A Mini-batch Stochastic Recursive Gradient Method with Barzilai-Borwein Step-Size for Machine Learning

Yi-Ming Yang
Xiangtan University
Fu-Sheng Wang (fswang2005@163.com)
Taiyuan Normal University
Zheng Peng
Xiangtan University
Xiao-Jun Zhou
Hebei University of Technology

Research Article

Keywords: Machine learning, Mini batches, SARAH algorithm, BB step-size

Posted Date: July 6th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3129748/v1

Additional Declarations: No competing interests reported.

A Mini-batch Stochastic Recursive Gradient Method with Barzilai-Borwein Step-Size for Machine Learning*

Yi-Ming Yang[1], Fu-Sheng Wang[2*], Zheng Peng[1], Xiao-Jun Zhou[3]

[1]School of Mathematics and Computational Science, Xiangtan University, Xiangtan, 411105, China.
[2]School of Mathematics and Statistics, Taiyuan Normal University, Jinzhong, 030619, China.
[3]School of Science, Hebei University of Technology, Tianjin, 300401, China.

*Corresponding author(s). E-mail(s): fswang2005@163.com;
Contributing authors: 202231510130@smail.xtu.edu.cn;
pzheng@xtu.edu.cn; 2846727486@qq.com;

Abstract

As a mini-batch version of the SARAH algorithm, the MB-SARAH algorithm has received extensive attention due to its simple recursive scheme for updating stochastic gradient estimates. In this paper, we give a modification of the MB-SARAH method via cooperating with the BB step-size, shorted to MB-SARAH-BB. The MB-SARAH-BB combines some advantages of both MB-SARAH and BB methods, providing robustness in selecting initial step size during the optimization process. In the framework of MB-SARAH-BB, we propose a novel implementable method, Ada-MB-SARAH-BB, which utilizes adaptive probability for sampling in the mini-batch stochastic recursive gradient computation during the inner loop iteration. We establish the linear convergence of the MB-SARAH-BB and Ada-MB-SARAH-BB methods under some mild assumptions. Numerical experiments on standard machine learning datasets demonstrate

*This paper is supported by Research Project Foundation of Shanxi Scholarship Council of China(No.2017-104); Basic Research Program of Shanxi Province (Free exploration) project (No.202103021224303,20210302124688), and National Natural Science Foundation of China (12071398)

that, the MB-SARAH-BB is effective and more competitive than the recent successful stochastic gradient methods. In addition, numerical experiments also demonstrate that the performance of Ada-MB-SARAH-BB is generally better than or comparable to MB-SARAH-BB method.

Keywords: Machine learning, Mini batches, SARAH algorithm, BB step-size

MSC Classification: 90C15 · 90C25 · 90C30

1 Introduction

Many optimization problems arising in statistics and machine learning, such as support vector machine[1], logistic regression[2], neural networks[3], can be widely expressed as follows:

$$\min_{\omega\in\mathbb{R}^d} P(\omega) = \frac{1}{n}\sum_{i=1}^{n} f_i(\omega), \tag{1}$$

where n is the sample size, and each $f_i, i \in \{1,\ldots,n\}$ is cost function which aims at estimating how well parameter ω fits the data of the i-th sample. In this paper, we focus on such problem where each f_i is strongly convex and has Lipschitz continuous gradient. For example, given a training set $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, the cost function of the least squares regression model is $f_i(\omega) = \left(x_i^T\omega - y_i\right)^2$, where λ is a regularization parameter and $\|\cdot\|$ denotes l_2-norm; and in the l_2-regularized logistic regression for binary classification problem, the cost function is $f_i(\omega) = \log\left(1+\exp\left[-y_i x_i^T\omega\right]\right)$ $(y_i \in \{-1,1\})$.

Since the objective function is smooth, some classical optimization methods, such as gradient descent and Newton method, are often used for solving problem (1). However, if the sample size n is extremely large, deduce that the exact full gradient of $P(\omega)$ is computationally expensive, the stochastic gradient descent (SGD) method, which can be traced back to the seminal work by [4], is probably an efficient approach. At the t-th iteration, the classical SGD method updates the iterates as follows:

$$\omega_{t+1} = \omega_t - \eta_t \nabla f_{i_t}(\omega_t), \tag{2}$$

where $\eta_t > 0$ is a step size, and the index i_t is chosen randomly from $\{1,2,\ldots,n\}$, $\nabla f_{i_t}(\omega_t)$ denotes the sample gradient. The expectation of the stochastic gradient estimator $\nabla f_{i_t}(\omega_t)$ is usually regarded as an unbiased estimation of $\nabla P(\omega_t)$, *i.e.*, $\mathbb{E}\left[\nabla f_{i_t}(\omega_t)\right] = \nabla P(\omega_t)$. Unfortunately, in practice the randomness may introduce variance [5, 6]. The performance of the SGD method can be highly sensitive to the variance of sample gradients $\nabla f_{i_t}(\omega_t)$. Even in the case of that the objective function is well-defined (i.e. strongly convex and smooth), the classical SGD method only

has the sub-linear convergence rate [7]. Recently, a surge of methods to improve the performance of SGD have been developed, the most popular methods are gradient aggregation algorithms, e.g., the stochastic average gradient (SAG) method[8] and the SAGA method[9], they compute a stochastic gradient as an average of stochastic gradients evaluated at previous iterates and then store previous stochastic gradients at the expense of memory. The other related methods, such as the stochastic variance reduced gradient (SVRG) method [10], the stochastic dual coordinate ascent(SDCA) method [11], the accelerated mini-batch Prox-SVRG(Acc-Prox-SVRG) method [12] and the mini-batch semi-stochastic gradient descent(mS2GD) method[13], have faster convergence rate than that of SGD. In addition, there are also some biased estimators that exhibit excellent performance. For instance, the SARAH method [14] utilizes a simple framework for updating stochastic gradient estimates, and the MB-SARAH [15] is a mini-batch variant of SARAH that is suitable for non-convex problems. Another example is the SPIDER [16] method, which is a stochastic Path-Integrated Differential Estimator that can identify an approximate stationary point for non-convex stochastic optimization problems and has been shown to outperform other existing algorithms of the same type. All methods mentioned above are widely used in the machine learning community for solving problem (1), which can achieve linear convergence rate on strongly convex optimization problems.

A large number of numerical experiments show that the performance of SGD type methods are greatly affected by the step size selection. One common approach is using a constant step size. Although constant step sizes are frequently used, they require manual tuning and can be time-consuming in practice. Another common approach is to adopt diminishing step sizes that must satisfy

$$\sum_{t=1}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=1}^{\infty} \eta_t^2 < \infty. \tag{3}$$

However, it often leads to SGD with a severely slow convergence rate [17]. Recently, due to the BB approach [18] can adaptively update the step size and has good numerical efficiency, many researchers have turned to incorporating it into SGD type algorithms. Sopya et al. [19] presented several variants of the BB method for SGD to train the linear SVM. Tan et al. [20] introduced SGD-BB and SVRG-BB, which use the BB method to determine step size for SGD and SVRG, respectively. Li et al. [21] applied the BB method to calculate the step size for SARAH, and other researchers such as, Liu et al. [22] and Yang et al. [23–25] incorporated the BB method to compute step size for the variants of SGD type algorithms.

Sampling strategies play a crucial role in improving the performance of SGD during training. Uniform sampling can provide an unbiased estimate of the full gradient,

but its high variance negatively impacts optimization convergence. The existing methods like Prox-SVRG [26] and Prox-SDCA [27] use importance sampling to reduce stochastic variance and can achieve faster convergence rates. By utilizing the adaptive probability for sampling, AdaSVRG and AdaSAGA [28] can achieve faster convergence rates than the original SVRG and SAGA.

All methods mentioned above have good numerical performance. Motivated by these related works, we propose the MB-SARAH-BB method, which uses the mini-batch version of BB method of Dai [29] to automatically compute step size in the MB-SARAH. Additionally, we explore the utilization of a non-uniform sampling strategy, adaptive probability, for sampling in the mini-batch stochastic recursive gradient computation during the inner loop iteration of MB-SARAH-BB, lead an implementation, i.e., Ada-MB-SARAH-BB method.

Our main contributions in this paper can be summarized as follows:

1) We incorporate the mini-batch version of BB method into MB-SARAH[15], which leads to a modified mini-batch stochastic recursive gradient method called MB-SARAH-BB, and establish the convergence of MB-SARAH-BB method under some mild assumptions.

2) We propose a mini-batch extension, Ada-MB-SARAH-BB, which incorporate the adaptive sampling in the inner loop iteration, and establish the convergence.

3) Adopt MB-SARAH-BB method and Ada-MB-SARAH-BB method to logistic regression problem for binary classification in machine learning, numerical experiments on different datasets show the effectiveness of our proposed methods.

The rest of this paper is organized as follows. In Section 2, we briefly introduce some backgrounds of the BB step size, MB-SARAH method and then propose the MB-SARAH-BB method. In addition, we present the Ada-MB-SARAH-BB. In Section 3, we provide an analysis of the convergence of the MB-SARAH-BB and Ada-MB-SARAH-BB methods under strongly convex and non strongly convex conditions. In Section 4, we demonstrate the numerical experiments to illustrate the efficiency of the proposed methods for both strongly convex and non-strongly convex optimization problem. Finally, we conclude this paper in Section 5.

2 The Algorithms

In this section, we will introduce two cutting-edge methods: the MB-SARAH-BB method [15] and the BB method [29] in Section 2.1 and 2.2, respectively. Following that, we propose our MB-SARAH-BB method in Section 2.3, which builds upon the MB-SARAH method by incorporating the mini-batch version of BB step size to further enhance its performance. To improve the computational efficiency of the MB-SARAH-BB, we propose the Ada-MB-SARAH-BB method by combining a non-uniform sampling technique in section 2.4. These methods will be developed with the

aim of achieving better performance than previous approaches. We use the following notations hereafter: $v_0 = \nabla P(\omega_0) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\omega_0)$ and $\nabla P_S(\omega_t) = \frac{1}{|S|}\sum_{i\in S} \nabla f_i(\omega_t)$, where S is the sample-set.

2.1 The MB-SARAH Method

The MB-SARAH method, proposed by Nguyen et al.[15], uses a new kind of stochastic estimate of $\nabla P(\omega_t)$, i.e., $v_t = \nabla P_S(\omega_t) - \nabla P_S(\omega_{t-1}) + v_{t-1}$, which is a mini-batch version of SARAH, the framework can be described in Algorithm 1. It is worth noting that, the MB-SARAH method accepts the k-th iterate $\tilde{\omega}_k = \omega_t$ which is a uniformly randomly picked iterate from inner loop.

Algorithm 1 MB-SARAH

Input: initial point $\tilde{\omega}_0$, learning rate $\eta > 0$, update frequency m, samples sizes b.

Output: $\tilde{\omega}_k$

for $k = 1, 2, \ldots$ **do**
  $\omega_0 = \tilde{\omega}_{k-1}$
  $v_0 = \nabla P(\omega_0)$
  $\omega_1 = \omega_0 - \eta v_0$
  for $t = 1, 2, \ldots, m-1$ **do**
    Randomly choose a subset $S \subset \{1, \ldots, n\}$ of size b
    Update the stochastic recursive gradient:

$$v_t = \nabla P_S(\omega_t) - \nabla P_S(\omega_{t-1}) + v_{t-1}$$

    Update the iterate:

$$\omega_{t+1} = \omega_t - \eta v_t$$

  end for
  $\tilde{\omega}_k = \omega_t$ with t chosen uniformly randomly from $\{0, 1, \ldots, m-1\}$
end for

2.2 Barzilai-Borwein step zize

The well-known Barzilai-Borwein (BB) method, originally proposed by Barzilai and Borwein in [18], which tries to fit the objective by a quadratic model at each iteration and find the optimal step size. It is widely used to solve unconstrained optimization problem:

$$\min_{\omega\in\mathbb{R}^d} f(\omega). \tag{4}$$

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

$$\omega_{k+1} = \omega_k - \eta_k^{-1} \nabla f(\omega_k), \tag{5}$$

where $\nabla f(\omega_k)$ denotes the gradient of $f(\omega)$ at ω_k. Here η_k is introduced such that $\eta_k^{-1} I$ is an approximation to the Hessian matrix of $f(\omega)$ at ω_k, so it usually follows some properties of quasi-Newton method, and it is get by solving the following problem:

$$\min_{\eta} \|\eta^{-1} s_k - y_k\|_2 \quad or \quad \min_{\eta} \|s_k - \eta y_k\|_2, \tag{6}$$

where $s_k = \omega_k - \omega_{k-1}$, $y_k = \nabla f(\omega_k) - \nabla f(\omega_{k-1})$. It yields

$$\eta_k^{BB1} = \frac{s_k^T s_k}{s_k^T y_k} \quad or \quad \eta_k^{BB2} = \frac{s_k^T y_k}{y_k^T y_k}. \tag{7}$$

When $s_k^T y_k > 0$, it is easy to obtain that $\eta_k^{BB1} \geq \eta_k^{BB2}$ which means η_k^{BB1} is a more advantageous step size to decrease the objective function. Fletcher [30] showed that η_k^{BB1} is superior to η_k^{BB2}. Recently, Dai et al. [29] studied some numerical instances, and showed that η_k^{BB1} may not be the best choice. They proposed a family of spectral gradient methods whose step size is determined by a convex combination of η_k^{BB1} and η_k^{BB2}, that is,

$$\eta_k = \tau \eta_k^{BB1} + (1 - \tau) \eta_k^{BB2}, \tag{8}$$

where $\tau \in [0, 1]$. It is worth mentioning that there has been the alternative use of the BB step size formula [31], which is given by

$$\eta_k^{ABB} = \begin{cases} \eta_k^{BB2}, & \text{if } \eta_k^{BB2}/\eta_k^{BB1} \leq \kappa \\ \eta_k^{BB1}, & \text{otherwise} \end{cases} \tag{9}$$

where $\kappa > 0$ is a constant. For the other related works using the BB method to calculate the step size, the readers are referred to [32, 33]. In this paper, we incorporate the mini-batch version of above step size (8) to the MB-SARAH, which leads to the proposed MB-SARAH-BB method.

2.3 The MB-SARAH-BB Method

In this section, we propose the MB-SARAH-BB method, which uses the mini-batch version of BB method (8) to compute the step size η_k instead of using a prefixed η in MB-SARAH. The pseudocode of MB-SARAH-BB method is described in Algorithm 2.

2.3.1 Basic Assumption

Assumption 1. *Each f_i, $i = 1, 2, \ldots, n$, is convex and first-order continuously differentiable, and the gradient of each component function f_i is L_i-Lipschitz continuous, i.e., there exists $L_i > 0$ such that for any ω, $\omega' \in \mathbb{R}^d$,*

$$\|\nabla f_i(\omega) - \nabla f_i(\omega')\|_2 \leq L_i \|\omega - \omega'\|_2. \tag{10}$$

Assumption 1 implies that $\nabla P(\omega)$ is also L-Lipschitz continuous, i.e., there exists a constant $L > 0$ such that for any ω, $\omega' \in \mathbb{R}^d$,

$$\|\nabla P(\omega) - \nabla P(\omega')\|_2 \leq L \|\omega - \omega'\|_2. \tag{11}$$

Moreover, by the property of L-Lipschitz continuous function in [34],

$$P(\omega) \leq P(\omega') + \nabla P(\omega')^T (\omega - \omega') + \frac{L\|\omega - \omega'\|^2}{2}. \tag{12}$$

2.3.2 Strongly convex optimization

In the case of that the objective function is strongly convex, we calculate η_k via

$$\eta_k = \frac{b}{m}\left(\tau \eta_k^{BB1} + (1-\tau)\eta_k^{BB2}\right) \tag{13}$$

where

$$\eta_k^{BB1} = \frac{\|\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2}\|_2^2}{(\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2})^T \left(v_0^k - v_0^{k-1}\right)}, \eta_k^{BB2} = \frac{(\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2})^T \left(v_0^k - v_0^{k-1}\right)}{\|v_0^k - v_0^{k-1}\|_2^2}$$

2.3.3 Non-strongly convex optimization

If the objective function is non-strongly convex, we calculate η_k via

$$\eta_k = \frac{b}{m} \min\{\tau \eta_k^{BB1} + (1-\tau)\eta_k^{BB2}, \frac{1}{\rho}\} \tag{14}$$

where $\rho < L$.

Remark 1. For the first epoch, MB-SARAH-BB uses initial step size η_0, the update frequency m is required to be given and the step size η_k should be updated by using the mini-batch version of the convex combination of η_k^{BB1} and η_k^{BB2}.

Remark 2. If we always set $\eta_k = \eta$ instead of using the BB step size in Algorithm 2, the MB-SARAH-BB is reducing to the original MB-SARAH method, other than one difference stated in Remark 3.

Remark 3. In step 15 of the MB-SARAH-BB, accept the k-th iterate to be $\tilde{\omega}_k = \omega_m$, the last iterate of the inner loop. This is one difference between MB-SARAH-BB

Algorithm 2 MB-SARAH-BB

Input: initial point $\tilde{\omega}_0 \in \mathbb{R}^d$ and step size $\eta_1 > 0$, update frequency m and min-batch size b, parameter $\tau \in (0, 1]$.

Output: $\tilde{\omega}_k$

for $k = 1, 2, \ldots$ **do**
  $\omega_0 = \tilde{\omega}_{k-1}$
  $v_0 = \nabla P(\omega_0)$
  if $k > 1$
    calculate η_k using the mini-batch version of BB method
  end if
  $\omega_1 = \omega_0 - \eta_k v_0$
  for $t = 1, 2, \ldots, m-1$ **do**
    Randomly choose a subset $S \subset \{1, \ldots, n\}$ of size b
    Update the stochastic recursive gradient:

$$v_t^k = \nabla P_S(\omega_t) - \nabla P_S(\omega_{t-1}) + v_{t-1}^k \tag{15}$$

    Update the iterate:

$$\omega_{t+1} = \omega_t - \eta_k v_t^k$$

  end for
  $\tilde{\omega}_k = \omega_m$
end for

(Algorithm 2) and MB-SARAH (Algorithm 1), and it seems a more reasonable choice since the latest information in each inner loop is used.

2.4 The MB-SARAH-BB Method with Non-Uniform Sampling

The original MB-SARAH-BB method recursively updates the stochastic gradient step v_t by adding component gradients and subtracting from the previous v_{t-1} in the inner loop. Here, we employ a sampling scheme that explicitly compute adaptive probability at each iteration of MB-SARAH-BB. The pseudo code for the Ada-MB-SARAH-BB is shown below as Algorithm 3.

At the k^{th} iteration, we denote the stochastic gradient $\mathbf{v}_t^k$ of Ada-MB-SARAH-BB in a uniform way:

$$\mathbf{v}_t^k := \beta_i^k / n p_i^k + v_{t-1}^k, \tag{16}$$

where $\beta_i^k := \frac{1}{|S|}\sum_{i\in S}[\nabla f_i(\omega_t) - \nabla f_i(\omega_{t-1})]$. We define the adaptive probability as

$$p_i^k = \frac{\left\|\beta_i^k\right\|}{\sum_{i=1}^n \left\|\beta_i^k\right\|}, i = 1, \ldots, n, \tag{17}$$

i.e. f_i is sampled with probability proportional to $\left\|\beta_i^k\right\|$.

Algorithm 3 Ada-MB-SARAH-BB

Input: initial point $\tilde{\omega}_0 \in \mathbb{R}^d$ and step size $\eta_1 > 0$, update frequency m and min-batch size b, parameter $\tau \in (0, 1]$.

Output: $\tilde{\omega}_k$

for $k = 1, 2, \ldots$ **do**
  $\omega_0 = \tilde{\omega}_{k-1}$
  $v_0 = \nabla P(\omega_0)$
  if $k > 1$
    calculate η_k using the mini-batch version of BB method through (13) and (14)
  end if
  $\omega_1 = \omega_0 - \eta_k v_0$
  Probability $Q = \{p_1, p_2, ..., p_n\}$ on $\{1, \ldots, n\}$ according to (17)
  for $t = 1, 2, \ldots, m-1$ **do**
    Choose a subset $S \subset \{1, \ldots, n\}$ with size b, where each $i \in S$ is chosen from $\{1, \ldots, n\}$ randomly according to Q
    Update the stochastic recursive gradient:

$$v_t^k = \frac{1}{|S|}\sum_{i\in S}\left[\frac{1}{np_i}\nabla f_i(\omega_t) - \nabla f_i(\omega_{t-1})\right] + v_{t-1}^k \tag{18}$$

    Update the iterate:

$$\omega_{t+1} = \omega_t - \eta_k v_t^k$$

  end for
  $\tilde{\omega}_k = \omega_m$
end for

3 Convergence Analysis

In this section, we analyze the linear convergence of MB-SARAH-BB. Then, we show that Ada-MB-SARAH-BB also converges linearly for both strongly and non-strongly convex objective functions.

3.1 Convergence Results of Strongly Convex Functions

In this subsection, we establish the linear convergence of our proposed methods when objective function is strongly convex. We assume that each f_i is convex and the objective function $P(\omega)$ is μ-strongly convex, i.e., there exists $\mu > 0$ such that for all ω, $\omega' \in \mathbb{R}^d$

$$P(\omega) \geq P(\omega') + \nabla P(\omega')^T(\omega - \omega') + \frac{\mu\|\omega - \omega'\|^2}{2}. \tag{19}$$

When setting $\omega_* = \arg\min_{\omega} P(\omega)$, [24] implies the strong convexity of $P(\omega)$ as below

$$2\mu\left[P(\omega) - P(\omega_*)\right] \leq \|\nabla P(\omega)\|^2, \quad \forall\, \omega \in \mathbb{R}^d. \tag{20}$$

Under Assumption 1, we have $L \leq \frac{1}{n}\sum_{i=1}^{n} L_i$. For simplicity, we denote L_Ω as

$$L_\Omega = \max_{i\in\{1,2,\ldots,n\}} \frac{L_i}{np_i}. \tag{21}$$

Then, $L_\Omega \geq \frac{1}{n}\sum_{i=1}^{n} L_i \geq L$.

Lemma 1. *Suppose that Assumption 1 holds and $P(\omega)$ is μ-strongly convex. Then for all $m > 0$ and $b > 1$, we have*

$$\frac{b}{mL} \leq \eta_k \leq \frac{b}{m\mu}. \tag{22}$$

Proof. By Lipschitz continuity of $\nabla P(\omega)$, it is easy to obtain that

$$\eta_k^{BB1} \geq \frac{\|\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2}\|^2}{L\|\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2}\|^2} = \frac{1}{L},\ \eta_k^{BB2} \geq \frac{\|v_0^k - v_0^{k-1}\|^2}{L\|v_0^k - v_0^{k-1}\|^2} = \frac{1}{L}. \tag{23}$$

Thus the lower bound of η_k is that

$$\eta_k = \frac{b}{m}\left(\tau\eta_k^{BB1} + (1-\tau)\,\eta_k^{BB2}\right) \geq \frac{b}{mL}. \tag{24}$$

Meanwhile, the strong convexity of $P(\omega)$ indicates that

$$\eta_k^{BB1} = \frac{\|\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2}\|^2}{\left(\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2}\right)^T\left(v_0^k - v_0^{k-1}\right)} \leq \frac{\|\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2}\|^2}{\mu\|\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2}\|^2} = \frac{1}{\mu}. \tag{25}$$

The upper bound of η_k is given by

$$\eta_k = \frac{b}{m}\left(\tau\eta_k^{BB1} + (1-\tau)\,\eta_k^{BB2}\right) \leq \frac{b}{m\mu}.$$

□

Lemma 2. *Suppose that Assumption 1 holds and $P(\omega)$ is μ-strongly convex. If v_t^k is denoted by (15) in MB-SARAH-BB, then for all $m > 0$ and $b > 1$, we have*

$$\sum_{t=0}^{m} \mathbb{E}\left[\|\nabla P(\omega_t)\|^2\right] \leq \frac{2m\mu}{b}\mathbb{E}\left[P(\omega_0) - P(\omega_*)\right] + \sum_{t=0}^{m} \mathbb{E}\left[\|\nabla P(\omega_t) - v_t^k\|^2\right] - \left(1 - \frac{Lb}{m\mu}\right)\sum_{t=0}^{m} \mathbb{E}\left[\|v_t^k\|^2\right], \tag{26}$$

where w_* is a global minimizer of P.

Proof. By (12) and $w_{t+1} = w_t - \eta_k v_t^k$, we have

$$\mathbb{E}[P(w_{t+1})] \leq \mathbb{E}[P(w_t)] - \eta_k \mathbb{E}[\nabla P(w_t)^T v_t^k] + \frac{L\eta_k^2}{2}\mathbb{E}[\|v_t^k\|^2].$$

The upper bound of the BB step size is $\eta_k \leq \frac{b}{m\mu}$, which can be given by Lemma 1. Then, we obtain that

$$\begin{aligned}\mathbb{E}[P(w_{t+1})] &\leq \mathbb{E}[P(w_t)] - \frac{b}{m\mu}\mathbb{E}\left[\nabla P(w_t)^T v_t^k\right] + \frac{Lb^2}{2m^2\mu^2}\mathbb{E}[\|v_t^k\|^2] \\ &= \mathbb{E}[P(w_t)] - \frac{b}{2m\mu}\mathbb{E}\left[\|\nabla P(w_t)\|^2\right] + \frac{b}{2m\mu}\mathbb{E}[\|\nabla P(w_t) \\ &\quad - v_t^k\|^2] - \left(\frac{b}{2m\mu} - \frac{Lb^2}{2m^2\mu^2}\right)\mathbb{E}\left[\|v_t^k\|^2\right].\end{aligned}$$

The last equality follows that $a^T b = \frac{1}{2}\left[\|a\|^2 + \|b\|^2 - \|a-b\|^2\right]$. By summing over $t = 0, \ldots, m$, we have

$$\begin{aligned}\mathbb{E}[P(w_{m+1})] \leq \mathbb{E}[P(w_0)] &- \frac{b}{2m\mu}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla P(w_t)\|^2\right] + \frac{b}{2m\mu}\cdot\sum_{t=0}^{m}\mathbb{E}[\|\nabla P(w_t) - v_t^k\|^2] \\ &- \left(\frac{b}{2m\mu} - \frac{Lb^2}{2m^2\mu^2}\right)\sum_{t=0}^{m}\mathbb{E}\left[\|v_t^k\|^2\right].\end{aligned}$$

Further, we have

$$\sum_{t=0}^{m} \mathbb{E}\left[\|\nabla P(w_t)\|^2\right] \leq \frac{2m\mu}{b}\mathbb{E}[P(w_0) - P(w_{m+1})] + \sum_{t=0}^{m} \mathbb{E}\left[\|\nabla P(w_t) - v_t^k\|^2\right] - \left(1 - \frac{Lb}{m\mu}\right)\sum_{t=0}^{m} \mathbb{E}\left[\|v_t^k\|^2\right]$$

$$\leq \frac{2m\mu}{b}\mathbb{E}[P(w_0) - P(w_*)] + \sum_{t=0}^{m}\mathbb{E}\left[\|\nabla P(w_t) - v_t^k\|^2\right]$$
$$- \left(1 - \frac{Lb}{m\mu}\right)\sum_{t=0}^{m}\mathbb{E}\left[\|v_t^k\|^2\right],$$

where the last inequality follows $\omega_* = \arg\min_{\omega} P(\omega)$. □

With modification of Lemma 3 in [15], we obtain the following lemma showing the upper bound for $\mathbb{E}[\|\nabla P(w_t) - v_t^k\|^2]$.

Lemma 3. *Suppose that Assumption 1 holds and v_t^k is denoted by (15) in MB-SARAH-BB, then for all $t \geq 1$, $m > 0$ and $b > 1$,*

$$\mathbb{E}[\|\nabla P(w_t) - v_t^k\|^2] \leq \frac{L^2 b}{\mu^2 m^2}\left(\frac{n-b}{n-1}\right)\sum_{j=1}^{t}\mathbb{E}[\|v_{j-1}^k\|^2].$$

Lemma 4. *Suppose that Assumptions 1 holds, $P(\omega)$ is μ-strongly convex and w_m is generated by MB-SARAH-BB within one outer loop. Assume that the parameters, $m > 0$ and $b > 1$, are chosen such that*

$$\frac{L^2 b}{\mu^2 m}\left(\frac{n-b}{n-1}\right) - \left(1 - \frac{Lb}{m\mu}\right) \leq 0, \tag{27}$$

then we have.

$$\mathbb{E}[\|\nabla P(w_m)\|^2] \leq \frac{2m\mu}{b(m+1)}[P(w_0) - P(w_*)]$$

Proof. By Lemma 3, we have

$$\mathbb{E}\left[\|\nabla P(w_k) - v_k\|^2\right] \leq \frac{L^2 b}{\mu^2 m^2}\left(\frac{n-b}{n-1}\right)\sum_{j=1}^{k}\mathbb{E}\left[\|v_{j-1}\|^2\right].$$

Since $\|\nabla P(w_0) - v_0\|^2 = 0$, hence by summing over $k = 0, \ldots, m$, we obtain

$$\sum_{k=0}^{m}\mathbb{E}\left[\|\nabla P(w_k) - v_k\|^2\right] \leq \tfrac{L^2 b}{\mu^2 m^2}\left(\tfrac{n-b}{n-1}\right)\Big[m\mathbb{E}\left[\|v_0\|^2\right]$$
$$+(m-1)\mathbb{E}\left[\|v_1\|^2\right] + \ldots + \mathbb{E}\left[\|v_{m-1}\|^2\right]\Big].$$

Further, we have

$$\begin{aligned}
&\sum_{k=0}^{m}\mathbb{E}\left[\left\|\nabla P\left(w_k\right)-v_k\right\|^2\right]-\left(1-\tfrac{Lb}{m\mu}\right)\sum_{k=0}^{m}\mathbb{E}\left[\left\|v_k\right\|^2\right]\\
&\leq \tfrac{L^2b}{\mu^2m^2}\left(\tfrac{n-b}{n-1}\right)\left[m\mathbb{E}\left[\left\|v_0\right\|^2\right]+(m-1)\mathbb{E}\left[\left\|v_1\right\|^2\right]\right.\\
&\left.+\ldots+\mathbb{E}\left[\left\|v_{m-1}\right\|^2\right]\right]-\left(1-\tfrac{Lb}{m\mu}\right)\sum_{k=0}^{m}\mathbb{E}\left[\left\|v_k\right\|^2\right]\\
&\leq \tfrac{L^2b}{\mu^2m}\left(\tfrac{n-b}{n-1}\right)-\left(1-\tfrac{Lb}{m\mu}\right)\mathbb{E}\left[\left\|v_{k-1}\right\|^2\right]\leq 0
\end{aligned}$$

Therefore, by Lemma 1, we have

$$\begin{aligned}
\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla P(\omega_t)\|^2\right] &\leq \frac{2m\mu}{b}\mathbb{E}\left[P(\omega_0)-P(\omega_*)\right]+\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla P(\omega_t)-v_t^k\|^2\right]\\
&\quad-\left(1-\frac{Lb}{m\mu}\right)\sum_{t=0}^{m}\mathbb{E}\left[\|v_t^k\|^2\right]\\
&\leq \frac{2m\mu}{b}\mathbb{E}\left[P(\omega_0)-P(\omega_*)\right],
\end{aligned}$$

By the definition of $\tilde{w}_k$ in Algorithm 2 and $\tilde{w}_k = w_m$, we have that

$$\begin{aligned}
\mathbb{E}\left[\left\|\nabla P\left(w_m\right)\right\|^2\right] &= \frac{1}{m+1}\sum_{k=0}^{m}\mathbb{E}\left[\left\|\nabla P\left(w_k\right)\right\|^2\right]\\
&\leqslant \frac{2m\mu}{b(m+1)}\mathbb{E}\left[P\left(w_0\right)-P\left(w_*\right)\right]
\end{aligned}$$

□

We now establish the linear convergence in expectation of the MB-SARAH-BB method with multiple outer loops in Theorem 1.

Theorem 1. *Suppose that Assumption 1 holds, $P(\omega)$ is μ-strongly convex and $\{\widetilde{w}_k\}$ is generated by MB-SARAH-BB. Assume that the parameters, $m > 0$ and $b > 1$, are chosen such that*

$$\frac{L^2b}{\mu^2m}\left(\frac{n-b}{n-1}\right)-\left(1-\frac{Lb}{m\mu}\right)\leq 0, \tag{28}$$

then we have

$$\mathbb{E}[\|\nabla P(\widetilde{w}_k)\|^2]\leq \gamma^k\|\nabla P(\widetilde{w}_0)\|^2,$$

where $\gamma = \frac{1}{b}$. Note that mini-batch size, b, is always greater than 1. Hence, we easily derive $\gamma < 1$, which means that the MB-SARAH-BB has linear convergence rate in expectation.

Proof. Note that $w_0 = \widetilde{w}_{k-1}$ and $\widetilde{w}_k = w_m$, $k \geq 1$. We obtain

$$\begin{aligned}\mathbb{E}[\|\nabla P(\widetilde{w}_k)|\widetilde{w}_{k-1}\|^2] &= \mathbb{E}[\|\nabla P(\widetilde{w}_k)|w_0\|^2] \\ &\leq \frac{2m\mu}{b(m+1)}\mathbb{E}[P(w_0) - P(w_*)] \\ &\leq \frac{m}{b(m+1)}\|\nabla P(w_0)\|^2 \\ &= \frac{m}{b(m+1)}\|\nabla P(\widetilde{w}_{k-1})\|^2.\end{aligned}$$

Hence, taking expectation,

$$\begin{aligned}\mathbb{E}[\|\nabla P(\widetilde{w}_k)\|^2] &\leq \frac{m}{b(m+1)}\mathbb{E}[\|\nabla P(\widetilde{w}_{k-1})\|^2] \\ &\leq \left[\frac{1}{b}\right]^k \|\nabla P(\widetilde{w}_0)\|^2.\end{aligned}$$

□

Theorem 2. *Suppose that Assumption 1 holds, $P(\omega)$ is μ-strongly convex and $\{\widetilde{w}_k\}$ are generated by Ada-MB-SARAH-BB. If v_t^k is denoted by (18) in Ada-MB-SARAH-BB. Assume that the parameters, $m > 0$ and $b > 1$, are chosen such that*

$$\frac{L_\Omega^2 b}{\mu^2 m}\left(\frac{n-b}{n-1}\right) - \left(1 - \frac{L_\Omega b}{m\mu}\right) \leq 0, \tag{29}$$

then we have

$$\mathbb{E}[\|\nabla P(\widetilde{w}_k)\|^2] \leq \gamma^k \|\nabla P(\widetilde{w}_0)\|^2,$$

where $\gamma = \frac{1}{b} \in (0,1)$, which means that the Ada-MB-SARAH-BB has linear convergence rate in expectation.

Proof. By Lemma 3 and (21), we have

$$\mathbb{E}[\|\nabla P(w_t) - v_t^k\|^2] \leq \frac{L^2 b}{\mu^2 m^2}\left(\frac{n-b}{n-1}\right)\sum_{j=1}^{t}\mathbb{E}[\|v_{j-1}^k\|^2] \leq \frac{L_\Omega^2 b}{\mu^2 m^2}\left(\frac{n-b}{n-1}\right)\sum_{j=1}^{t}\mathbb{E}[\|v_{j-1}^k\|^2].$$

From Lemma 4, if we choose the parameters such that

$$\frac{L_\Omega^2 b}{\mu^2 m}\left(\frac{n-b}{n-1}\right) - \left(1 - \frac{L_\Omega b}{m\mu}\right) \leq 0.$$

Then,

$$\mathbb{E}\left[\left\|\nabla P\left(w_m\right)\right\|^2\right] = \frac{1}{m+1}\sum_{k=0}^{m}\mathbb{E}\left[\left\|\nabla P\left(w_k\right)\right\|^2\right]$$
$$\leqslant \frac{2m\mu}{b(m+1)}\mathbb{E}\left[P\left(w_0\right) - P\left(w_*\right)\right]$$

Note that $w_0 = \widetilde{w}_{k-1}$ and $\widetilde{w}_k = w_m$, $k \geq 1$. Hence,

$$\mathbb{E}[\|\nabla P(\widetilde{w}_k)\|^2] \leq \frac{m}{b(m+1)}\mathbb{E}[\|\nabla P(\widetilde{w}_{k-1})\|^2]$$
$$\leq \left[\frac{1}{b}\right]^k \|\nabla P(\widetilde{w}_0)\|^2.$$

□

3.2 Convergence Results of Non-strongly Convex Functions

In this part, we establish linear convergence of our MB-SARAH-BB method for non-strongly convex functions satisfying the Polyak-Łojasiewicz inequality [36]. That is

$$\frac{1}{2}\|\nabla P(w)\|^2 \geqslant \nu\left(P(w) - P^*\right), \tag{30}$$

where $\nu > 0$, P is the optimal value.

Lemma 5. *Suppose that Assumption 1 hold, $P(\omega)$ satisfies the Polyak-Łojasiewicz inequality and $\nabla P(\omega)$ is L-Lipschitz continuous. Then for all $k, m > 0$ we have*

$$\frac{b}{mL} \leq \eta_k \leq \frac{b}{m\rho}. \tag{31}$$

Proof. By Lipschitz continuity of $\nabla P\left(\omega\right)$, it is easy to obtain that

$$\eta_k^{BB1} \geq \frac{\|\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2}\|^2}{L\|\tilde{\omega}_{k-1} - \tilde{\omega}_{k-2}\|^2} = \frac{1}{L},\ \eta_k^{BB2} \geq \frac{\|v_0^k - v_0^{k-1}\|^2}{L\|v_0^k - v_0^{k-1}\|^2} = \frac{1}{L}. \tag{32}$$

By definition of (14),

$$\eta_k = \frac{b}{m}\min\{\tau\eta_k^{BB1} + (1-\tau)\,\eta_k^{BB2}, \frac{1}{\rho}\}$$

Thus we obtain that

$$\frac{b}{mL} \leq \eta_k \leq \frac{b}{m\rho}.$$

□

Lemma 6. *Suppose that Assumption 1 holds, $P(\omega)$ satisfies the Polyak-Łojasiewicz inequality and w_m is generated by MB-SARAH-BB within one outer loop. If v_t^k is denoted by (15) in MB-SARAH-BB, then for all $m > 0$ and $b > 1$, we have*

$$\begin{aligned}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla P(\omega_t)\|^2\right] \leq \frac{2m\rho}{b}\mathbb{E}\left[P(\omega_0) - P(\omega_*)\right] + \sum_{t=0}^{m}\mathbb{E}\left[\|\nabla P(\omega_t) - v_t^k\|^2\right] \\ - \left(1 - \frac{Lb}{m\rho}\right)\sum_{t=0}^{m}\mathbb{E}\left[\|v_t^k\|^2\right],\end{aligned} \tag{33}$$

where w_* is a global minimizer of P.

Proof. By (12) and $w_{t+1} = w_t - \eta_k v_t^k$, we have

$$\mathbb{E}[P(w_{t+1})] \leq \mathbb{E}[P(w_t)] - \eta_k\mathbb{E}[\nabla P(w_t)^T v_t^k] + \frac{L\eta_k^2}{2}\mathbb{E}[\|v_t^k\|^2].$$

The upper bound of the BB step size is given by Lemma 5, which is $\eta_k \leq \frac{b}{m\rho}$. So, we get

$$\begin{aligned}\mathbb{E}[P(w_{t+1})] &\leq \mathbb{E}[P(w_t)] - \frac{b}{m\rho}\mathbb{E}\left[\nabla P(w_t)^T v_t^k\right] + \frac{Lb^2}{2m^2\rho^2}\mathbb{E}[\|v_t^k\|^2] \\ &= \mathbb{E}[P(w_t)] - \frac{b}{2m\rho}\mathbb{E}\left[\|\nabla P(w_t)\|^2\right] + \frac{b}{2m\rho}\mathbb{E}[\|\nabla P(w_t) \\ &\quad - v_t^k\|^2] - \left(\frac{b}{2m\rho} - \frac{Lb^2}{2m^2\rho^2}\right)\mathbb{E}\left[\|v_t^k\|^2\right].\end{aligned}$$

The last equality follows that $a^T b = \frac{1}{2}\left[\|a\|^2 + \|b\|^2 - \|a-b\|^2\right]$. By summing over $t = 0, \ldots, m$, we have

$$\begin{aligned}\mathbb{E}[P(w_{m+1})] \leq \mathbb{E}[P(w_0)] - \frac{b}{2m\rho}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla P(w_t)\|^2\right] + \frac{b}{2m\rho}\cdot\sum_{t=0}^{m}\mathbb{E}[\|\nabla P(w_t) - v_t^k\|^2] \\ - \left(\frac{b}{2m\rho} - \frac{Lb^2}{2m^2\rho^2}\right)\sum_{t=0}^{m}\mathbb{E}\left[\|v_t^k\|^2\right].\end{aligned}$$

Further, we have

$$\begin{aligned}\sum_{t=0}^{m}\mathbb{E}\left[\|\nabla P(w_t)\|^2\right] \leq \frac{2m\rho}{b}\mathbb{E}[P(w_0) - P(w_{m+1})] + \sum_{t=0}^{m}\mathbb{E}\left[\|\nabla P(w_t) - v_t^k\|^2\right] \\ - \left(1 - \frac{Lb}{m\rho}\right)\sum_{t=0}^{m}\mathbb{E}\left[\|v_t^k\|^2\right]\end{aligned}$$

$$\leq \frac{2m\rho}{b}\mathbb{E}[P(w_0) - P(w_*)] + \sum_{t=0}^{m} \mathbb{E}\left[\|\nabla P(w_t) - v_t^k\|^2\right]$$
$$- \left(1 - \frac{Lb}{m\rho}\right) \sum_{t=0}^{m} \mathbb{E}\left[\|v_t^k\|^2\right],$$

where the last inequality follows $\omega_* = \arg\min_{\omega} P(\omega)$. □

With modification of Lemma 3 in [15], we obtain the following lemma showing the upper bound for $\mathbb{E}[\|\nabla P(w_t) - v_t^k\|^2]$.

Lemma 7. *Suppose that Assumption 1 holds, $P(\omega)$ satisfies the Polyak-Łojasiewicz inequality and v_t^k is denoted by (15) in MB-SARAH-BB, then for all $t \geq 1$, $m > 0$ and $b > 1$,*

$$\mathbb{E}[\|\nabla P(w_t) - v_t^k\|^2] \leq \frac{L^2 b}{\rho^2 m^2}\left(\frac{n-b}{n-1}\right) \sum_{j=1}^{t} \mathbb{E}[\|v_{j-1}^k\|^2].$$

We now establish the linear convergence in expectation of the MB-SARAH-BB method with multiple outer loops in Theorem 3.

Theorem 3. *Suppose that Assumption 1 holds,$P(\omega)$ satisfies Polyak-Łojasiewicz inequality (30) and $\{\widetilde{w}_k\}$ are generated by MB-SARAH-BB. If v_t^k is denoted by (15) in MB-SARAH-BB. Assume that the parameters, $m > 0$ and $b > 1$, are chosen such that*

$$\frac{L^2 b}{\rho^2 m}\left(\frac{n-b}{n-1}\right) - \left(1 - \frac{Lb}{m\rho}\right) \leq 0, \tag{34}$$

then we have

$$\mathbb{E}[\|\nabla P(\widetilde{w}_k)\|^2] \leq \gamma^k \|\nabla P(\widetilde{w}_0)\|^2,$$

where $\gamma = \frac{\rho}{b\nu}$, if we choose that $b > \frac{\rho}{\nu}$, then the MB-SARAH-BB has linear convergence rate in expectation.

Proof. Note that $w_0 = \widetilde{w}_{k-1}$ and $\widetilde{w}_k = w_m$, $k \geq 1$. We obtain

$$\begin{aligned}
\mathbb{E}[\|\nabla P(\widetilde{w}_k)|\widetilde{w}_{k-1}\|^2] &= \mathbb{E}[\|\nabla P(\widetilde{w}_k)|w_0\|^2] \\
&\leq \frac{2m\rho}{b(m+1)}\mathbb{E}[P(w_0) - P(w_*)] \\
&\leq \frac{\rho m}{b\nu(m+1)}\|\nabla P(w_0)\|^2 \\
&< \frac{\rho}{b\nu}\|\nabla P(\widetilde{w}_{k-1})\|^2.
\end{aligned}$$

Hence, taking expectation, we obtain

$$\begin{aligned}\mathbb{E}[\|\nabla P(\widetilde{w}_k)\|^2] &\leq \frac{\rho}{b\nu}\mathbb{E}[\|\nabla P(\widetilde{w}_{k-1})\|^2] \\ &\leq \left[\frac{\rho}{b\nu}\right]^k \|\nabla P(\widetilde{w}_0)\|^2.\end{aligned}$$

□

Theorem 4. *Suppose that Assumption 1 holds, $P(\omega)$ satisfies Polyak-Łojasiewicz inequality (30) and $\{\widetilde{w}_k\}$ are generated by Ada-MB-SARAH-BB. If v_t^k is denoted by (18) in Ada-MB-SARAH-BB. Assume that the parameters, $m > 0$ and $b > 1$, are chosen such that*

$$\frac{L_\Omega^2 b}{\mu^2 m}\left(\frac{n-b}{n-1}\right) - \left(1 - \frac{L_\Omega b}{m\mu}\right) \leq 0, \tag{35}$$

then we have

$$\mathbb{E}[\|\nabla P(\widetilde{w}_k)\|^2] \leq \gamma^k \|\nabla P(\widetilde{w}_0)\|^2,$$

where $\gamma = \frac{\rho}{b\nu}$, if we choose that $b > \frac{\rho}{\nu}$, then the Ada-MB-SARAH-BB has linear convergence rate in expectation.

Proof. By Lemma 7 and (21), we have

$$\mathbb{E}[\|\nabla P(w_t) - v_t^k\|^2] \leq \frac{L^2 b}{\rho^2 m^2}\left(\frac{n-b}{n-1}\right)\sum_{j=1}^{t}\mathbb{E}[\|v_{j-1}^k\|^2] \leq \frac{L_\Omega^2 b}{\rho^2 m^2}\left(\frac{n-b}{n-1}\right)\sum_{j=1}^{t}\mathbb{E}[\|v_{j-1}^k\|^2].$$

From Lemma 7, if we choose the parameters such that

$$\frac{L_\Omega^2 b}{\rho^2 m}\left(\frac{n-b}{n-1}\right) - \left(1 - \frac{L_\Omega b}{m\rho}\right) \leq 0.$$

Note that $w_0 = \widetilde{w}_{k-1}$ and $\widetilde{w}_k = w_m$, $k \geq 1$. Hence,

$$\begin{aligned}\mathbb{E}[\|\nabla P(\widetilde{w}_k)\|^2] &\leq \frac{\rho}{b\nu}\mathbb{E}[\|\nabla P(\widetilde{w}_{k-1})\|^2] \\ &\leq \left[\frac{\rho}{b\nu}\right]^k \|\nabla P(\widetilde{w}_0)\|^2.\end{aligned}$$

□

4 Experiments

In this section, we present numerical results to demonstrate the efficiency of our MB-SARAH-BB and Ada-MB-SARAH-BB methods.

4.1 Experimental Settings

We apply MB-SARAH-BB method to solve the following l_2-regularized logistic regression problem for binary classification in machine learning:

$$\min_{w\in\mathbb{R}^d} \quad P(w) := \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i x_i^T w)) + \frac{\lambda}{2}\|w\|^2, \tag{36}$$

where $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \{+1, -1\}^n$ is a collection of training examples. Table 1 shows the basic data including the size n, dimension d, and regularization parameter λ, which can be downloaded from the LIBSVM website [1].

Table 1 DATA INFORMATION OF EXPERIMENTS

Datasets	Instances (n)	Features (d)	λ
splice	1000	60	10^{-2}
mushrooms	8124	112	10^{-2}
ijcnn1	49,990	22	10^{-2}
phishing	11,055	68	10^{-2}
covtype	581,012	54	10^{-2}
w8a	49,749	300	10^{-2}

For fair comparison, all the tests have been performed on an Intel Core i7 processor with 10GB RAM under the Python computing environment. In following figures, we use the horizontal axis represents the number of effective passes over the data, where each effective passes evaluates n component gradients. The vertical axis denotes the $||\nabla P(w)||^2$ in Fig.1 to Fig.8. Here, ω_* is obtained by running MB-SARAH with the best-tuned step size until it converges. In addition, all the compared methods use the same initial point $\omega_0 = (0, 0, ..., 0)$. For our methods, the parameter τ is used for all the six datasets. Best-tuned parameters are used for other methods.

4.2 Empirical Study on Logistic Regression Problem

In Fig.1 to Fig.3, we first analyze the influence of batch size b, initial step size η_0 and the parameter τ for MB-SARAH-BB method. In Fig.4 to Fig.6, we compare the MB-SARAH-BB with other related methods for solving binary classification problem with

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

objective function defined in (36). In the Ada-MB-SARAH-BB, an adaptive version of the MB-SARAH-BB, two choices of sampling density p_i will be tested. In the case of that $f_i(\omega) = \log\left(1 + \exp\left[-y_i x_i^T \omega\right]\right)$, we have $\|\nabla\phi_i(w)\|_2 \leqslant \|x_i\|_2 \leqslant \sqrt{d}\,\|x_i\|_\infty$ due to $y_i \in \{-1, 1\}$, we choose

$$p_i^k = \frac{\left\|x_i^k\right\|_\infty}{\sum_{j=1}^n \left\|x_j^k\right\|_\infty}.$$

The second choice is (17), i.e.,

$$p_i^k = \frac{\left\|\beta_i^k\right\|}{\sum_{i=1}^n \left\|\beta_i^k\right\|}, i = 1, \ldots, n.$$

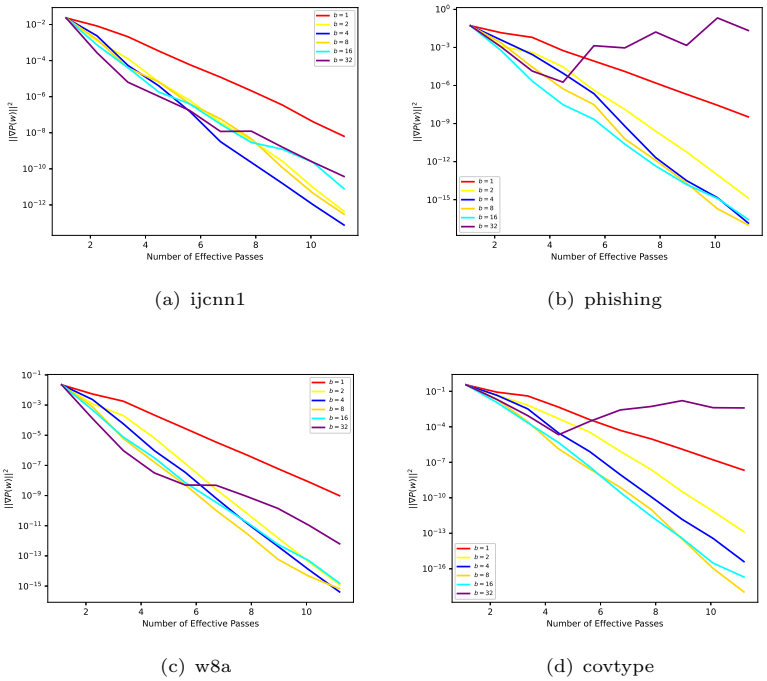
The choice of $p_i^k = \frac{\left\|x_i^k\right\|_\infty}{\sum_{j=1}^n \left\|x_j^k\right\|_\infty}$ is corresponding to Ada-MB-SARAH-BB-I, and the choice of $p_i^k = \frac{\left\|\beta_i^k\right\|}{\sum_{i=1}^n \left\|\beta_i^k\right\|}, i = 1, \ldots, n$ is corresponding to Ada-MB-SARAH-BB-II. In Fig.7 to Fig.8, we make comparison of Ada-MB-SARAH-BB methods with MB-SARAH-BB and MB-SARAH for solving both strongly and non-strongly convex optimization problems. Here we set $\lambda = 0$ in logistic regression problem (36) for non-strongly convex optimization problem.

4.2.1 Mini-batch Sizes and Initial Stepsizes

Firstly, we investigate the effect of mini-batch sizes of MB-SARAH-BB on the ijcnn1, phishing, w8a and covtype. One can find in Fig.1 that, by increasing the mini-batch size to b = 2, 4, 8, 16 and 32, the performance of MB-SARAH-BB is better than or comparable to that with b = 1. In addition, for phishing and covtype, the performance of MB-SARAH-BB improves as the mini-batch size increases to 8. However, a large mini-batch size, say 32, may deteriorate the performance.

Secondly, we investigate the effect of initial stepsize η_0 of MB-SARAH-BB on the ijcnn1, phishing, w8a and covtype. Three different values are tested for MB-SARAH-BB with b = 4. As shown in Fig.2, the MB-SARAH-BB is not sensitive to the initial stepsizes, which is favorable in practice.

At the last, we test the effect of parameter τ, the parameter for hybrid BB stepsize. To this end, MB-SARAH-BB with b = 4 is tested on the ijcnn1, phishing, w8a and covtype with τ chosen from 0.1 to 0.9. Fig.3 shows that, the MB-SARAH-BB is also not sensitive on τ.



(a) ijcnn1 (b) phishing

(c) w8a (d) covtype

Fig. 1 The performance of MB-SARAH-BB with respect to different batch sizes b.

4.2.2 Comparison of the MB-SARAH-BB and MB-SARAH

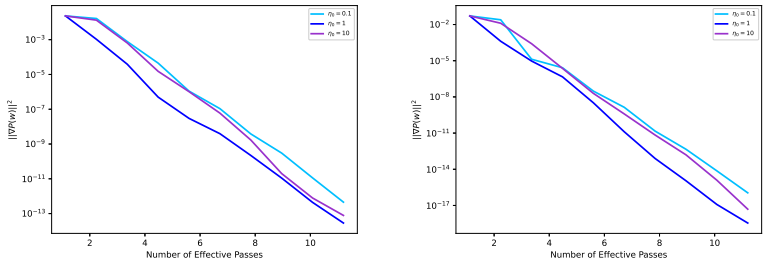
In this part, we compare MB-SARAH-BB with MB-SARAH for solving strongly convex problem and non-strongly convex problemwith respect to ijcnn1, phishing, w8a and covtype, respectively. The results shows that, as displayed in Fig.4, the performance of MB-SARAH is closely related to stepsize (fixed), whereas MB-SARAH-BB is not sensitive to the initial stepsize. One can also conclude from Fig.4 and Fig.5 that, the MB-SARAH-BB method is better than MB-SARAH for strongly convex problem, and more competitive and effective for non-strongly convex problem.

4.2.3 Comparison with other related methods

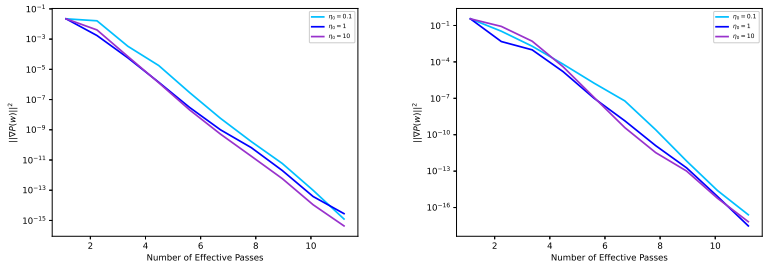
To further demonstrate the efficiency of our proposed method, we compare the MB-SARAH-BB method on four datasets, ijcnn1, phishing, w8a and covtype with the following methods:

1) **MB-SARAH-ABB**: The MB-SARAH using the adaptive BB step size η_k^{ABB} in (9).

2) **MB-SARAH-BB1**: The MB-SARAH with mini-batch version η_k^{BB1} in (7).



(a) ijcnn1 (b) phishing



(c) w8a (d) covtype

Fig. 2 The performance of MB-SARAH-BB with different initial step sizes.

3) **SARAH-I-BB**: A linearly convergent stochastic recursive gradient method with BB step size for convex optimization proposed in Liu, et al [22], using the same settings on parameters.

4) **SARAH-BB**: Stochastic recursive gradient method with the BB step size η_k^{BB1} in (7).

5) **mS2GD-BB**: A batch version of SVRG-BB proposed in [23], with the same settings on parameters.

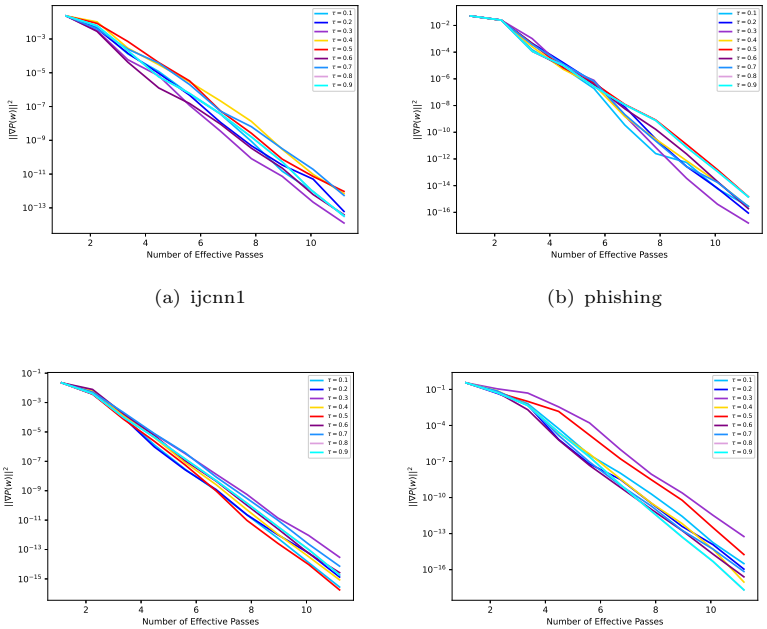
6) **SVRG-BB**: Stochastic variance reduced gradient method with BB step size [20].

7) **STSG**: Stochastic variance reduced gradient method with ABB step size [35].

8) **SVRG**: Stochastic variance reduced gradient method [10].

9) **SARAH**: Stochastic recursive gradient method [14].

Fig.6 illustrates that the performance of MB-SARAH-BB is significantly superior to some modern stochastic gradient methods that use the fixed step size, such as SVRG and SARAH. Additionally, it also can be seen that, the MB-SARAH-BB is generally better or slightly better than the other successful stochastic gradient methods which calculate step size using BB or its variants.



(a) ijcnn1 (b) phishing

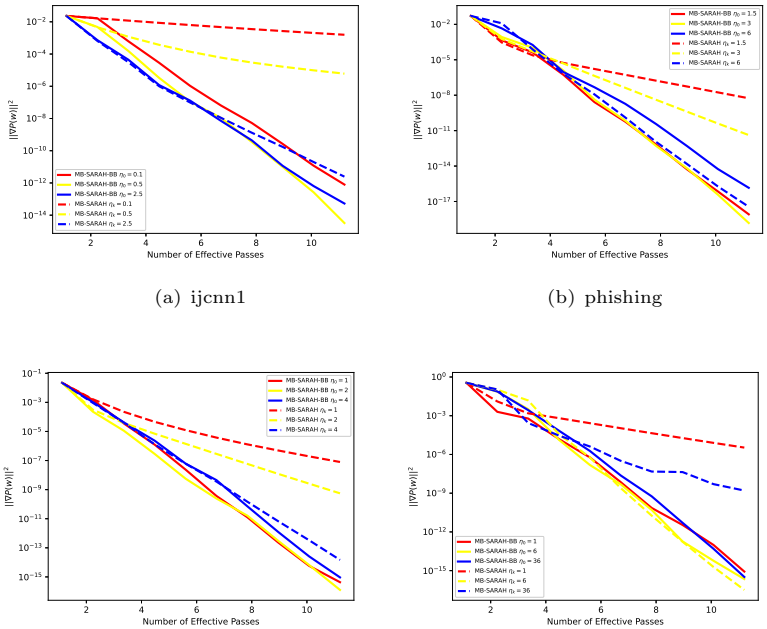
(c) w8a (d) covtype

Fig. 3 The performance of MB-SARAH-BB with different τ.

4.2.4 Comparison of Ada-MB-SARAH method with MB-SARAH-BB and MB-SARAH

Firstly, we make comparison of Ada-MB-SARAH-BB with MB-SARAH-BB and MB-SARAH for solving strongly convex optimization problem (36) on splice, mushrooms, ijcnn1, phishing, w8a and covtype. Three different methods are tested with $b = 4$ and initial stepsize $\eta_0 = 0.5$. As can be seen from Fig.7, the performance of Ada-MB-SARAH-BB method significantly outperforms MB-SARAH, while performs slightly better than MB-SARAH-BB.

Secondly, we present the comparison results of Ada-MB-SARAH-BB and MB-SARAH for solving non-strongly convex optimization problem. From the results in Fig.8, we see that Ada-MB-SARAH-BB method outperforms MB-SARAH for solving the non-strongly convex problem.



(a) ijcnn1 (b) phishing

(c) w8a (d) covtype

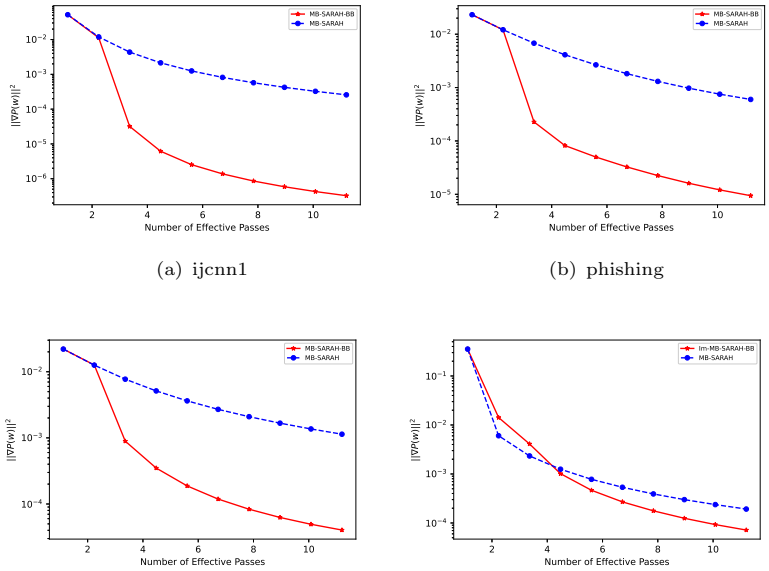
Fig. 4 Comparison of MB-SARAH-BB and MB-SARAH w.r.t different step sizes. (strongly convex)

5 Conclusions

We first proposed a modified MB-SARAH-BB algorithm, which can automatically update the step size by taking the better advantages of MB-SARAH and the mini-batch version of BB method. Furthermore, we proposed a mini-batch extension, Ada-MB-SARAH-BB, which utilizes adaptive probability for sampling in the mini-batch stochastic recursive gradient computation during the inner loop iteration of MB-SARAH-BB, which is more flexible than the uniform sampling choice in MB-SARAH-BB.

We established the linear convergence in expectation for the MB-SARAH-BB and Ada-MB-SARAH-BB under the strongly and non-strongly convex conditions. Compared with existing algorithms, the MB-SARAH-BB and Ada-MB-SARAH-BB is simple and with good theoretical properties.

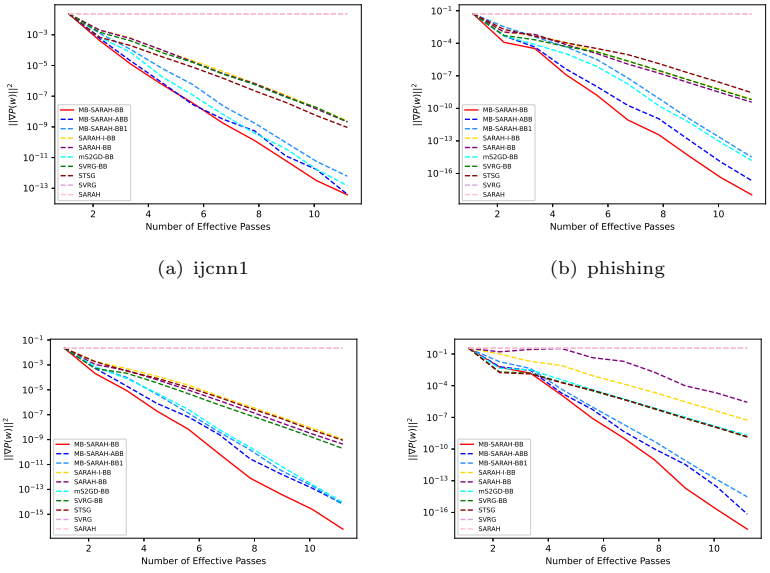
Numerical results indicate that the MB-SARAH-BB is robust to the selection of the initial step sizes, and are more effective and competitive than the modern stochastic gradient methods. We also discussed the effect of different mini-batch sizes b on the performances of MB-SARAH-BB and then give a suggestion on how to choose the better parameter b in practice. Additional numerical results indicate that



(a) ijcnn1 (b) phishing

(c) w8a (d) covtype

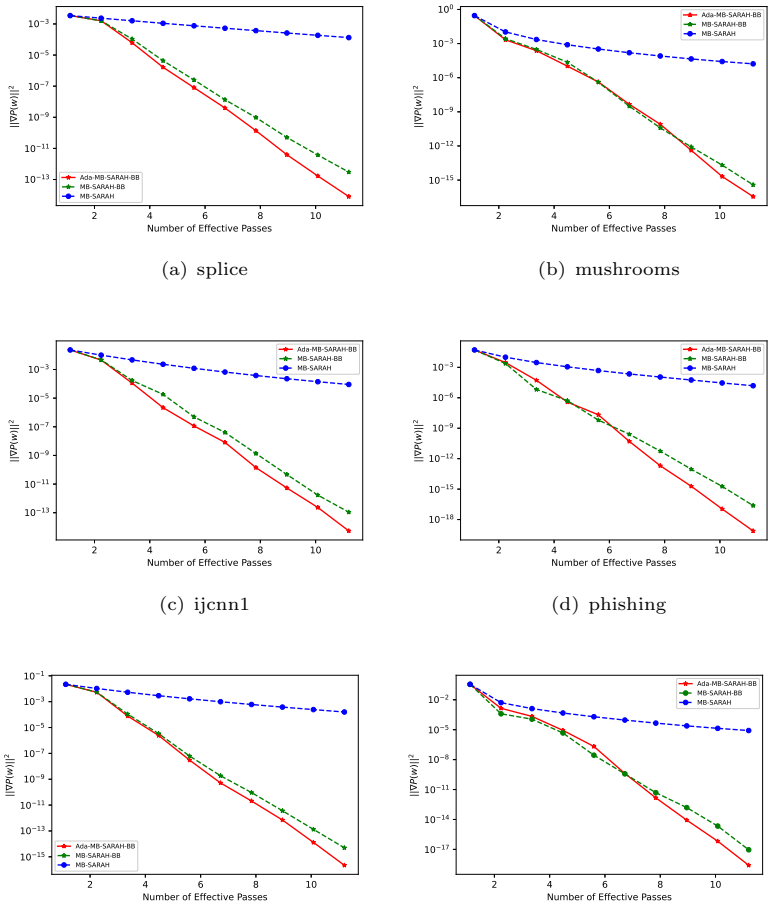
Fig. 5 Comparison of MB-SARAH-BB and MB-SARAH on non-strongly convex problem.



(a) ijcnn1 (b) phishing

(c) w8a (d) covtype

Fig. 6 Comparison of MB-SARAH-BB with different methods.



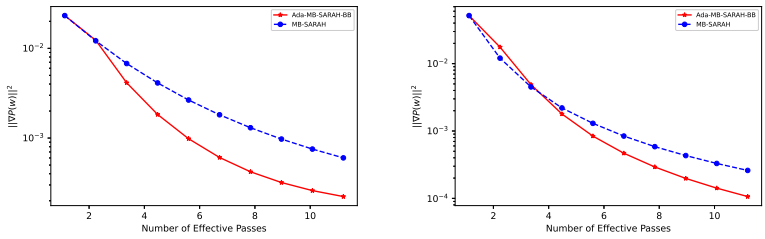
(a) splice (b) mushrooms

(c) ijcnn1 (d) phishing

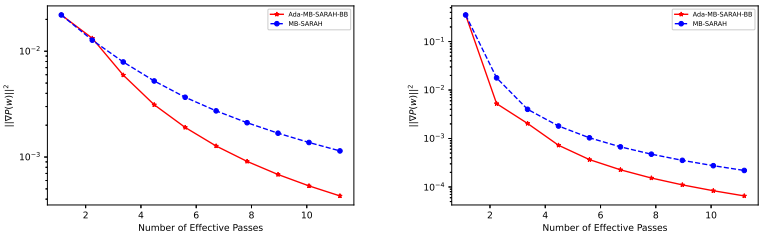
(e) w8a (f) covtype

Fig. 7 Comparison of Ada-MB-SARAH-BB, MB-SARAH-BB and MB-SARAH.

the performance of Ada-MB-SARAH-BB is better than and sometimes comparable to MB-SARAH-BB method.



(a) splice (b) mushrooms



(c) phishing (d) w8a

Fig. 8 Comparison of Ada-MB-SARAH-BB and MB-SARAH on non-strongly convex optimization problem.

References

[1] Xue, Z., Zhang, R., Qin, C. et al. An adaptive twin support vector regression machine based on rough and fuzzy set theories. Neural Comput and Applic. 32(9), 4709–4732 (2020)

[2] Chen H, Wu H C, Chan S C, et al. A stochastic quasi-Newton method for large-scale nonconvex optimization with applications. IEEE transactions on neural networks and learning systems. 32, 4776-4790 (2019)

[3] Dean J, Corrado G, Monga R, et al. Large scale distributed deep networks. In: Advances in Neural information processing systems, pp. 1232-1240 (2012)

[4] Robbins H, Monro S. A stochastic approximation method. The annals of mathematical statistics. 22(3), 400-407 (1951)

[5] Bottou L, Curtis F E, Nocedal J. Optimization methods for large-scale machine learning. Siam Review. 60(2), 223-311 (2018)

[6] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 521, 436-444 (2015)

[7] Moulines E, Bach F R. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In: Advances in Neural Information Processing Systems, pp. 451-459 (2011)

[8] Schmidt M, Le Roux N, Bach F. Minimizing finite sums with the stochastic average gradient. Mathematical Programming. 162(1), 83-112 (2017)

[9] Defazio A, Bach F, Lacoste-Julien S. SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In: Advances in Neural Information Processing Systems, pp. 1646-1654 (2014)

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

[11] Shalev-Shwartz S, Zhang T. Stochastic dual coordinate ascent methods for regularized loss minimization. Journal of Machine Learning Research. 14(1), 567-599 (2013)

[12] Nitanda, A. Stochastic proximal gradient descent with acceleration techniques. In: Advances in Neural Information Processing Systems, pp. 1574-1582 (2014)

[13] Konečný J, Liu J, Richtárik P, Takáč M. Mini-batch semi-stochastic gradient descent in the proximal setting. IEEE Journal of Selected Topics in Signal Processing. 10(2), 242-255 (2015)

[14] Nguyen L M, Liu J, Scheinberg K, et al. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In: International Conference on Machine Learning. PMLR, pp. 2613-2621 (2017)

[15] Nguyen L M, Liu J, Scheinberg K, et al. Stochastic recursive gradient algorithm for nonconvex optimization. 2017, arXiv:1705.07261.

[16] Fang C, Li C J, Lin Z, et al. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In: Advances in Neural Information Processing Systems, 31 (2018)

[17] Bottou L. Online learning and stochastic approximations. On-line learning in neural networks. 17(9), 142 (1998)

[18] Barzilai J, Borwein J M. Two-point step size gradient methods. IMA journal of numerical analysis. 8(1), 141-148 (1988)

[19] Sopyla K, Drozda P. Stochastic gradient descent with Barzilai-Borwein update step for SVM. Information Sciences. 316, 218-233 (2015)

[20] Tan C, Ma S, Dai Y H, et al. Barzilai-borwein step size for stochastic gradient descent. In: Advances in Neural information processing systems, 2016, 29.

[21] Li B, Giannakis G B. Adaptive step sizes in variance reduction via regularization. 2019, arXiv:1910.06532.

[22] Liu Y, Wang X, Guo T. A linearly convergent stochastic recursive gradient method for convex optimization. Optimization Letters. 14(8), 2265-2283 (2020)

[23] Yang Z, Wang C, Zang Y, et al. Mini-batch algorithms with Barzilai–Borwein update step. Neurocomputing. 314, 177-185 (2018)

[24] Yang Z, Wang C, Zhang Z, et al. Accelerated stochastic gradient descent with step size selection rules. Signal Processing. 159, 171-186 (2019)

[25] Yang Z, Chen Z, Wang C. Accelerating mini-batch sarah by step size rules. Information Sciences. 558, 157-173 (2021)

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

[27] Zhao P, Zhang T. Stochastic optimization with importance sampling for regularized loss minimization. In: International Conference on Machine Learning, pp.1-9 (2015)

[28] Shen Z, Qian H, Zhou T, Mu T. Adaptive variance reducing for stochastic gradient descent. In: IJCAI, pp. 1990–1996 (2016)

[29] Dai Y H, Huang Y, Liu X W. A family of spectral gradient methods for optimization. Computational Optimization and Applications. 74(1), 43-65 (2019)

[30] Fletcher R. On the barzilai-borwein method. Optimization and control with applications. Springer, Boston, MA, 235-256 (2005)

[31] Zhou B, Gao L, Dai Y H. Gradient methods with adaptive step-sizes. Computational Optimization and Applications. 35(1), 69-86 (2006)

[32] Dai Y H, Fletcher R. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. Numerische Mathematik. 100(1), 21-47 (2005)

[33] Huang Y, Dai Y H, Liu X W, et al. On the acceleration of the Barzilai-Borwein method. Computational Optimization and Applications. 81(3), 717-740 (2022)

[34] Nesterov, Y. Introductory Lectures on Convex Optimization: A Basic Course, Springer, Boston 2014.

[35] Shao G M, Xue W, Yu G H, et al. Improved SVRG for finite sum structure optimization with application to binary classification. Journal of Industrial and Management Optimization. 16(5), 2253-2266 (2020)

[36] Polyak B T. Gradient methods for the minimisation of functionals. USSR Computational Mathematics and Mathematical Physics. 3(4), 864-878 (1963)