 68 (2016), 716–738.
[4] W. Cheng and Y. Dai, Gradient-based method with active set strategy for $l_1$ optimization, Math. Comp., 87 (2018), 1283–1305.
[5] Y. Dai and R. Fletcher, On the asymptotic behaviour of some new gradient methods, Math. Program., 103 (2005), 541–559.
[6] Y. Dai and R. Fletcher, Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming, Numer. Math., 100 (2005), 21–47.
[7] R. Johnson and T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in Adv. Neural Inf. Process. Syst., (2013), 315–323.
[8] H. Liu, Z. Liu and X. Dong, A new adaptive Barzilai and Borwein method for unconstrained optimization, Optim. Lett., 12 (2018), 845–873.
[9] J. Mairal, Incremental majorization-minimization optimization with application to large-scale machine learning, Optim. Lett., 25 (2015), 829–855.
[10] E. Min, Y. Zhao, J. Long, C. Wu, K. Li and J. Yin, SVRG with adaptive epoch size, in Proc. IEEE Int. Joint Conf. Neural Netw., (2017), 2935–2942.
[11] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Springer, Boston, 2004.
[12] A. Nitanda, Accelerated stochastic gradient descent for minimizing finite sums, in Proc. Int. Conf. Artif. Intell. Statist., (2016), 195–203.
[13] A. Nitanda, Stochastic proximal gradient descent with acceleration techniques, in Adv. Neural Inf. Process. Syst., (2016), 1574–1582.
[14] W. B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality, John Wiley and Sons, New Jersey, 2007.
[15] M. Raydan, On the Barzilai and Borwein choice of steplength for the gradient method, IMA J. Numer. Anal., 13 (1993), 321–326.
[16] H. Robbins and S. Monro, A stochastic approximation method, Ann. Math. Statist., 22 (1951), 400–407.
[17] N. L. Roux, M. Schmidt and F. Bach, A stochastic gradient method with an exponential convergence rate for finite training sets, in Adv. Neural Inf. Process. Syst., (2012), 2663–2671.
[18] Z. Shen, H. Qian, T. Zhou and T. Mu, Adaptive variance reducing for stochastic gradient descent, in Proc. Int. Joint Conf. Artif. Intell., (2016), 1990–1996.
[19] C. Tan, S. Ma, Y. Dai and Y. Qian, Barzilai-Borwein step size for stochastic gradient descent, in Adv. Neural Inf. Process. Syst., (2016), 685–693.
[20] Z. Wen, C. Yang, X. Liu and Y. Zhang, Trace-penalty minimization for large-scale eigenspace computation, J. Sci. Comput., 66 (2016), 1175–1203.
[21] L. Xiao and T. Zhang, A proximal stochastic gradient method with progressive variance reduction, SIAM J. Optim., 24 (2014), 2057–2075.
[22] W. Xue, W. Zhang and J. Ren, Online learning based on stochastic spectral gradient, Comput. Sci., 43 (2016), 47–51.
[23] G. Yu and S. Niu, Multivariate spectral gradient projection method for nonlinear monotone equations with convex constraints, J. Ind. Manag. Optim., 9 (2013), 117–129.
[24] B. Zhou, L. Gao and Y. Dai, Gradient methods with adaptive step-sizes, Comput. Optim. Appl., 35 (2006), 69–86.
[25] R. Zhou, X. Shen and L. Niu, A fast algorithm for nonsmooth penalized clustering, Neurocomputing, 273 (2018), 583–592.

Received May 2018; revised November 2018.

E-mail address: yixiasgm@gmail.com
E-mail address: csxue@ahut.edu.cn
E-mail address: maghyu@163.com
E-mail address: xzheng@ahut.edu.cn