STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **12**, January 2024, pp 1–14.
Published online in International Academic Press (www.IAPress.org)

A Modified Inexact SARAH Algorithm With Stabilized Barzilai-Borwein Step-Size in Machine Learning

Fusheng Wang [1,*], Yiming Yang [2], Xiaotong Li [1], Ovanes Petrosian[3]

[1] *School of Mathematics and Statistics, Taiyuan Normal University, China*
[2] *School of Mathematics and Computational Science, Xiangtan University,China*
[3] *Department of Mathematical Modeling of Energetic Systems, Saint-Petersburg State University, Russia*

Abstract The Inexact SARAH (iSARAH) algorithm as a variant of SARAH algorithm, which does not require computation of the exact gradient, can be applied to solving general expectation minimization problems rather than only finite sum problems. The performance of iSARAH algorithm is frequently affected by the step size selection, and how to choose an appropriate step size is still a worthwhile problem for study. In this paper, we propose to use the stabilized Barzilai-Borwein (SBB) method to automatically compute step size for iSARAH algorithm, which leads to a new algorithm called iSARAH-SBB. By introducing this adaptive step size in the design of the new algorithm, iSARAH-SBB can take better advantages of both iSARAH and SBB methods. We analyse the convergence rate and complexity of the modified algorithm under the usual assumptions. Numerical experimental results on standard data sets demonstrate the feasibility and effectiveness of our proposed algorithm.

Keywords Stochastic gradient algorithms, Mini-batches, Stochastic optimization, BB method

AMS 2010 subject classifications 90C15, 90C25, 90C30

DOI: 10.19139/soic-2310-5070-1712

1. Introduction

We consider the following type of stochastic optimization problem in the context of large scale machine learning:

$$\min_{w\in\mathbb{R}^d} \{F(w) = \mathbb{E}[f(w;\xi)]\}, \tag{1}$$

where ξ is a random variable, f is a convex function which stands for the loss function, and $w \in \mathbb{R}^d$ is the Parameter to be adjusted in machine learning. One of the most popular applications of this problem is expected risk minimization in supervised learning. In this case, random variable ξ represents a random data sample (x, y), or a set of such samples $\{(x_i, y_i)\}_{i\in I}$. We consider a set of realizations $\{\xi_{[i]}\}_{i=1}^n$ of ξ corresponding to a set of random samples $\{(x_i, y_i)\}_{i=1}^n$, and define $f_i(w) := f(w;\xi_{[i]})$. Then the sample average approximation of $F(w)$, known as the empirical risk in supervised learning, can be written as

$$\min_{w\in\mathbb{R}^d} \left\{ F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w) \right\}, \tag{2}$$

*Correspondence to: Fusheng Wang (Email: fswang2005@163.com). School of Mathematics and Statistics, Taiyuan Normal University. Jinzhong, Shanxi Province, China (030619).

ISSN 2310-5070 (online) ISSN 2311-004X (print)

where n is the sample size and $d \ll n$. Throughout this paper, we assume that each f_i is convex and differentiable. For a given training set $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$, the least squares regression model is written as $f_i(\omega) = \left(x_i^T\omega - y_i\right)^2 + \frac{\lambda}{2}\|\omega\|^2$, where λ is a regularization parameter and $\|\cdot\|$ denotes the l_2-norm; and the l_2-regularized logistic regression loss for binary classification problems is written as $f_i(\omega) = \log\left(1 + \exp\left[-y_i x_i^T\omega\right]\right) + \frac{\lambda}{2}\|\omega\|^2$ $(y_i \in \{-1, 1\})$.

When n is extremely large, the predominant methodology to solve the above problem advocates the use of stochastic gradient descent (SGD) methods [1–4]. In the t th iteration, the classical SGD method updates iterates as follows:

$$\omega_{t+1} = \omega_t - \eta_t \nabla f_{i_t}(\omega_t), \tag{3}$$

where $\eta_t > 0$ refer to the step size, the index i_t can be chosen uniformly at random from $\{1, 2, \ldots, n\}$, and $\nabla f_{i_t}(\omega_t)$ denotes the sample gradient. The expectation of the stochastic gradient estimator $\nabla f_{i_t}(\omega_t)$ is usually regarded as unbiased estimation for $\nabla F(\omega_t)$,*i.e.*, $\mathbb{E}\left[\nabla f_{i_t}(\omega_t)\right] = \nabla F(\omega_t)$. Unfortunately, the randomness may introduce variance in practice, which is caused by the fact that stochastic gradient $\nabla f_{i_t}(\omega_t)$ equals the full gradient $\nabla F(\omega_t)$ in expectation, but each $\nabla f_{i_t}(\omega_t)$ is different. In fact, the performance of SGD method is often too sensitive to the variance in the sample gradients $\nabla f_{i_t}(\omega_t)$, even if the objective function is strongly convex and smooth, it only converges sub-linearly [5].

In recent years, a surge of methods to improve the performance of SGD have been developed. The stochastic average gradient(SAG) method [6] and the SAGA method [7] computed a stochastic gradient as an average of stochastic gradients evaluated at previous iterates and then store previous stochastic gradients at the expense of memory. However, both SAG and SAGA are expensive when n is extremely large. The stochastic variance reduced gradient (SVRG) method [8] selected a stochastic gradient with low variance as an unbiased estimate of the full gradient. Nguyen et al. [10, 11] presented the stochastic recursive gradient algorithm(SARAH) with one-loop or multiple-loop, which has an additional practical advantage of being able to use an adaptive inner loop size, and their convergence rates matches that of SVRG in the strongly convex case. In a recent paper [12], Nguyen et al. made further improvement for SARAH by replacing the exact gradient computation with a stochastic gradient based on a sufficiently large mini-batch, which can be regarded as an inexact version of SARAH (iSARAH), the iSARAH performs variance reduction by computing a sufficiently accurate gradient estimate in the outer loop and performs the stochastic gradient updates in the inner loop. It is shown that the multiple-loop iSARAH achieves the best convergence rate under an additional assumption among the compared stochastic methods.

What they have in common is that in the process of solving practical problems, they usually use a fixed step size. A large number of numerical experiments show that the performance of SGD type methods are greatly affected by the step size selection. One common strategy is using a constant step size, but it usually needs to be hand-picked, which is time consuming in practice. Another common approach is to adopt diminishing step sizes that requires to satisfy

$$\sum_{t=1}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=1}^{\infty} \eta_t^2 < \infty. \tag{4}$$

However, it often leads to SGD with a severe slow convergence rate [13]. There are some related works about the choice of step size in SGD type methods. AdaGrad [14] and Adam [15] adaptively selected the step size for every component based on the sum of the squares of the past gradients. Recently, due to the BB approach [16–18] can adaptively update the step size and has good numerical performance, many researchers try to incorporate it into SGD type algorithms. For instance, Sopya et al. [19] presented several variants of the BB method for SGD to train the linear SVM, Tan et al. [20] used the BB method to calculate the step size for SGD and SVRG, and put forward two approaches, named as SGD-BB and SVRG-BB, respectively, Li et al. [21] used the BB method to calculate the step size for SARAH, Liu et al. [22] and Yang et al. [23, 24] also incorporated the BB method to compute step size for the variants of SGD type algorithms, all of them have good numerical performance. In fact, it can not avoid the denominator being close to zero or even negative when using the BB or RBB formulas to compute step size. To overcome this shortcoming, Ma et al. [25] incorporated the stabilized Barzilai-Borwein(SBB) step size into the SVRG method and generated a new method called SVRG-SBB for the ordinal embedding problem. Motivated by

above articles, we consider using the stabilized BB method(SBB) to automatically compute step size for iSARAH algorithm and then propose a new algorithm called iSARAH-SBB.

The primary contributions of our work in this paper can be summarized as follows:

1) We incorporate stabilized BB method [25] into iSARAH [10], which leads to a modified stochastic recursive gradient methods called iSARAH-SBB.
2) We establish the convergence and complexity analysis of iSARAH-SBB method for strongly convex optimization problem.
3) We conduct extensive experiments for iSARAH-SBB method on solving logistic regression problem. Numerical experimental results on standard data sets show the effectiveness of our proposed method.

This paper is organized as follows. Section 2 briefly introduces the background and motivation; Section 3 analyzes the convergence and the complexity of our proposed iSARAH-SBB; Section 4 demonstrates the experimental results; Section 5 concludes this paper.

2. The Algorithms

In this section, we first introduce the stabilized BB method and the iSARAH method in Section 2.1 and Section 2.2, respectively, and then put forward our modified method iSARAH-SBB in Section 2.3. The BB step size can be described as follows.

2.1. Barzilai-Borwein step zize

The well-known Barzilai-Borwein(BB) step size, originally proposed by Barzilai and Borwein in [16] , [17], which tries to fit the objective by a quadratic model in each iteration and find the optimal step size. It is widely used to solve the generic unconstrained optimization problem:

$$\min_{\omega\in\mathbb{R}^d} f(\omega), \tag{5}$$

when minimizing a first-order continuously differentiable function $f(\omega)$, the standard BB method updates the iterates through

$$\omega_{k+1} = \omega_k - \eta_k^{-1}\nabla f(\omega_k), \tag{6}$$

where $\nabla f(\omega_k)$ denotes the gradient of $f(\omega)$ at ω_k and η_k is introduced such that ηI is an approximation to the Hessian matrix of $f(\omega)$ at ω_k, so it usually follows some properties of quasi-Newton method and by solving the following problem:

$$\min_{\eta} \|\eta^{-1}s_k - y_k\|_2 \quad or \quad \min_{\eta} \|s_k - \eta y_k\|_2, \tag{7}$$

where $s_k = \omega_k - \omega_{k-1}$ and $y_k = \nabla f(\omega_k) - \nabla f(\omega_{k-1})$, it can yield that

$$\eta_k^{BB1} = \frac{s_k^T s_k}{s_k^T y_k} \quad or \quad \eta_k^{BB2} = \frac{s_k^T y_k}{y_k^T y_k}. \tag{8}$$

In fact, when $s_k^T y_k > 0$, it is easy to obtain that $\eta_k^{BB1} \geq \eta_k^{BB2}$. [18] has proved that η_k^{BB1} is superior to η_k^{BB2}, which means η_k^{BB1} is a more aggressive step size to decrease the objective function. As the aforementioned description in Section 1, Tan et al. and Yang et al. directly introduced the BB method into SGD and its variant for solving problem (2). Ma et al.(2018) incorporated the SBB step size into the SVRG method for solving the ordinal embedding problem, where the updating scheme of the step size can be formulated as:

$$\eta_k = \frac{s_k^T s_k}{|s_k^T y_k| + \triangle}, \tag{9}$$

where $\triangle = \sigma s_k^T s_k \, (\sigma > 0)$. In fact, when $\sigma = 0$, the SBB method may always exist instability. To avoid the denominator of the SBB step size tending to zero, we choose $\sigma > 0$ in numerical experiments. A large number of numerical experiments have verified the advantages of the SBB step size, inspired by these related works, we propose to use the SBB step size to automatically compute the step size for iSARAH.

2.2. The iSARAH Algorithm

As a variant of SARAH methods, iSARAH consists of the outer loop and the inner loop. The outer loop performs variance reduction by computing sufficiently accurate gradient estimate and the inner loop performs recursive stochastic gradient updates. Specifically, when given an iterate ω_0 at the beginning of each outer loop, iSARAH replaces the exact gradient computation by a gradient estimate based on a sample set of size b. For general empirical risk minimization problem (2), v_0 is computed as

$$v_0 = \frac{1}{b} \sum_{\xi \in S} \nabla f_\xi(w_0), \tag{10}$$

where ω_0 is the initial iteration point , S is a subset of $\{1, 2, \ldots, n\}$ with size b.

The key step of the algorithm is a recursive update of the stochastic gradient estimate that we call SARAH update

$$v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}, \tag{11}$$

followed by the iterate update

$$w_{t+1} = w_t - \eta v_t. \tag{12}$$

Let $\mathcal{F}_t = \sigma(w_0, w_1, \ldots, w_t)$ be the σ-algebra generated by $w_0, w_1, \ldots, w_t$. We note that i_t is independent of $\mathcal{F}_t$, and v_t is a *biased estimator* of the gradient $\nabla F(w_t)$, thus

$$\mathbb{E}[v_t|\mathcal{F}_t] = \nabla F(w_t) - \nabla F(w_{t-1}) + v_{t-1}, \tag{13}$$

and since $\mathbb{E}[\nabla f_\xi(w_0)|\omega_0] = \nabla F(w_0)$, we have

$$\mathbb{E}[v_0|\omega_0] = \frac{1}{b} \sum_{\xi \in S} \nabla F(w_0) = \nabla F(w_0). \tag{14}$$

.

The iSARAH algorithm in [12] can be described as follows:

Algorithm 1 (iSARAH)

Parameters: initial point $\tilde{\omega}_0$, fixed step size $\eta > 0$, the inner loop size m, the sample set size b

for $k = \{1, 2, \ldots\}$ **do**
  $\omega_0 = \tilde{\omega}_{k-1}$
  Randomly choose a subset $S \subset \{1, \ldots, n\}$ of size b
  Compute $v_0 = \frac{1}{b} \sum_{\xi \in S} \nabla f_\xi(w_0)$.
  $\omega_1 = \omega_0 - \eta v_0$
  for $t = 1, \ldots \, m - 1$ **do**
    Randomly pick $i_t \in \{1, 2, \ldots, n\}$
    Update the stochastic recursive gradient:
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$.
    Update the iterate:
    $\omega_{t+1} = \omega_t - \eta v_t$
  end for
  Set $\tilde{\omega}_k = \omega_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
end for

2.3. The iSARAH-SBB Algorithm

In this subsection, we propose the iSARAH-SBB algorithm, which uses the SBB method to automatically compute the step size η_k in iSARAH. In addition, we set $\tilde{\omega}_k = \omega_m$ rather than use $\tilde{\omega} = \omega_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$. Now we are ready to describe our iSARAH-SBB method in the following Algorithm 2.

Algorithm 2 (iSARAH-SBB)

Parameters: $\sigma > 0$, initial point $\tilde{\omega}_0$, initial step size $\eta_1 > 0$, the inner loop size m, the sample set size b

for $k = \{1, 2, \ldots\}$ **do**
  $\omega_0 = \tilde{\omega}_{k-1}$
  Randomly choose a subset $S \subset \{1, \ldots, n\}$ of size b
  Compute $v_0^k = \frac{1}{b}\sum_{\xi \in S} \nabla f_\xi(w_0)$.
  if $k > 1$ **then**
    $\eta_k = \frac{1}{m} \frac{\|\tilde{\omega}_k - \tilde{\omega}_{k-1}\|_2^2}{(\tilde{\omega}_k - \tilde{\omega}_{k-1})^T \left(v_0^k - v_0^{k-1}\right) + \sigma \|\tilde{\omega}_k - \tilde{\omega}_{k-1}\|_2^2}$
  end if
  $\omega_1 = \omega_0 - \eta_k v_0^k$
  for $t = 1$ **to** $m - 1$ **do**
    Randomly pick $i_t \in \{1, 2, \ldots, n\}$
    Update the stochastic recursive gradient:
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$.
    Update the iterate:
    $\omega_{t+1} = \omega_t - \eta_k v_t^k$
  end for
  $\tilde{\omega}_k = \omega_m$
end for

Remark 1. It can be seen from Algorithm 2 that if we set $\eta_k = \eta$ instead of using the BB step size, the iSARAH-SBB reduces to the original iSARAH method.

Remark 2. In the step size update rule, the reason for dividing η_k by m is that in order to update ω_t in the inner loop, m gradient estimates need to be added to ω_0 in turn.

3. Convergence analysis

In this section,we prove the convergence of iSARAH-SBB for solving problem (2) with the strongly convex objective function $F(\omega)$, our analysis is conducted based on the following key assumptions.

3.1. The Key Assumptions

Assumption 1 (L-smooth)
$f_\xi(w)$ is L-smooth for every realization of ξ, i.e., there exists a constant $L > 0$ such that

$$\|\nabla f_\xi(w) - \nabla f_\xi(w')\| \leq L\|w - w'\|, \ \forall\, w, w' \in \mathbb{R}^d. \tag{15}$$

Assumption 2 (μ-strongly convex)
The function
$F(\omega) : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex, i.e., there exists a constant $\mu > 0$ such that $\forall w, w' \in \mathbb{R}^d$,

$$\left(\nabla F(w) - \nabla F(w')\right)^T (w - w') \geq \mu\|w - w'\|^2, \tag{16}$$

or equivalently

$$F(w) \geq F(w') + \nabla F(w')^\top (w - w') + \tfrac{\mu}{2}\|w - w'\|^2. \tag{17}$$

Under Assumption 2, then there exists an unique optimal solution of problem (2), which is denoted by w_*. Thus the strong convexity of F implies that

$$2\mu[F(w) - F(w_*)] \leq \|\nabla F(w)\|^2, \ \forall w \in \mathbb{R}^d. \tag{18}$$

Assumption 3 (Convex)
$f_\xi(w)$ is convex for every realization of ξ, i.e., $\forall w, w' \in \mathbb{R}^d$,

$$f_\xi(w) \geq f_\xi(w') + \nabla f_\xi(w')^\top (w - w'). \tag{19}$$

Assumption 4
There exists some $\tau_* > 0$ such that

$$\mathbb{E}[\|\nabla f_\xi(w_*)\|^2] \leq \sigma_*^2, \tag{20}$$

where w_* is any optimal solution of $F(w)$; and ξ is some random variable.

3.2. *Existing Results*

In this subsection, we review some well-known results from the past literature that can support our theoretical analysis.

Lemma 1 (Lemma 3.6 in [12])
Suppose that Assumption 1 holds. Consider iSARAH (Algorithm 1). Then

$$\sum_{t=0}^{m} \mathbb{E}[\|\nabla F(w_t)\|^2] \leq \frac{2}{\eta}\mathbb{E}[F(w_0) - F(w_*)] + \sum_{t=0}^{m} \mathbb{E}[\|\nabla F(w_t) - v_t\|^2] - (1 - L\eta)\sum_{t=0}^{m} \mathbb{E}[\|v_t\|^2],$$

where $w_* = \arg\min_w F(w)$.

Lemma 2 (Lemma 3.8 in [12])
Suppose that Assumptions 1 and 3 hold. Consider v_t defined as (11) in SARAH (Algorithm 1) with $\eta < 2/L$. Then for any $t \geq 1$,

$$\mathbb{E}[\|\nabla F(w_t) - v_t\|^2] \leq \frac{\eta L}{2 - \eta L}\left[\mathbb{E}[\|v_0\|^2] - \mathbb{E}[\|v_t\|^2]\right] + \mathbb{E}[\|\nabla F(w_0) - v_0\|^2].$$

3.3. *Strongly Convex Case Results*

Lemma 3
Suppose that Assumptions 1 and 3 hold. Consider iSARAH (Algorithm 1) with $\eta \leq 1/L$. Then

$$\begin{aligned}\mathbb{E}[\|\nabla F(w_m)\|^2] \leq{}& \frac{2}{\eta(m+1)}\mathbb{E}[F(w_0) - F(w_*)] + \frac{\eta L}{2 - \eta L}\mathbb{E}[\|\nabla F(w_0)\|^2] \\ &+ \frac{2}{2 - \eta L}\left(\frac{4L\mathbb{E}[F(w_0) - F(w_*)]}{b}\right) \\ &+ \frac{2}{2 - \eta L}\left(\frac{2\mathbb{E}\left[\|\nabla f(w_*;\xi)\|^2\right] - \mathbb{E}[\|\nabla F(w_0)\|^2]}{b}\right),\end{aligned}$$

where w_* is any optimal solution of $F(w)$ and ξ is the random variable.

Proof
By Lemma 2, we have

$$\sum_{t=0}^{m}\mathbb{E}[\|\nabla F(w_t)-v_t\|^2]\leq\frac{m\eta L}{2-\eta L}\mathbb{E}[\|v_0\|^2]+(m+1)\mathbb{E}[\|\nabla F(w_0)-v_0\|^2]. \tag{21}$$

Hence, by Lemma 1 with $\eta\leq 1/L$,

$$\begin{aligned}\sum_{t=0}^{m}\mathbb{E}[\|\nabla F(w_t)\|^2]\leq&\frac{2}{\eta}\mathbb{E}[F(w_0)-F(w_*)]+\sum_{t=0}^{m}\mathbb{E}[\|\nabla F(w_t)-v_t\|^2]\\ \leq&\frac{2}{\eta}\mathbb{E}[F(w_0)-F(w_*)]+\frac{m\eta L}{2-\eta L}\mathbb{E}[\|v_0\|^2]\\ &+(m+1)\mathbb{E}[\|\nabla F(w_0)-v_0\|^2].\end{aligned}$$

Since $\tilde{w_k}=w_m$, the following holds,

$$\begin{aligned}&\mathbb{E}[\|\nabla F(w_m)\|^2]=\frac{1}{m+1}\sum_{t=0}^{m}\mathbb{E}[\|\nabla F(w_t)\|^2]\\ \leq&\frac{2}{\eta(m+1)}\mathbb{E}[F(w_0)-F(w_*)]+\frac{\eta L}{2-\eta L}\mathbb{E}[\|v_0\|^2]+\mathbb{E}[\|\nabla F(w_0)-v_0\|^2]\\ \leq&\frac{2}{\eta(m+1)}\mathbb{E}[F(w_0)-F(w_*)]+\frac{\eta L}{2-\eta L}\mathbb{E}[\|\nabla F(w_0)\|^2]\\ &+\frac{2}{2-\eta L}\left(\frac{4L\mathbb{E}[F(w_0)-F(w_*)]}{b}\right)\\ &+\frac{2}{2-\eta L}\left(\frac{2\mathbb{E}\left[\|\nabla f(w_*;\xi)\|^2\right]-\mathbb{E}[\|\nabla F(w_0)\|^2]}{b}\right).\end{aligned}$$

□

Lemma 4 (The bound of SBB step size)
Suppose that Assumption 1 and 2 hold. We have that

$$\frac{1}{m(L+\sigma)}\leq\eta_k\leq\frac{1}{m\sigma}. \tag{22}$$

Proof

$$\begin{aligned}\eta_k=&\frac{1}{m}\frac{\|\tilde{\omega}_k-\tilde{\omega}_{k-1}\|_2^2}{(\tilde{\omega}_k-\tilde{\omega}_{k-1})^T\left(v_0^k-v_0^{k-1}\right)+\sigma\|\tilde{\omega}_k-\tilde{\omega}_{k-1}\|_2^2}\\ \geq&\frac{1}{m}\frac{\|\widetilde{\omega}_k-\widetilde{\omega}_{k-1}\|_2^2}{\|\widetilde{\omega}_k-\widetilde{\omega}_{k-1}\|\left\|v_0^k-v_0^{k-1}\right\|+\sigma\|\widetilde{\omega}_k-\widetilde{\omega}_{k-1}\|_2^2}\\ \geq&\frac{1}{m}\frac{\|\tilde{\omega}_k-\tilde{\omega}_{k-1}\|_2^2}{(L+\sigma)\|\tilde{\omega}_k-\tilde{\omega}_{k-1}\|_2^2}=\frac{1}{m(L+\sigma)}\end{aligned} \tag{23}$$

In addition, we can easily obtain that

$$\eta_k\leq\frac{1}{m}\frac{\|\tilde{\omega}_k-\tilde{\omega}_{k-1}\|_2^2}{\sigma\|\tilde{\omega}_k-\tilde{\omega}_{k-1}\|_2^2}=\frac{1}{m\sigma}, \tag{24}$$

Thus

$$\frac{1}{m(L+\sigma)} \leq \eta_k \leq \frac{1}{m\sigma}. \tag{25}$$

□

We now turn to the discussion on the convergence of iSARAH under the strong convexity assumption on F.

Theorem 1
Suppose that Assumptions 1, 2, 3,and 4 hold.Consider iSARAH-SBB (Algorithm 2) with the choice of σ,m,and b such that

$$\alpha = \frac{\sigma m}{\mu(m+1)} + \frac{1}{2m\sigma - L}\left\{L + \frac{m\sigma}{b}(4\kappa - 2)\right\} < 1.$$

(where $\kappa = L/\mu$.) Then

$$\mathbb{E}[\|\nabla F(\tilde{w}_k)\|^2] - \Delta \leq \alpha^k(\|\nabla F(\tilde{w}_0)\|^2 - \Delta), \tag{26}$$

where

$$\Delta = \frac{\delta}{1-\alpha} \text{ and } \delta = \frac{4}{b(2 - \frac{L}{m\sigma})}\tau_*^2. \tag{27}$$

Proof
By Lemma 3, with $w_m = \tilde{w}_k$ and $w_0 = \tilde{w}_{k-1}$, we have

$$\begin{aligned}
\mathbb{E}[\|\nabla F(\tilde{w}_k)\|^2] &\leq \frac{2}{\eta(m+1)}\mathbb{E}[F(\tilde{w}_{k-1}) - F(w_*)] \\
&+ \frac{\eta L}{2-\eta L}\mathbb{E}[\|\nabla F(\tilde{w}_{k-1})\|^2] + \frac{2}{2-\eta L}\left(\frac{4L\mathbb{E}[F(\tilde{w}_{k-1}) - F(w_*)]}{b}\right) \\
&+ \frac{2}{2-\eta L}\left(\frac{2\mathbb{E}\left[\|\nabla f(w_*;\xi)\|^2\right] - \mathbb{E}[\|\nabla F(\tilde{w}_{k-1})\|^2]}{b}\right) \\
&\leq \left(\frac{1}{\mu\eta(m+1)} + \frac{\eta L}{2-\eta L} + \frac{4\kappa - 2}{b(2-\eta L)}\right)\mathbb{E}[\|\nabla F(\tilde{w}_{k-1})\|^2] \\
&\quad + \frac{4}{b(2-\eta L)}\mathbb{E}\left[\|\nabla f(w_*;\xi)\|^2\right] \\
&\leq \left(\frac{1}{\mu\eta(m+1)} + \frac{\eta L}{2-\eta L} + \frac{4\kappa - 2}{b(2-\eta L)}\right)\mathbb{E}[\|\nabla F(\tilde{w}_{k-1})\|^2] \\
&\quad + \frac{4}{b(2-\eta L)}\tau_*^2 \\
&\leq \left(\frac{\sigma m}{\mu(m+1)} + \frac{L}{2m\sigma - L} + \frac{4\kappa - 2}{b(2-\frac{L}{m\sigma})}\right)\mathbb{E}[\|\nabla F(\tilde{w}_{k-1})\|^2] \\
&\quad + \frac{4}{b(2-\frac{L}{m\sigma})}\tau_*^2 \\
&= \alpha\mathbb{E}[\|\nabla F(\tilde{w}_{k-1})\|^2] + \delta \\
&\leq \alpha^k\|\nabla F(\tilde{w}_0)\|^2 + \alpha^{k-1}\delta + \cdots + \alpha\delta + \delta \\
&\leq \alpha^k\|\nabla F(\tilde{w}_0)\|^2 + \delta\frac{1-\alpha^k}{1-\alpha} \\
&= \alpha^k\|\nabla F(\tilde{w}_0)\|^2 + \Delta(1-\alpha^k)
\end{aligned}$$

$$= \alpha^k(\|\nabla F(\tilde{w}_0)\|^2 - \Delta) + \Delta.$$

By adding $-\Delta$ to both sides, we achieve the desired result. □

Corollary 1
Let $m = 20\kappa - 1$, $\sigma = \frac{5L}{40\kappa-2}$ and $b = \max\left\{20\kappa - 10, \frac{20\tau_*^2}{\epsilon}\right\}$ in Theorem1. Then, the total work complexity to achieve $\mathbb{E}[\|\nabla F(\tilde{w}_k)\|^2] \leq \epsilon$ is $\mathcal{O}\left(\max\left\{\frac{\tau_*^2}{\epsilon}, \kappa\right\}\log\left(\frac{1}{\epsilon}\right)\right)$.

Proof
With $m = 20\kappa - 1$, $\sigma = \frac{5L}{40\kappa-2}$ and $b = \max\left\{20\kappa - 10, \frac{20\tau_*^2}{\epsilon}\right\}$,

$$\begin{aligned}\mathbb{E}[\|\nabla F(\tilde{w}_k)\|^2] &\leq \left(\frac{1}{8} + \frac{1}{4} + \frac{1}{8}\right)\mathbb{E}[\|\nabla F(\tilde{w}_{k-1})\|^2] + \frac{\epsilon}{8} \\ &\leq \frac{1}{2}\mathbb{E}[\|\nabla F(\tilde{w}_{k-1})\|^2] + \frac{\epsilon}{8} \\ &\leq \left(\frac{1}{2}\right)^k \|\nabla F(\tilde{w}_0)\|^2 + \frac{\epsilon}{4}.\end{aligned}$$

Since $\mathbb{E}[\|\nabla f(w_*;\xi)\|^2$ is finite, to guarantee that $\mathbb{E}[\|\nabla F(\tilde{w}_k)\|^2] \leq \epsilon$, it is sufficient to make $\left(\frac{1}{2}\right)^k \|\nabla F(\tilde{w}_0)\|^2 = \frac{3\epsilon}{4}$ or equivalently $k = \tau * log(\frac{\|\nabla F(\tilde{w}_0)\|^2}{\frac{3\epsilon}{4}})$, where τ is a constant. This implies that the total complexity to achieve an ϵ-accuracy solution is

$$(b+m)k = \mathcal{O}\left(\max\left\{\frac{\sigma_*^2}{\epsilon}, \kappa + \kappa\right\}\log\left(\frac{1}{\epsilon}\right)\right) = \mathcal{O}\left(\max\left\{\frac{\sigma_*^2}{\epsilon}, \kappa\right\}\log\left(\frac{1}{\epsilon}\right)\right).$$

□

4. Numerical experiments

In this section, we present our numerical experiments. All the tests have been performed on an Intel Core i7 processor with 10GB RAM under the Python computing environment. We study the binary classification problem with f_i being l_2-regularized logistic regression on data sets heart, splice, a9a and w8a as Table 1 which can be downloaded from the LIBSVM website (www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/).

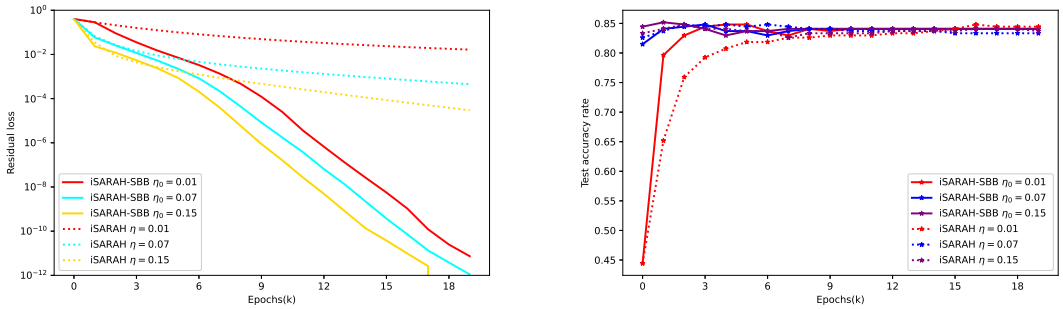
The l_2-regularized logistic regression problem can be described as follows:

$$\min_{w\in\mathbb{R}^d} \quad F(w) := \frac{1}{n}\sum_{i=1}^{n}\log(1 + \exp(-y_i x_i^T w)) + \frac{\lambda}{2}\|w\|^2, \tag{28}$$

where $\{(x_i, y_i)\}_{i=1}^n \subset R^d \times \{+1, -1\}^n$ is a collection of training examples.

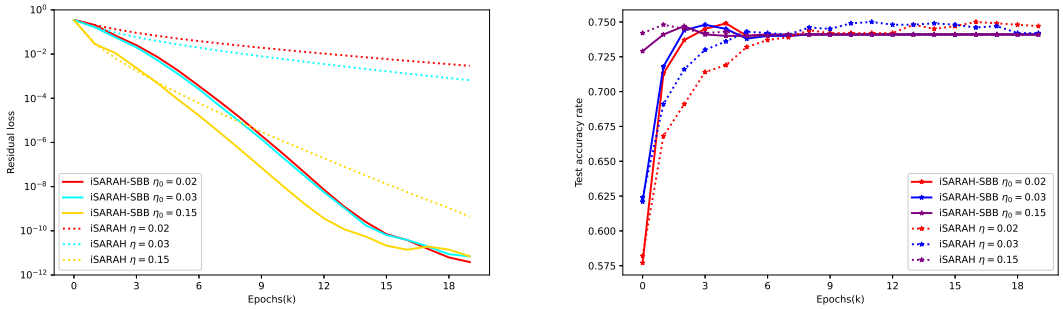
Table 1. Data information of experiments.

Dataset	**n**	**d**	λ
heart	270	13	10^{-4}
splice	1000	60	10^{-4}
a9a	22,696	123	10^{-4}
w8a	49,749	300	10^{-4}



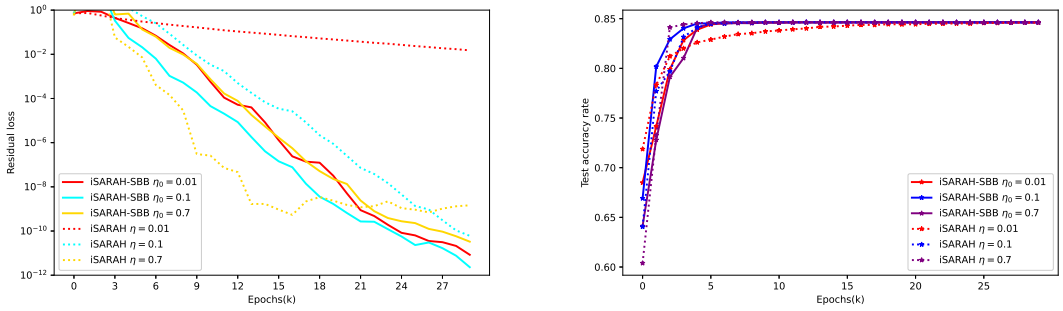
(a) Residual loss on heart (b) Test accuracy rate on heart

Figure 1. Comparison of iSARAH-SBB and iSARAH with fixed step sizes on residual loss and test accuracy rate using heart.



(a) Residual loss on splice (b) Test accuracy rate on splice

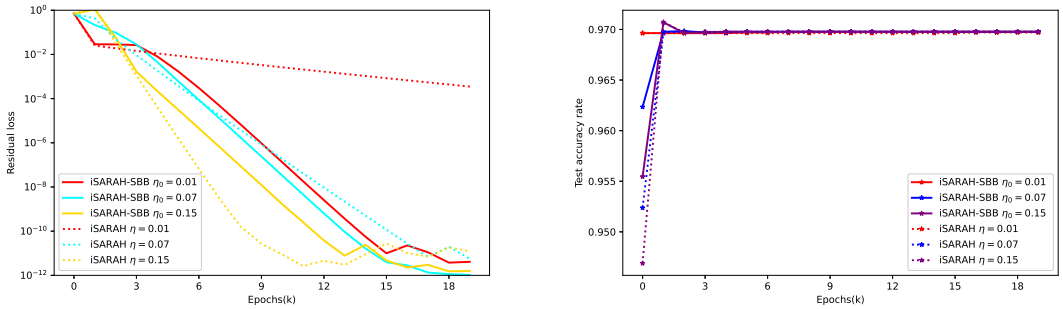
Figure 2. Comparison of iSARAH-SBB and iSARAH with fixed step sizes on residual loss and test accuracy rate using splice.



(a) Residual loss on a9a (b) Test accuracy rate a9a

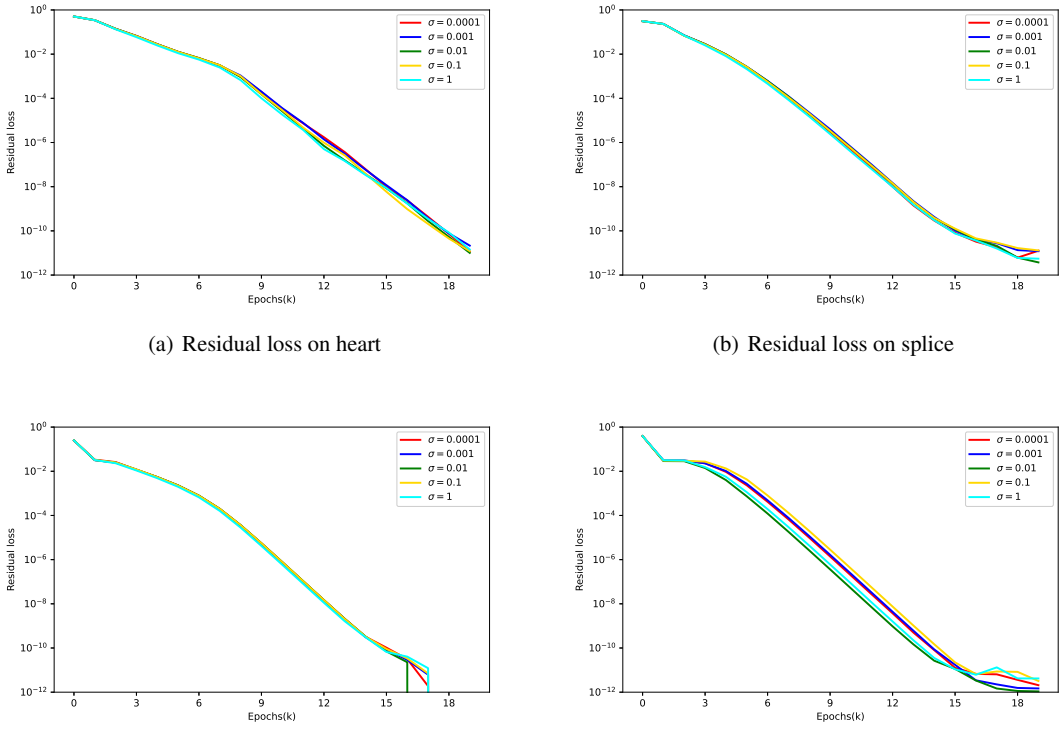
Figure 3. Comparison of iSARAH-SBB and iSARAH with fixed step sizes on residual loss and test accuracy rate using a9a.

Our experiments include three parts. In the first part, we compare the performance of iSARAH-SBB and iSARAH for solving problem (28) in Fig.1 to Fig.4. We compare the residual loss and test accuracy rate of them,



(a) Residual loss on w8a (b) Test accuracy rate on w8a

Figure 4. Comparison of iSARAH-SBB and iSARAH with fixed step sizes on residual loss and test accuracy rate using w8a.



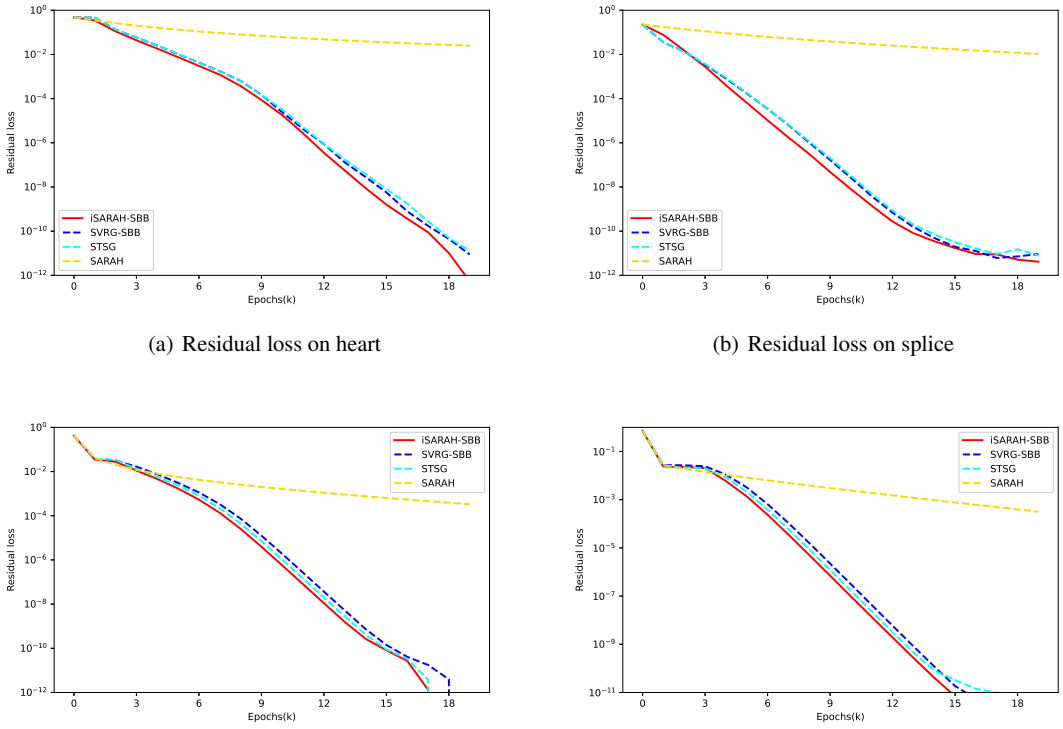
(a) Residual loss on heart (b) Residual loss on splice

(c) Residual loss on a9a (d) Residual loss on w8a

Figure 5. The performance of iSARAH-SBB with different σ on different datasets.

respectively. In the second part, we analyze the influence of the parameter σ on the iSARAH-SBB in Fig.5. Finally, we compare the residual loss of SVRG-SBB [25], STSG [26] and SARAH [11]with the iSARAH-SBB method in Fig.6 and we also compare the test accuracy rate of them in Fig.7.

Fig.2 to Fig.4 show the comparison results on residual loss and test accuracy of iSARAH-SBB and iSARAH on datasets heart, splice, a9a and w8a. We use three different fixed step sizes for iSARAH and three different initial step sizes for iSARAH-SBB. The y-axis represents the residual loss in the left subfigures and the test accuracy rate



(a) Residual loss on heart

(b) Residual loss on splice

(c) Residual loss on a9a

(d) Residual loss on w8a

Figure 6. Comparison of iSARAH-SBB, SVRG-SBB, STSG, SARAH on residual loss on different datasets.

in the right subfigures. In all the subfigures, the dashed lines correspond to iSARAH with fixed step sizes η, and the solid lines correspond to iSARAH-SBB with different initial step sizes η_0. Fig.5 show the influence of σ on the performance of iSARAH-SBB.

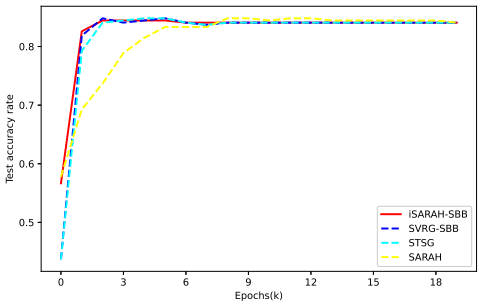
From the left subfigures in Fig.2 to Fig.4, it can be seen that iSARAH-SBB always outperforms iSARAH with the three choices of step sizes. In addition, from the right subfigures in Fig.2 to Fig.4, it can be seen that iSARAH-SBB can improve the test accuracy rate when choose different datasets.

Fig.5 shows that different σ have little effect on the performance of iSARAH-SBB method, which indicate that iSARAH-SBB method is not sensitive to the parameter σ. To present this case,we set $\sigma = 0.0001, 0.001, 0.01, 0.1$ and 1 in heart, splice, a9a and w8a.

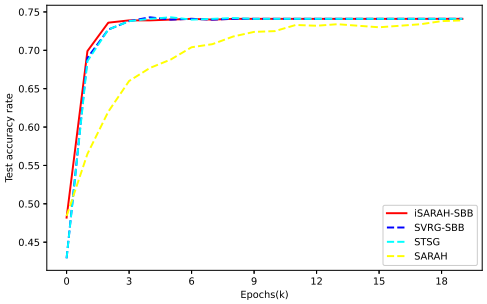
From the Fig.6 to Fig.7, it can be seen that iSARAH-SBB always performs better than SVRG-SBB, STSG and SARAH, and iSARAH can achieve the same level of the test accuracy rate as SVRG-SBB, STSG and SARAH.

5. Conclusion

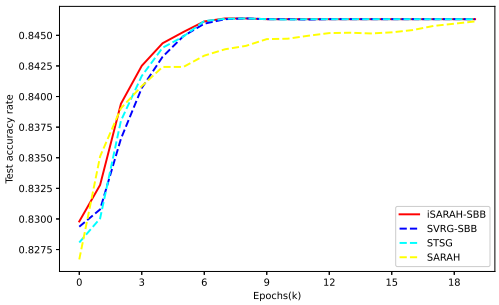
In this paper, we have proposed a modified algorithm iSARAH-SBB, which used the SBB method to dynamically solve the step size and can take better advantages of both iSARAH and SBB methods. Under the usual assumptions, we establish the convergence and complexity analysis of our proposed iSARAH-SBB algorithm. Compared with the existing algorithms, the new algorithm is simple and with good theoretical properties. We also discussed the effect of different σ on the performances of iSARAH-SBB. Numerical experiment results on standard datasets



(a) Test accuracy rate on heart



(b) Test accuracy rate on splice



(c) Test accuracy rate on a9a

Figure 7. Comparison of iSARAH-SBB, SVRG-SBB, STSG, SARAH on test accuracy rate on different datasets.

demonstrate that the new algorithm is robust to the selection of the initial step size, and it is effective and more competitive.

Acknowledgement

The research described in this paper was supported by Basic Research Program of Shanxi Province (Free Exploration) Project under Grants (No.202103021224303,20210302124688), Hunan Provincial Innovation Foundation For Postgraduate(CX20230617),and Saint-Petersburg State University, project ID: 94062114.

The authors are indebted to the editors and anonymous referees for their time and work.

REFERENCES

1. H. Robbins, S. Monro, *A stochastic approximation method* , Ann Math Statist, vol. 3, no. 22, pp. 400–407, 1951.
2. F. Ding, H.Z. Yang, F. Liu, *Performance analysis of stochastic gradient algorithms under weak conditions* , Sci China Ser.F-Inf.Sci, vol. 9, no. 51, pp. 1269–1280, 2008.
3. L. Bottou, F.E. Curtis, J. Nocedal, *Optimization methods for large-scale machine learning* , SIAM Rev ,vol. 2, no. 60, pp. 223–311,2016.
4. N. Le Roux, M. Schmidt, F. Bach, *A stochastic gradient method with an exponential convergence rate for finite training sets* , in: Proceedings of 26th Conference on Neural Information Processing Systems, pp. 2663–2671,2012.
5. E. Moulines, F.R.Bach, *Non-asymptotic analysis of stochastic approximation algorithms for machine learning* , in: Advances in Neural Information Processing Systems, pp. 451–459,2011.

6. M. Schmit, N. Le Roux, F, *Bach, Minimizing finite sums with the stochastic average gradient* , Math.Program, vol. 1, no. 162, pp. 83–112,2017.
7. A. Defazio, F. Bach, S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, in:Proceedings of Neural Information Processing Systems. Montreal:Curran Associates, pp.1646-1654, 2014.
8. R. Johnson, T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in NIPS, pp.315-323, 2013.
9. J. Konečnýj , P. Richtárik, *Semi-stochastic gradient descent methods*, Front Appl Math Stat, pp. 3–9, 2017.
10. L.M. Nguyen, J. Liu, K. Scheinberg, M. Takáč, *Stochastic recursive gradient algorithm for nonconvex optimization*, 2017,arXiv:1705.07261.
11. L. Nguyen, J. Liu, K. Scheinberg, M. Takáč, *SARAH: A novel method for machine learning problems using stochastic recursive gradient*, ICML, pp. 2613–2621, 2017.
12. L.M. Nguyen, K. Scheinberg, M. Takáč, *Inexact SARAH algorithm for stochastic optimization*, Optim Method Softw, 36, 237-258(2020)
13. L. Bottou, *Online learning and stochastic approximations, Online Learn* , Neural Netw, no. 17, pp. 9–42, 1998.
14. J.C. Duchi, E. Hazan, Y.J, *Singer, Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research. no. 12, pp.2121–2159, 2011.
15. D.P. Kingma, J. Ba, *Adam:a method for stochastic optimization*, in: International Conference on Learning Representations, pp.1-13, 2015.
16. J. Barzilai, J.M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal, vol. 1, no. 8, pp. 141–148, 1988.
17. Y.H. Dai, L.Z. Liao, *R-linear convergence of the Barzilai and Borwein gradient method*, IMA Journal of Numerical Analysis, 22(1),1-10(2002)
18. R. Fletcher, *On the Barzilai-Borwein method, in Optimization and Control with Applications* , L.Q. Qi, K.L. Teo, X.Q. Yang, and D.W. Hearn, eds., Applied Optimization, Springer, Amsterdam, vol. 96, pp. 235-256, 2005.
19. Sopya, P.K. Drozda, *Stochastic gradient descent with barzilai-borwein update step for svm* , Information Sciences,vol.316,pp.218–233, 2015.
20. C. Tan, S. Ma, Y.H. Dai, Y. Qian, *Barzilai-Borwein step size for stochastic gradient descent* , in:Neural Information Processing Systems, pp.685–693, 2016.
21. B.C. Li, G.B, *Giannakis, Adaptive Step Sizes in Variance Reduction via Regularization* , 2019, Available at arXiv:1910.06532.
22. Y. Liu, X. Wang, T.D. Guo, *A linearly convergent stochastic recursivegradient method for convex optimization* , Optim. Lett. A,2020. doi:10.1007/s11590-020-01550-x.
23. Z. Yang, C. Wang, Z.M. Zhang, J. Li, *Random Barzilai-Borwein step size for mini-batch algorithms* , Engineering Applications of Artificial Intelligence. no. 72, pp.124–135, 2018.
24. Z. Yang, Z.P. Chen, C. Wang, *Accelerating Mini-batch SARAH by Step Size Rules* , Information Sciences. vol. 558, no. 1, pp.157–173, 2021.
25. K. Ma, J. Zeng, J.Xiong, Q. Xu, X. Cao, W. Liu, Y. Yao, *Stochastic Non-convex Ordinal Embedding with Stabilized Barzilai-Borwein step size* , in: AAAI Conference on Artificial Intelligence, 2018.
26. G.M. Shao, W. Xue, G.H. YU, *Improved SVRG for finite sum structure optimization with application to binary classification* , J. Ind. Manag. Optim. vol. 13, no. 5, pp. 1–14, 2019.