Expert Systems With Applications 238 (2024) 122295

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

SARAH-M: A fast stochastic recursive gradient descent algorithm via momentum

Zhuang Yang

School of Computer Science and Technology, Soochow University, Suzhou, 215006, China

ARTICLE INFO

Keywords:
Machine learning
Variance reduction
Momentum
Adaptive step size
Stochastic optimization

ABSTRACT

As a simple but effective way, the momentum method has been widely adopted in stochastic optimization algorithms for large-scale machine learning problems and the success of stochastic optimization with the momentum term for many applications in machine learning and other related areas has been reported everywhere. However, the understanding of how the momentum improves the performance of modern variance reduced stochastic gradient algorithms, e.g., the stochastic dual coordinate ascent average gradient (SDCA) method, the stochastically controlled stochastic gradient (SCSG) method, the stochastic recursive gradient algorithm (SARAH), etc., is still limited. To tackle this issue, this work studies the performance of SARAH with the momentum term theoretically and empirically, and develops a novel variance reduced stochastic gradient algorithm, termed as SARAH-M. We rigorously prove that SARAH-M attains a linear rate of convergence for minimizing the strongly convex function. We further propose an adaptive SARAH-M method (abbreviated as AdaSARAH-M) by incorporating the random Barzilai–Borwein (RBB) technique into SARAH-M, which provides an easy way to determine the step size for the original SARAH-M algorithm. The theoretical analysis that shows AdaSARAH-M with a linear convergence speed is also provided. Moreover, we show that the complexity of the proposed algorithms can outperform modern stochastic optimization algorithms. Finally, the numerical results, compared with state-of-the-art algorithms on benchmarking machine learning problems, verify the efficacy of the momentum in variance reduced stochastic gradient algorithms.

1. Introduction

The great success of stochastic optimization algorithms in large-scale machine learning, computer vision and neuroscience makes them gain immense popularity in recent years (Amari, 2013; Lan, 2020; Mu et al., 2016; Xie et al., 2021; Xin et al., 2020), where the applications of these problems can be formulated as the minimization of the stochastic composite optimization problem, i.e.,

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

where $w \in \mathbb{R}^d$ stands for the parameters of a model, $f_i : \mathbb{R}^d \to \mathbb{R}$ denotes the empirical loss corresponding to the ith data point and n characters the number of data sets. In this work, we are interested in that the empirical loss $F(w)$ and each empirical loss are Lipschitz smooth and strongly convex.

Compared to the deterministic optimization algorithm which faces greatly challenging in dealing with extremely large data sets (i.e., $n \to \infty$), the typical stochastic optimization algorithm usually worked with a single sample, which immensely reduced the computational burden (Duchi et al., 2012; Spall, 2012). The embarrassing, however, is that the high variance generating by the sampling strategy makes the stochastic optimization algorithm often converge slowly and contribute to the performance of stochastic optimization algorithms sensitive to some crucial parameters.

To date, there exists a great deal of literature on modifications of the conventional stochastic optimization algorithms to promote their empirical and theoretical performance. Here, we summarize several classical instances as follows:

Momentum: The commonly used momentum method in stochastic optimization algorithms includes the Nesterov's accelerated gradient (NAG) (Nesterov, 2004), heavy-ball momentum (HBM) (Polyak, 1987), and quasi-hyperbolic momentum (QHM) (Ma & Yarats, 2018). This work mainly focuses on the performance of HBM in stochastic gradient descent (SGD) optimization algorithms, where the iteration of the form is simply written as

$$w_{k+1} = w_k - \lambda_k \nabla f_i(w_k) + \beta_k(w_k - w_{k-1}),$$

where λ_k and β_k are two positive sequences (Sebbouh et al., 2021). The term, $w_k - w_{k-1}$, is often called the momentum term.

E-mail address: zhuangyng@163.com.

https://doi.org/10.1016/j.eswa.2023.122295
Received 23 May 2022; Received in revised form 19 October 2023; Accepted 20 October 2023
Available online 31 October 2023
0957-4174/© 2023 Elsevier Ltd. All rights reserved.

Gradient Averaging: Closely linked to the momentum method is taking the sample average of all past gradients in stochastic optimization algorithms (Ruszczynski & Syski, 1983), where the iteration scheme uses the iterative form of

$$w_{k+1} = w_k - \frac{\lambda_k}{k} \sum_{i=1}^{k} \nabla f_i(w_k).$$

Such average scheme makes the algorithm converge with a constant step size and improves the constant in the convergence speed (Mou et al., 2020).

Iterate Averaging: Instead of averaging the gradients, some works treat the SGD as a basic iteration and use an average over the sequence $\{w_k\}$ (Neu & Rosasco, 2018). Such iterative procedure makes SGD achieve the same performance as Newton-type SGD methods when taking an appropriate step size sequence (Polyak & Juditsky, 1992).

Mini-batching: Mini-batching adopts a set of random samples to address the noise issue of SGD, which effectively reduce the variance of the stochastic estimate (Li et al., 2014). Certainly, SGD with mini-batching technique often requires more computational cost, which, in practice, is undesirable.

Importance Sampling: Importance sampling runs according to a gingerly data driven of the probabilities of choosing instances in the iterative process, contributing to a reduction of the high variance of stochastic optimization algorithms thus chosen (Csiba & Richtárik, 2018). However, the importance sampling often employs importance values denoted by the complete gradient information which alter during optimization and obtain favorable theoretical properties, but may compute prohibitively.

Diminishing Step Sizes: Diminishing step sizes deal with the noisy issue by a gradual and direct scaling down process, which make sure the convergence of SGD (Nguyen et al., 2019). Obviously, such strategy slows down the convergence rate of stochastic optimization algorithms. We summarize that while conventional methods manage to reduce the variance in the stochastic optimization algorithms, this does not come for free.

Additionally, apart from the above mentioned techniques, several variance reduced approaches have been proposed to reduce the variance of stochastic optimization algorithms by constructing a more sophisticated and accurate gradient estimator such as ADAM (according to adaptive estimates of lower order moments) (Kingma & Ba, 2015), the stochastic variance reduced gradient (SVRG) method (Johnson & Zhang, 2013), the stochastically controlled stochastic gradient (SCSG) method (Lei & Jordan, 2017), the stochastic average gradient (SAG) method (Roux et al., 2012a), SAGA (Defazio et al., 2014), the stochastic dual coordinate ascent average gradient (SDCA) method (Shalev-Shwartz & Zhang, 2013), the stochastic recursive gradient algorithm (SARAH) (Nguyen et al., 2017a), the stochastic path-integrated differential estimator (SPIDER) method (Fang et al., 2018), etc. To obtain the step size readily, Roux et al. (2012a) considered using the line search to reckon the step size for SAG, leading to SAG-LS. Similarly, Yang et al. (2020) applied the hypergradient descent (HD) technique to compute the step size into SARAH under the mini-batch setting, generating MB-SARAH-HD. Note that, in contrast to SAG and SAGA, SARAH does not have to store past gradients by adopting a simple recursive framework for renewing stochastic gradient estimates. Additionally, compared with the SVRG-like algorithm, SARAH is regarded as a type of biased stochastic optimization algorithms. Specifically, most existing variance reduced techniques control the variance by using the historical information in different ways.

What above mentioned variance-reduced stochastic gradient algorithms and the momentum method have in common is that they both use the past information. Several researches try to study the effect of the momentum in these variance-reduced stochastic gradient algorithms. For example, by designing an easy and effective momentum acceleration trick, Shang et al. (2018) proposed an accelerated proximal stochastic variance reduced gradient (ASVRG) approach. Zhou et al. (2019) developed a directly accelerated variant of SAGA employing a new sampled negative momentum, which had the best known oracle complexity for strongly convex cases. Wang et al. (2019) proposed a new momentum scheme to speed up SpiderBoost for composite optimization and proved that the proposed algorithm obtained the near-optimal oracle complexity in theory. Utilizing the ideas of NAG and variance reduced technique of SVRG in the mini-batch setting, Nitanda (2014) proposed and analyzed a novel method named as the accelerated mini-bacth prox-SVRG (Acc-Prox-SVRG) method.

To understand how the momentum affects the variance reduced stochastic gradient algorithm deeply, we develop and analyze a class of variance-reduced stochastic gradient algorithms with the momentum term. Furthermore, we equip the stochastic optimization algorithms with momentum with the capacity to acquire an online learning step size automatically, which is challenging and intractable in practical applications. Specifically, we summarize the main results and insights derived in the remainder of this work as follows:

(1) We provide an analysis of SARAH with the momentum term and develop a novel variance reduced stochastic gradient algorithm, termed as SARAH-M. We prove the convergence of SARAH-M when the loss function is strongly convex and show that it attains a linear convergence rate.

(2) Besides, we propose an adaptive SARAH-M method, referred to as AdaSARAH-M, by incorporating the random Barzilai–Borwein (RBB) technique into SARAH-M, which provides a simple way to determine the step size for the original SARAH-M algorithm. The theoretical analysis that shows AdaSARAH-M with a linear convergence speed is also supplied.

(3) Further, we rigorously analyze the complexity of the proposed algorithms, SARAH-M and AdaSARAH-M. Specifically, we show that under an appropriate condition, the proposed algorithms can outperform state-of-the-art stochastic optimization algorithms.

(4) Finally, a range of numerical results, compared with state-of-the-art algorithms on benchmarking machine learning problems verify the effectiveness of the proposed algorithms.

More recently, we have found that an increasing number of studies confirm the effectiveness of different types of momentum in improving stochastic optimization and various stochastic optimization algorithms with momentum have been proposed and analyzed. This work particularly focuses on the mechanism of momentum in enhancing modern variance reduced stochastic optimization algorithms. More specifically, we explore the theoretical properties and numerical features of a representative type of biased stochastic optimization algorithms (a.k.a. SARAH) with momentum. Additionally, the study of selecting the step size for stochastic optimization with momentum is still quite limited. Motivated by this gap, this work further equips stochastic optimization algorithms with momentum with the ability to figure out an online step size automatically, which greatly reduces the difficulty in selecting the step size, a very important hyperparameter, in practice. Importantly, the characteristics of theory and experiments are also provided. It is worth pointing out that although we analyze the theoretical results of the resulting algorithms for the case that the objective function only keeps the strongly convex assumption. However, we can apply the resulting algorithms into non-convex optimization problems easily and fluently.

The rest of this paper is arranged as follows: Section 2 reviews several related works. Section 3 introduces some preliminaries. Section 4 introduces our first algorithm, SARAH-M, and presents its convergence analysis and computational complexity. Section 5 depicts our second algorithm, AdaSARAH-M, and shows the theoretical results for AdaSARAH-M. Section 6 contains numerical results for our proposed algorithms applied to benchmarking machine learning problems. Section 7 concludes the paper.

Notations: For convenience, we will introduce a few notations used throughout. The symbol $\|w\|$ is used to denote the Euclidean norm of a vector w. The symbol $[n]$ denote the set $\{1, 2, \ldots, n\}$. We use $\nabla F(\cdot)$ to stand for the gradient of $F(\cdot)$. Let $\mathbb{R}^d$ denote a set of d-dimension vectors. We use the symbol $\langle \cdot, \cdot \rangle$ to denote the inner product in the Euclidean space. In addition, we adopt the symbol $f = O(g)$ to denote $f \leq Cg$ for some positive constant $C(>0)$. We denote the expectation of a random variable φ by $\mathbb{E}[\varphi]$.

2. Related work

In some cases, the momentum methods, e.g., NAG, can be seen as a generalization of the Anderson acceleration (AA) method, maintaining m recent iterates of an optimization approach, where its convergence properties have been discussed in different background, both deterministic (Higham & Strabić, 2016; Toth & Kelley, 2015) and stochastic (Scieur et al., 2020; Toth et al., 2017). In more recent work, by incorporating damped projection and adaptive regularization to classical Anderson mixing (AM), Wei et al. (2021) proposed a stochastic Anderson Mixing (SAM) scheme to address non-convex stochastic optimization problems. Under mild assumptions, the authors provided the convergence theory of SAM.

Another closely related to the momentum method is extrapolation methods that used the last few iterates of an optimization approach to generate a better estimator of the optimal solution (Sidi, 2003). Scieur et al. (2017) studied the extrapolation approach in a stochastic case, where the iterates were derived by either a simple or an accelerated stochastic gradient approach. Xu et al. (2019) analyzed gradient and stochastic gradient descent approach with extrapolation for non-convex minimization. Gidel et al. (2018) developed a stochastic extragradient method for dealing with the min–max saddle point problem from the view of variational inequality.

Recently, Liu et al. (2019) designed a new and simple momentum to speed up the conventional SAGA algorithm, and developed a direct accelerated incremental gradient descent approach. Gitman et al. (2019) deepened the understanding of the role of momentum in general stochastic gradient approaches. Loizou and Richtárik (2020) studied several kinds of stochastic optimization algorithms (a.k.a. SGD, stochastic Newton, stochastic proximal point and stochastic dual subspace ascent) enriched with HBM. Tran et al. (2021) combined two modern ideas widely adopted in optimization for machine learning: shuffling technique and momentum strategy to develop a new shuffling gradient based approach with momentum for non-convex finite sum optimization problems. To solve the issue that counterexamples existed and prevented NAG from offering similar acceleration in the stochastic optimization background, Allen-Zhu (2017) developed and analyzed Katyusha. Further, to reduce the probability in computing the exact gradient, Kovalev et al. (2020) designed loopless variant of Katyusha, named L-Katyusha.

More recently, to solve the accumulating errors of NAG in stochastic gradient descent (SGD) algorithms, Wang et al. (2022) developed a new NAG-type algorithm, scheduled restart SGD (SRSGD). Hou et al. (2022) put forward a distributed stochastic Frank-Wolfe solver via judiciously combining NAG and gradient tracking techniques for stochastic convex and non-convex optimization over networks. By generalizing Storm (Cutkosky & Orabona, 2019) from smooth cases to non-smooth cases, Xu and Xu (2023) designed a momentum-biased variance-reduced mirror-prox SGD method for tackling non-convex non-smooth stochastic problems.

3. Preliminaries

Throughout this paper, the presented results are derived in this paper based on the following assumptions.

Assumption 1. Each loss function, $f_i(w)$, is differentiable with

$$\|\nabla f_i(w) - \nabla f_i(v)\| \leq L\|w - v\|. \tag{2}$$

for some $0 < L < \infty$ and all $i \in \{1, 2, \ldots, n\}$.

Assumption 1 implies that the loss function $F(w)$ also satisfies $\|\nabla F(w) - \nabla F(v)\| \leq L\|w - v\|$. In addition, as a direct result of Assumption 1, for any $w, v \in \mathbb{R}^d$, we have the following crucial inequality:

$$F(w) \leq F(v) + \langle \nabla F(v), w - v \rangle + \frac{L}{2}\|w - v\|^2. \tag{3}$$

Assumption 2. $F(w)$ satisfies the μ-strongly convex condition, i.e., there exists $\mu > 0$ such that for any w, $v \in \mathbb{R}^d$,

$$\langle \nabla F(w) - \nabla F(v), w - v \rangle \geq \mu\|w - v\|^2, \tag{4}$$

or equivalently

$$F(w) \geq F(v) + \langle \nabla F(v), w - v \rangle + \frac{\mu}{2}\|w - v\|^2. \tag{5}$$

Utilizing the strong convexity of the loss function $F(w)$, it holds the following fact

$$2\mu[F(w) - F(w_*)] \leq \|\nabla F(w)\|^2, for\ any\ w \in \mathbb{R}^d \tag{6}$$

where we define $w_* = \arg\min_w F(w)$. For more details, we refer interested readers to Bottou et al. (2018).

Further, we introduce the following definition that will be used in the analysis of the complexity.

Definition 1. We say a stochastic gradient optimization method has an ε-accuracy solution in k iterations when the condition $\mathbb{E}[\|\nabla F(w_k)\|^2] \leq \varepsilon$ holds.

4. SARAH with momentum

This section starts with the introduction of our first algorithm, termed as SARAH-M, in Section 4.1. Then, we discuss the convergence property of the proposed algorithm in Section 4.2.

4.1. SARAH-M

Since our SARAH-M algorithm has a close link with SARAH and momentum, we will first bring into the details of SARAH in Algorithm 1 and provide more detailed information about the momentum technique later, which has been briefly described in the above section.

Algorithm 1 SARAH

```
Inut: the step size $\eta > 0$, the initial point $\widetilde{w}_0$ and the initial loop size $m$.
for $s = 1$ to $S$ do
    $w_0 = \widetilde{w}_{s-1}$
    $v_0 = \nabla F(w_0)$
    $w_1 = w_0 - \eta_0 v_0$
    for $k = 1$ to $m - 1$ do
        Randomly choose $i \in [n]$ and update weight,
        $v_k = \nabla f_i(w_k) - \nabla f_i(w_{k-1}) + v_{k-1}$        (7)
        $w_{k+1} = w_k - \eta v_k$
    end for
    $\widetilde{w}_s = w_k$ with $k$ is picked up uniformly at random from $\{0, 1, \ldots, m\}$.
end for
```

Generally, for tackling Problem (1), SGD with momentum employs the iterations of the form

$$w_{k+1} = w_k - \eta_k \nabla f_i(w_k) + \beta_k(w_k - w_{k-1}), \tag{8}$$

where $\eta_k > 0$ is the step size and $\beta_k \in [0,1]$ denotes the momentum coefficient. In practice, we often set all $\beta_k = \beta$. Without loss of generality, in our theoretical analysis, we set $\beta_k \le \hat{\beta}$, where $\hat{\beta} \in [0,1]$.

Note that this paper considers SARAH in the mini-batch setting, i.e., the stochastic gradient estimate v_k is updated using the following iterative scheme:

$$\hat{v}_k = \nabla F_S(w_k) - \nabla F_S(w_{k-1}) + \hat{v}_{k-1},$$

where $\nabla F_S(w_k) = \frac{1}{b}\sum_{i\in S}\nabla f_i(w_k)$, $\nabla F_S(w_{k-1}) = \frac{1}{b}\sum_{i\in S}\nabla f_i(w_{k-1})$ and $S \subseteq [n]$ with size $|S| = b$. Actually, our proposed algorithms also achieve better performance when setting $b = 1$.

Followed, we will present our first algorithm, SARAH-M, where the details of SARAH-M are provided in Algorithm 2.

Algorithm 2 SARAH-M

```
Input: update frequency $m$; mini-batch size $b \in [n]$; initial point $\widetilde{w}_0$;
learning rate $\eta$; non-negative sequence $\{\beta_k\}$
for $s = 1, 2, \ldots$ do
    $v_0 = \widetilde{w}_{s-1}$
    $G_0 = \nabla F(v_0)$
    $v_1 = w_1 = v_0 - \eta G_0$
    for $k = 1, \ldots, m$ do
        Randomly choose subset $S \subseteq [n]$ of size $b$ and compute a new
        stochastic estimate of $\nabla F(w_k)$;
        $G_k = \nabla F_S(v_k) - \nabla F_S(v_{k-1}) + G_{k-1}$
        $w_{k+1} = v_k - \eta G_k$
        $v_{k+1} = w_{k+1} + \beta_k(w_{k+1} - w_k)$
    end for
    Set $\widetilde{w}_{s+1} = w_{m+1}$
end for
```

Remark. Different with the original SARAH algorithm (appearing in Algorithm 1) that sets $\tilde{w}_s = w_k$, where k is picked up uniformly at random from $\{0, 1, \ldots, m\}$, we set $\tilde{w}_s$ to be the last iteration w_{m+1}. Practically, for the algorithms with inner and outer loops, it makes sense to keep the last computed $\tilde{w}_s$ if multiple outer loop iterations were used, where this is indeed the case and widely employed by many existing variance-reduced optimization algorithms, see, Nguyen et al. (2022), Xiao and Zhang (2014) and Yasuda et al. (2019). This makes the algorithm do not generate the last result with respect to the probability distribution and greatly simplifies the computation.

4.2. Convergence analysis for SARAH-M

This subsection begins with the following lemmas that used in our theoretical analysis.

Lemma 1. *Under Assumption 1, SARAH-M (Algorithm 2) with a single outer loop iteration satisfies*

$$\begin{aligned}&\mathbb{E}\left[\|\nabla F(w_k)\|^2\right]\\ &\le \frac{2}{\eta-\hat{\beta}}\mathbb{E}[F(w_0) - F(w_{m+1})] + \frac{\eta}{\eta-\hat{\beta}}\sum_{k=0}^{m}\mathbb{E}\left[\|\ \nabla F(w_k)\right.\\ &\left.- G_k\ \|^2\right] - \frac{2}{\eta-\hat{\beta}}\left(\frac{\eta}{2} - L\eta^2\right)\sum_{k=0}^{m}\mathbb{E}\left[\|G_k\|^2\right]\end{aligned}$$

Proof. See Appendix A.1. □

From lemma 3 in Nguyen et al. (2017b), the upper bound of $\mathbb{E}[\|\nabla F(v_k) - G_k\|^2]$ are deduced:

Lemma 2. *Under Assumption 1, consider G_k denoted in SARAH-M (Algorithm 2), then for any $k \ge 1$, we deduce*

$$\mathbb{E}\left[\|\nabla F(v_k) - G_k\|^2\right] \le \frac{1}{b}\left(\frac{n-b}{n-1}\right)L^2\eta^2\sum_{j=1}^{k}\mathbb{E}[\|G_{j-1}\|^2]. \tag{9}$$

Theorem 1. *Under Assumptions 1, 2, Lemmas 1, and 2, let $w^* = \arg\min_w F(w)$ and pick up $S \subset \{1, \ldots, n\}$ with mini-batch samples b. Consider SARAH-M (Algorithm 2) with a single outer loop iteration, when the following condition is satisfied*

$$\frac{2}{b}\left(\frac{n-b}{n-1}\right)L^2\eta^2 m - \left(\frac{\eta}{2} - L\eta^2\right) \le 0, \tag{10}$$

then it holds that

$$\mathbb{E}[\|\nabla F(w_m)\|^2] \le \frac{2}{(m+1)(\eta-\hat{\beta})}\mathbb{E}[F(w_0) - F(w_*)].$$

Proof. See Appendix A.2. □

Theorem 1 indicates that SARAH-M (Algorithm 2) with a single outer loop iteration has a sub-linear convergence speed. In particular, to hold

$$\frac{2}{(m+1)(\eta-\hat{\beta})}\mathbb{E}[F(w_0) - F(w_*)] \le \epsilon,$$

it is appropriate to set $m = O\big(\frac{1}{(\eta-\hat{\beta})\epsilon}\big)$. Therefore, we conclude that the gradient complexity of SARAH-M (Algorithm 2) to attain ϵ-accurate solution is $n + 2bm = O\big(n + \frac{b}{(\eta-\hat{\beta})\epsilon}\big)$.

For simplicity and clarity, we recap the gradient complexity of SARAH-M (Algorithm 2) with one outer loop iteration in the following corollary.

Corollary 1. *Under Assumptions 1, 2, Lemmas 1, and 2, consider SARAH-M (Algorithm 2) with a single outer iteration, then $\|\nabla F(w_s)\|^2$ attains a sub-linear convergence with rate $O\big(\frac{1}{m(\eta-\hat{\beta})}\big)$, and its gradient complexity is $O\big(n + \frac{b}{(\eta-\hat{\beta})\epsilon}\big)$.*

Further, we easily obtain the following theoretical results of SARAH-M (Algorithm 2) with multiple outer loop iterations:

Theorem 2. *Under Assumptions 1, 2 and Lemmas 1, 2, let $w_* = \arg\min_w F(w)$ and pick up $S \subset \{1, \ldots, n\}$ with mini-batch samples b. Consider SARAH-M (Algorithm 2) with*

$$\frac{2}{b}\left(\frac{n-b}{n-1}\right)L^2\eta^2 m - \left(\frac{\eta}{2} - L\eta^2\right) \le 0,$$

then, we deduce

$$\mathbb{E}[\|\nabla F(\widetilde{w}_s)\|^2] \le \rho^s\|\nabla F(\widetilde{w}_0)\|^2,$$

where the parameter ρ is set to $\rho = \frac{1}{\mu(m+1)(\eta-\hat{\beta})}$.

Proof. See Appendix A.3. □

From Theorem 2, we have that the number of outer loop iterations can be set to $s = O\big(\log\big(\frac{1}{\epsilon}\big)\big)$, leading to the condition $\mathbb{E}[\|\nabla F(\widetilde{w}_s)\|^2] \le \rho^s\|\nabla F(\widetilde{w}_0)\|^2 \le \epsilon$ hold. Therefore, we can recap the total gradient complexity of SARAH-M (Algorithm 2) in the following corollary.

Corollary 2. *Under Assumptions 1, 2, Lemmas 1, and 2, the total gradient complexity of SARAH-M (Algorithm 2) with multiple outer loop iterations to achieve an ϵ-accurate solution is $O\left(\left(n + \frac{b}{(\eta-\hat{\beta})\epsilon}\right)\log\left(\frac{1}{\epsilon}\right)\right)$.*

5. Adaptive SARAH-M

This section provides an automatic way to figure out the step size for SARAH-M (Algorithm 2) by incorporating the RBB method into SARAH-M, leading to a new algorithm, referred to as AdaSARAH-M. We arrange this section by providing our second algorithm in Section 5.1 and convergence analysis in Section 5.2.

5.1. AdaSARAH-M

In this section, we start of the description of the RBB method, followed by the detail of our second algorithm in Algorithm 3.

Yang et al. (2021) proposed the RBB method and applied it into SGD-like algorithm (a.k.a. SARAH), where the key iteration step for solving Problem (1) is

$$w_{k+1} = w_k - \eta_k \nabla F_S(w_k),$$

$$\eta_k = \frac{\gamma}{b_H} \cdot \frac{\|w_k - w_{k-1}\|^2}{\langle w_k - w_{k-1}, \nabla F_{S_H}(w_k) - \nabla F_{S_H}(w_{k-1})\rangle},$$

where $\nabla F_{S_H}(w_k) = \frac{1}{b_H}\sum_{i\in S_H}\nabla f_i(w_k)$, $\nabla F_{S_H}(w_{k-1}) = \frac{1}{b_H}\sum_{i\in S_H}\nabla f_i(w_{k-1})$ and $S_H \subseteq [n]$ with size $|S_H| = b_H$.

Next, we begin to describe our AdaSARAH-M algorithm in the following.

Algorithm 3 AdaSARAH-M

```
Input: update frequency $m$; mini-batch sizes $b, b_H \in [n]$; initial point $\widetilde{w}_0$; initial learning rate $\eta_0$; non-negative sequence $\{\beta_k\}$
for $s = 1, 2, \ldots$ do
    $v_0 = \widetilde{w}_s$
    $G_0 = \nabla F(v_0)$
    $v_1 = w_1 = v_0 - \eta_0 G_0$
    for $k = 1, \ldots, m$ do
        Randomly pick up mini-batch $S \subseteq [n]$ of size $b$ and compute a stochastic estimate of $\nabla F(w_k)$;
        $G_k = \nabla F_S(v_k) - \nabla F_S(v_{k-1}) + G_{k-1}$
        Randomly choose mini-batch $S_H \subseteq [n]$ of size $b_H$ and compute the step size
        $\eta_k = \frac{\gamma}{b_H} \cdot \frac{\|v_k - v_{k-1}\|^2}{\langle v_k - v_{k-1}, \nabla F_{S_H}(v_k) - \nabla F_{S_H}(v_{k-1})\rangle}$
        $w_{k+1} = v_k - \eta_k G_k$
        $v_{k+1} = w_{k+1} + \beta_k(w_{k+1} - w_k)$
    end for
    Set $\widetilde{w}_{s+1} = w_{m+1}$
end for
```

5.2. Convergence analysis for AdaSARAH-M

To complete the proof of AdaSARAH-M (Algorithm 3), we also need the following lemma, showing the bound of the step size.

Lemma 3. *Under Assumptions 1 and 2, the step size η_k used in AdaSARAH-M (Algorithm 3) is belonging to $\left[\frac{\gamma}{Lb_H}, \frac{\gamma}{\mu b_H}\right]$.*

Proof. See Appendix B.1. □

Additionally, we need the following lemma, providing the upper bound of $\mathbb{E}[\|\nabla F(w_k)\|^2]$.

Lemma 4. *Under Assumptions 1, 2, and Lemma 3, AdaSARAH-M (Algorithm 3) with a single outer loop iteration satisfies*

$$\mathbb{E}\left[\|\nabla F(w_k)\|^2\right] \le \frac{2\mu b_H}{\gamma - \hat{\beta}\mu b_H}\mathbb{E}[F(w_0) - F(w_{m+1})]$$

$$+ \frac{\gamma}{\gamma - \hat{\beta}\mu b_H}\sum_{k=0}^{m}\mathbb{E}\left[\|\nabla F(w_k) - G_k\|^2\right] - \frac{\gamma}{\gamma - \hat{\beta}\mu b_H}$$

$$\cdot \left(1 - \frac{2L\gamma^2}{\mu b_H}\right)\sum_{k=0}^{m}\mathbb{E}\left[\|G_k\|^2\right]$$

Proof. See Appendix B.2. □

Similar to SARAH-M (Algorithm 2), we first establish the theoretical results of AdaSARAH-M (Algorithm 3) with one outer loop iteration in the following theorem.

Theorem 3. *Under Assumptions 1, 2 and Lemmas 1, 2, 3, 4, let $w^* = \arg\min_w F(w)$ and choose $S, S_H \subset \{1, \ldots, n\}$ with mini-batch samples b and b_H, respectively. Consider AdaSARAH-M (Algorithm 3) with a single outer loop iteration, when the following condition is satisfied*

$$\frac{2L^2 b_H^2}{b\mu^2 b_H^2}\left(\frac{n-b}{n-1}\right)m - \left(1 - \frac{2L\gamma^2}{\mu b_H}\right) \le 0, \tag{11}$$

then we deduce

$$\mathbb{E}[\|\nabla F(w_m)\|^2] \le \frac{2\mu b_H}{(m+1)(\gamma - \hat{\beta}\mu b_H)}\mathbb{E}[F(w_0) - F(w_*)].$$

Proof. See Appendix B.3. □

Next, we will provide the theoretical results of AdaSARAH-M (Algorithm 3) with multiple outer loop iterations in Theorem 4.

Theorem 4. *Under Assumptions 1, 2 and Lemmas 1, 2, 3, 4, let $w^* = \arg\min_w F(w)$ and choose $S, S_H \subset \{1, \ldots, n\}$ with mini-batch samples b and b_H, respectively. Consider AdaSARAH-M (Algorithm 3) with multiple outer loop iterations, when the condition*

$$\frac{2L^2 b_H^2}{b\mu^2 b_H^2}\left(\frac{n-b}{n-1}\right)m - \left(1 - \frac{2L\gamma^2}{\mu b_H}\right) \le 0, \tag{12}$$

is satisfied, then we have

$$\mathbb{E}[\|\nabla F(w_m)\|^2] \le \tau^s \mathbb{E}[F(w_0) - F(w_*)],$$

where the parameter τ is set to $\tau = \frac{b_H}{(m+1)(\gamma - \hat{\beta}\mu b_H)}$.

According to the proof of Theorem 2, the results of Theorem 4 can be easily obtained.

Similar to the analysis of SARAH-M (Algorithm 2), we provide the gradient complexity of AdaSARAH-M (Algorithm 3) with a single or multiple outer loop iterations in the following corollary.

Corollary 3. *Under Assumptions 1, 2 and Lemmas 1, 2, 3, 4, consider AdaSARAH-M (Algorithm 3) with a single outer loop iteration, then $\|\nabla F(\hat{w}_s)\|^2$ attains a sub-linear convergence with rate $O\left(\frac{\mu b_H}{m(\gamma - \hat{\beta}\mu b_H)}\right)$, and its gradient complexity attains $O\left(n + \frac{\mu b_H(b+b_H)}{(\gamma - \hat{\beta}\mu b_H)\varepsilon}\right)$. Moreover, the total complexity of AdaSARAH-M (Algorithm 3) with multiple outer loop iterations attains $O\left(\left(n + \frac{\mu b_H(b+b_H)}{(\gamma - \hat{\beta}\mu b_H)\varepsilon}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$.*

In order to comprehend the computational complexity of SARAH-M (Algorithm 2) and AdaSARAH-M (Algorithm 3) clearly, we offer the overall complexity of several modern SGD-type algorithms in Table 1.

Note that in Table 1, the mark κ represented the condition number, where it was often taken as $\kappa = \frac{L}{\mu}$. For L-Katyusha, p denoted the probability for determining when computing the exact gradient. For MB-SARAH-HD, α denoted the hypergradient learning rate and M was the upper boundary of $\|\nabla F(w)\|$. Actually, from Corollary 2 and Corollary 3, we can conclude that under the appropriate conditions, the gradient complexity of the proposed algorithms (SARAH-M and AdaSARAH-M) matches or even outperforms that of modern stochastic optimization algorithms. For instance, when setting $b \le \kappa\varepsilon(\eta - \hat{\beta})$, the gradient complexity of SARAH-M (Algorithm 2) outperforms that of SAG, SVRG, SARAH, etc.

Table 1
Comparison of the total complexity suitable for solving Problem (1).

Algorithm	Complexity
SAG (Roux et al., 2012b)	$O\left((n+\kappa)\log\left(\frac{1}{\varepsilon}\right)\right)$
SAGA (Defazio et al., 2014)	$O\left((n+\kappa)\log\left(\frac{1}{\varepsilon}\right)\right)$
SDCA (Shalev-Shwartz & Zhang, 2013)	$O\left((n+\kappa)\log\left(\frac{1}{\varepsilon}\right)\right)$
SARAH (Nguyen et al., 2017a)	$O\left((n+\kappa)\log\left(\frac{1}{\varepsilon}\right)\right)$
Katyusha (Allen-Zhu, 2017)	$O\left(\left(n+\sqrt{n\kappa}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$
SVRG/Prox-SVRG (Johnson & Zhang, 2013; Xiao & Zhang, 2014)	$O\left((n+\kappa)\log\left(\frac{1}{\varepsilon}\right)\right)$
L-Katyusha (Kovalev et al., 2020)	$O\left(\left((1+pn)\sqrt{\kappa/p}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$
SVRG/Prox-SVRG (Johnson & Zhang, 2013; Xiao & Zhang, 2014)	$O\left((n+\kappa)\log\left(\frac{1}{\varepsilon}\right)\right)$
ASVRG (Shang et al., 2018)	$O\left(\left(n+\sqrt{n\kappa}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$
Acc-Prox-SVRG (Nitanda, 2014)	$O\left(n+\min\{\kappa, n\sqrt{\kappa}\}\log\left(\frac{1}{\varepsilon}\right)\right)$
MB-SARAH-HD (Yang et al., 2020)	$O\left(\left(n+\frac{1}{M\sqrt{\alpha\varepsilon}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$
SARAH-M (Algorithm 2)	$O\left(\left(n+\frac{b}{(\eta-\hat{\beta})\varepsilon}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$
AdaSARAH-M (Algorithm 3)	$O\left(\left(n+\frac{\mu b_H(b+b_H)}{(\gamma-\hat{\beta}\mu b_H)\varepsilon}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$

Table 2
Data sets information used in the experiments.

Data sets	Sizes	Feature	λ
a8a	22,696	123	10^{-2}
ijcnn1	49,990	22	10^{-2}
MNIST	60,000	784	10^{-2}
w8a	49,749	300	10^{-2}
news20.binary	19,996	1,355,191	10^{-2}

6. Numerical results

To show the performance of the proposed algorithms, here, the experiments performed on ℓ_2-regularized logistic regression problem are provided, where the loss function is formulated as

$$\min_{w\in\mathbb{R}^d} F(w) := \frac{1}{n}\sum_{i=1}^{n}\log\left(1+\exp\left(-y_i x_i^T w\right)\right) + \frac{\lambda}{2}\|w\|^2, \tag{13}$$

where $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \{-1, +1\}^n$ denotes a set of data points and the last term in (13) often is regarded as the regularizer term. All tests were run on an Intel (R) Core (TM) i7-10750H CPU @ 2.60 GHz, 2.59 GHz and 32 GB RAM performing MATLAB R2019a.

Besides, in our experiments, the data sets, $a8a$, $ijcnn1$, $MNIST$ and $w8a$ are used, where these data sets can be got from LIBSVM (Chang & Lin, 2011). Particularly, we listed the details of these data sets information in Table 2,[1] including the training size and the feature size.

6.1. Properties of SARAH-M

This subsection studies the properties of SARAH-M with different parameters. As a benchmark, we provide the comparison results between SARAH-M (Algorithm 2) and the original SARAH approach (Algorithm 1) with the carefully selected step size. For convenience and clarity, when discussing the properties of SARAH-M (Algorithm 2) with a certain parameter, the other parameters are fixed.

As seen from SARAH-M (Algorithm 2), its performance depends on the parameters b, b_H and η. Therefore, we discuss the performance of SARAH-M (Algorithm 2) with the parameters b and β, respectively in the following, while the effect of the step size η in SARAH-M (Algorithm 2) is shown in Section 6.2. In all figures, the horizontal axis stands for the number of effective passes and the vertical axis stands for the sub-optimality: $F(\tilde{w}_s) - F(w_*)$, or the evaluation of $\|v_k\|^2$. The quantity $F(\tilde{w}_s) - F(w_*)$ (a.k.a. the sub-optimality) measures how the loss function $F(\tilde{w}_s)$ approximates the optimal value function $F(w_*)$. In other word, the smaller the value of $F(\tilde{w}_s) - F(w_*)$, the faster the corresponding algorithm converges. On the other side, we also employ the quantity $\|v_k\|^2$ to show the ability of the algorithm in controlling the variance for stochastic optimization algorithms. Obviously, the smaller the variance, the less impact the noisy gradients have on the algorithm. In other word, the algorithm converges speedily and executes robustly in practice.

The performance of SARAH-M (Algorithm 2) with the mini-batch size b on $a8a$ and $ijcnn1$ is displayed in Fig. 1. When discussing the numerical behavior of SARAH-M (Algorithm 2) with the mini-batch size b, we fix the parameters β and η. In detail, when performing SARAH-M (Algorithm 2) with different mini-batch sizes, we set $\eta = 0.2$ and $\beta = 0.1$ for different data sets. The selection of different mini-batch sizes is clearly provided in the legend of Fig. 1.

Fig. 1 shows that SARAH-M (Algorithm 2) has a much faster convergence rate than the original SARAH (Algorithm 1) with the carefully selected step size. Additionally, Fig. 1 implies that SARAH-M (Algorithm 2) is robust to the mini-batch size b. More specifically, the bottom line of Fig. 1 confirms the effectiveness of SARAH-M (Algorithm 2) in reducing the variance. Moreover, Fig. 1 further demonstrates momentum bring positive impact for variance reduced stochastic optimization algorithms.

Next, we will discuss the properties of SARAH-M (Algorithm 2) with the parameter β_k on $w8a$ and $MNIST$. We run SARAH-M (Algorithm 2) with different momentum coefficients under the setting of $b = 10$ and $\eta = 0.4$ on $a8a$ and $b = 10$ and $\eta = 0.6$ on $ijcnn1$. Specifically, we display the numerical results in Fig. 2. For the selection of other parameters for SARAH (Algorithm 1) and SARAH-M (Algorithm 2) are displayed in the legend of Fig. 2.

Fig. 2 shows that SARAH-M (Algorithm 2) performs well when selecting a small β.

In order to catch on SARAH-M (Algorithm 2) better, we further explore the impact of the step size in SARAH-M. Concretely, we fix the parameters β and b, then perform SARAH-M with different step sizes. The numerical results of such the case can be found in Fig. 3. Specifically, we show the details of step sizes in the legend of Fig. 3.

Observe by Fig. 3, the performance of SARAH-M (Algorithm 2) highly depends on the step size. Therefore, we should select the step size carefully for SARAH-M (Algorithm 2) in practice. This is why we develop the second algorithm AdaSARAH-M (Algorithm 3), which utilizes random Barzilai–Borwein technique to automatically obtain an appropriate online step size. Actually, Fig. 3 implies that the learning rate that can be chosen from (0.01, 0.2) arbitrarily will result in a better performance of SARAH-M (Algorithm 2) on different datasets. Such

[1] These data sets can be downloaded from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

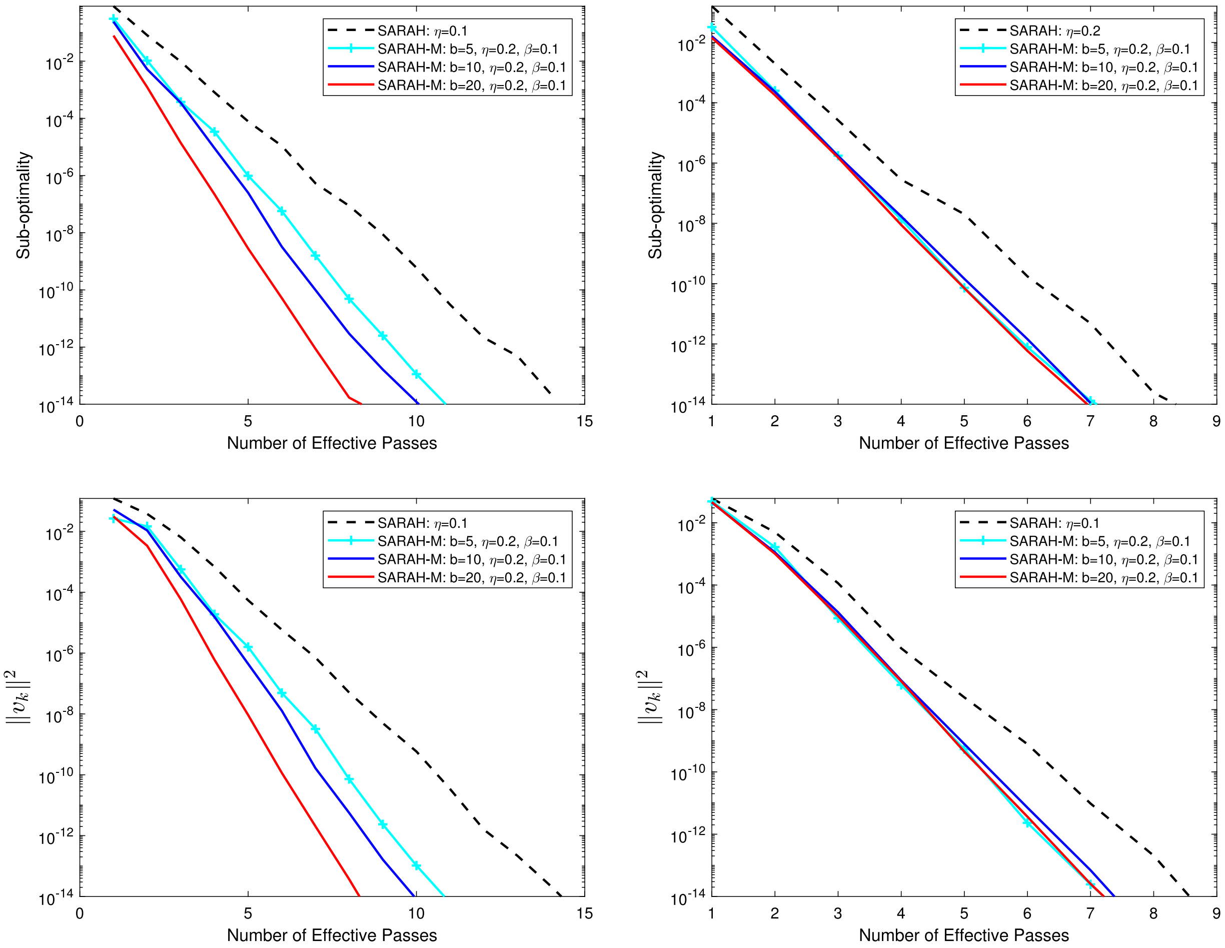

Fig. 1. Performance of SARAH-M with the mini-batch size b on $a8a$ (left) and $ijcnn1$ (right).

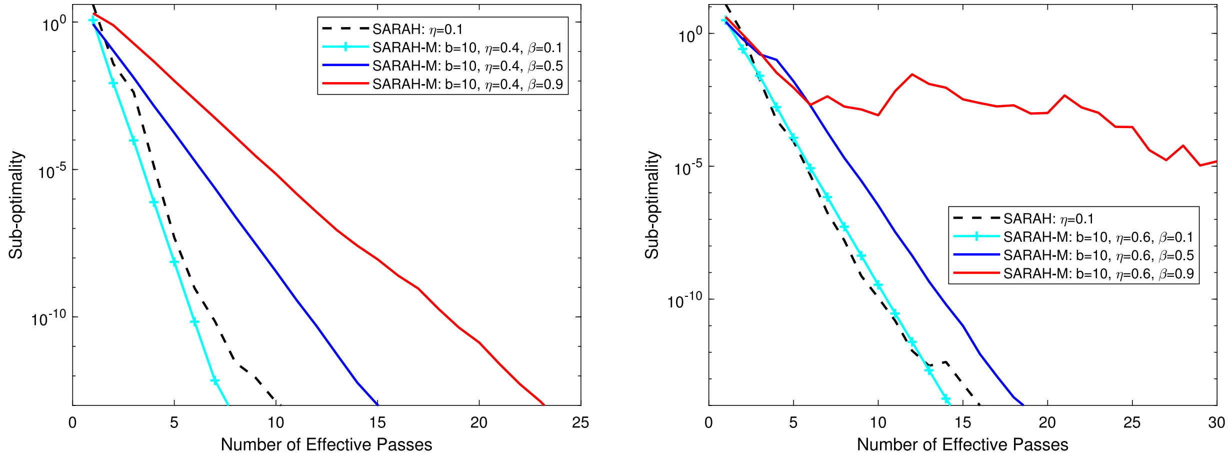

Fig. 2. Performance of SARAH-M with the momentum parameter β on $w8a$ (left) and $MNIST$ (right).

practice provides users some information to determine the learning rate practically.

6.2. Properties of AdaSARAH-M

In this subsection, we research the properties of AdaSARAH-M (Algorithm 3) with different parameters b, b_H and γ. To better understand the properties of AdaSARAH-M (Algorithm 3), when studying the effect of the parameters b and b_H in AdaSARAH-M (Algorithm 3), we provide the comparison results among SARAH (Algorithm 1), SARAH-M (Algorithm 2) and AdaSARAH-M (Algorithm 3). The details of different parameters are displayed in the legend of different figures.

The performance of AdaSARAH-M (Algorithm 3) with the parameter b is arranged in Fig. 4. We equip SARAH-M (Algorithm 2) with the same mini-batch sample b as AdaSARAH-M (Algorithm 3).

Here, to further validate the efficacy of stochastic optimization with momentum, we offer the comparison results of the time cost among the original SARAH algorithm, SARAH-M (Algorithm 2) and AdaSARAH-M (Algorithm 3) under the case that the latter two algorithms employ different mini-batch samples b. Particularly, the numerical results on

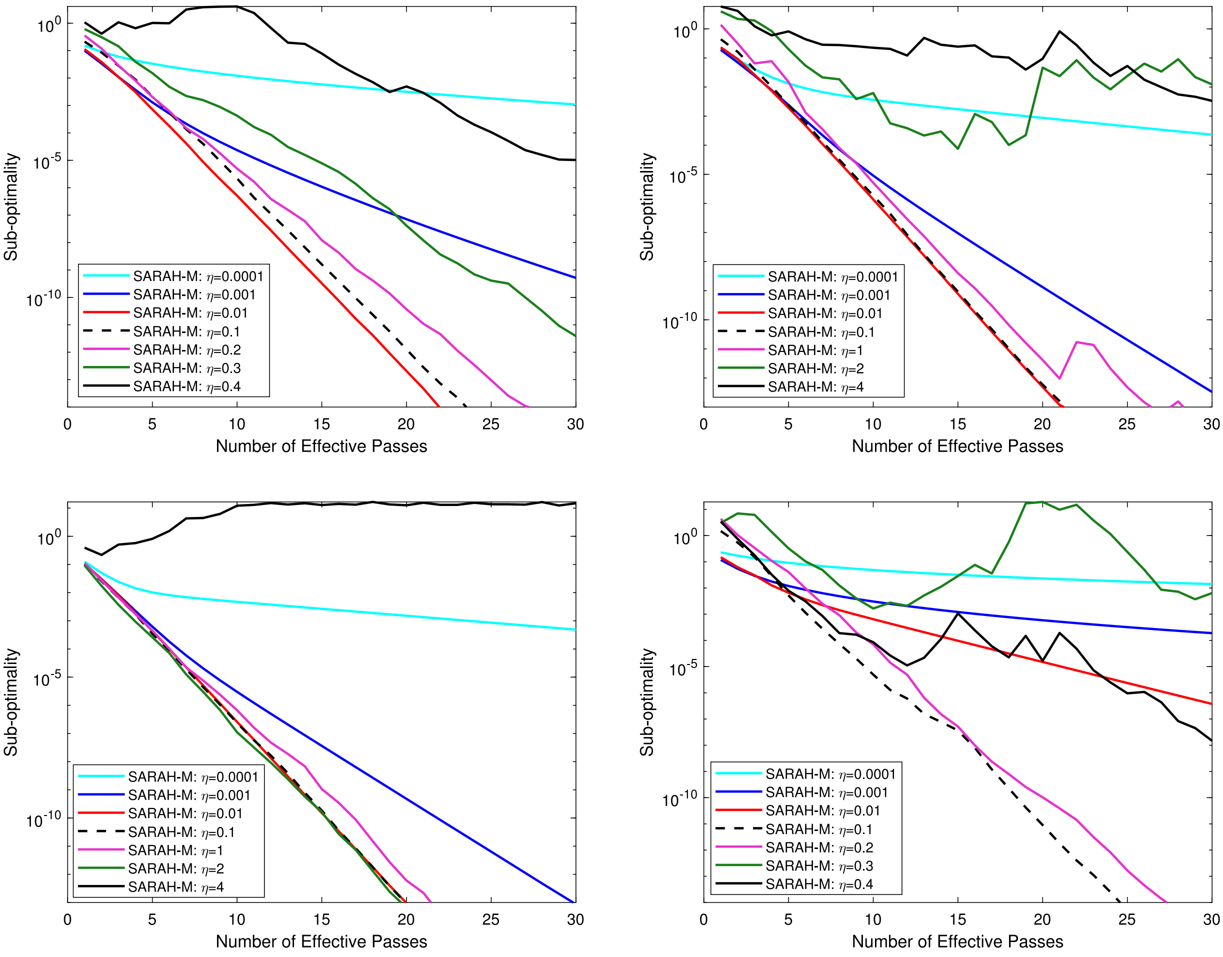

Fig. 3. Performance of SARAH-M with different step sizes on $a8a$ (top left), $w8a$ (top right), $ijcnn1$ (bottom left) and $MNIST$ (bottom right).

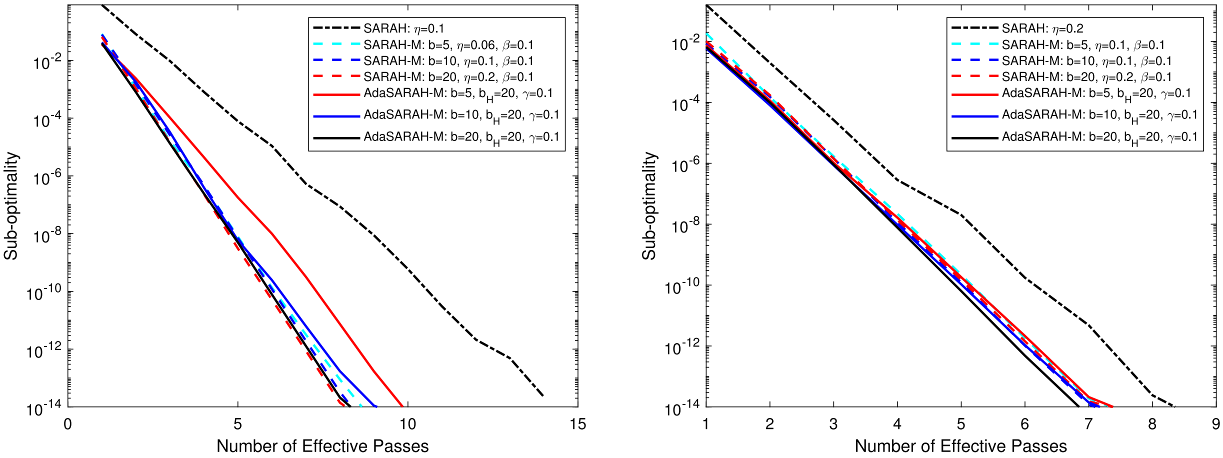

Fig. 4. Performance of AdaSARAH-M with the mini-batch size b on $a8a$ (left) and $ijcnn1$ (right).

$a8a$ and $ijcnn1$ are provided in Fig. 5. Notice that in Fig. 5, the x-axis is still employed to denote the number of effective passes, while the y-axis is used to represent the time consuming of each number of effective passes for different algorithms.

As seen from Fig. 5, the time cost of SARAH-M and AdaSARAH-M is lower than that of the original SARAH algorithm per each number of effective passes. One may think that a fast rate of SARAH-M and AdaSARAH-M because of the use of the mini-batch technique in SARAH-M and AdaSARAH-M. Actually, when adopting the same mini-batch samples between SARAH and SARAH-M, SARAH achieves similar time consuming to SARAH-M. Also, Fig. 5 demonstrates that SARAH-M and AdaSARAH-M achieve similar time consuming. To further show this, the total executing time of SARAH, SARAH-M and AdaSARAH-M is listed in Table 3.

As plotted in Fig. 4, AdaSARAH-M (Algorithm 3) performs well than the original SARAH (Algorithm 1). In addition, AdaSARAH-M

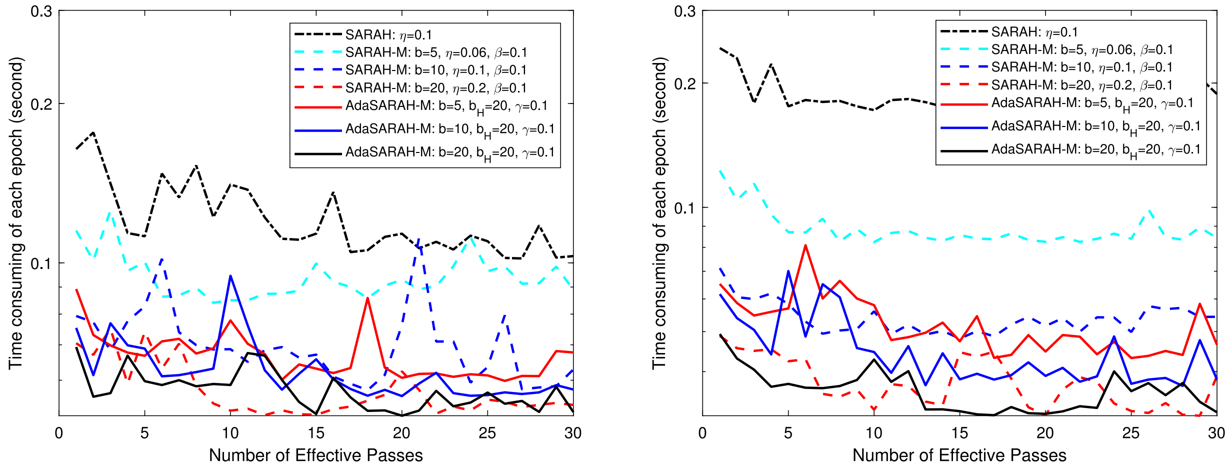

Fig. 5. Time consumption of SARAH, SARAH-M (Algorithm 2) and AdaSARAH-M (Alorithm 3) under the case that the latter two algorithms employ different mini-batch samples b on $a8a$ (left) and $ijcnn1$ (right).

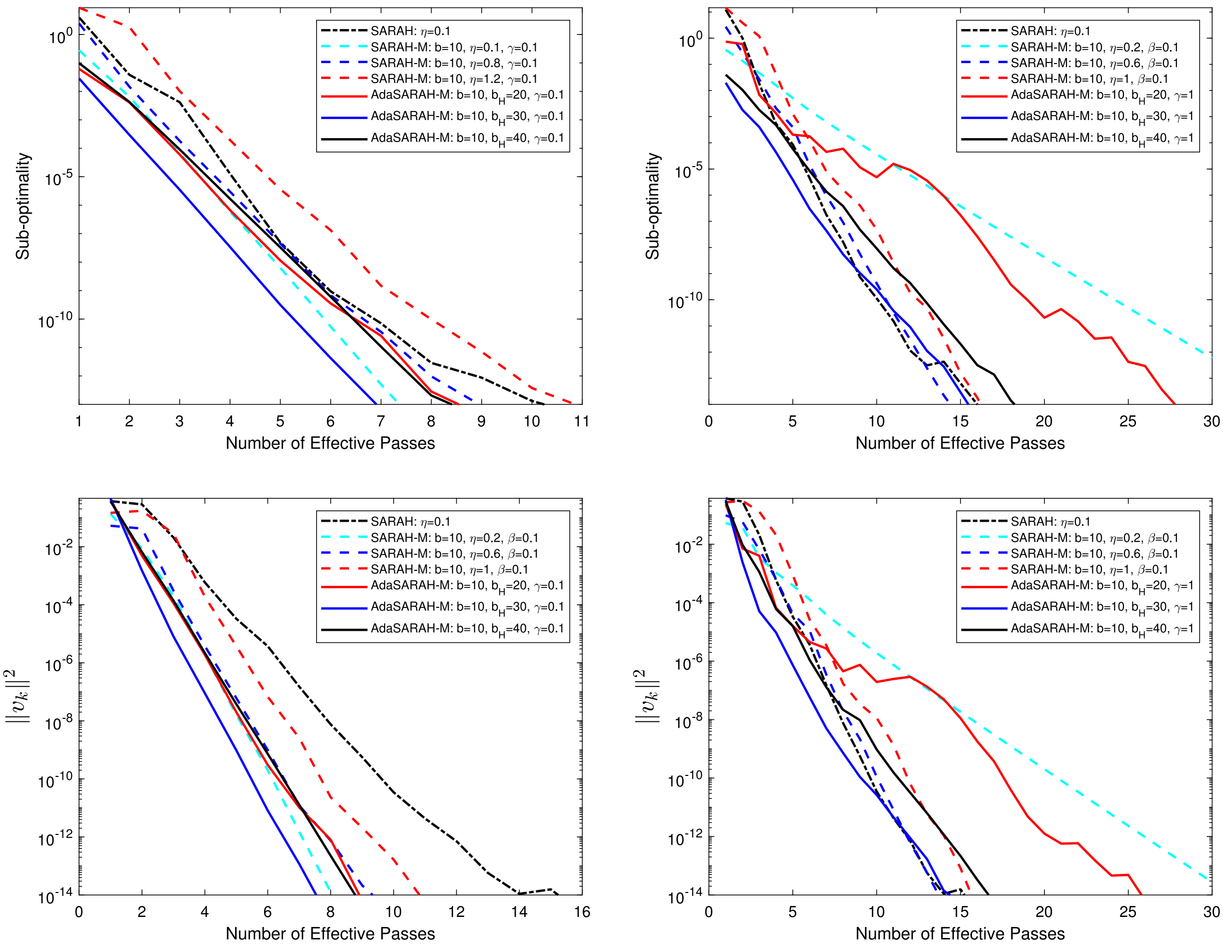

Fig. 6. Performance of AdaSARAH-M with the mini-batch size b_H on $w8a$ (left) and $MNIST$ (right).

(Algorithm 3) matches or even outperforms SARAH-M (Algorithm 2) with carefully selected step size.

Further, we research the properties of AdaSARAH-M (Algorithm 3) with the mini-batch size b_h. When performing SARAH-M (Algorithm 2), we set the same mini-batch sample b as AdaSARAH-M (Algorithm 3). Besides, we perform SARAH-M (Algorithm 2) with three different step sizes. For the details of these parameters, please see the legend of Fig. 6. Moreover, we plot the case of AdaSARAH-M (Algorithm 3) reducing the variance in Fig. 6.

The comparison results among SARAH (Algorithm 1), SARAH-M (Algorithm 2) and AdaSARAH-M (Algorithm 3) are displayed in Fig. 6. Fig. 6 shows that AdaSARAH-M (Algorithm 3) can achieve better performance than SARAH (Algorithm 1) and SARAH-M (Algorithm 2). Both Figs. 4 and 6 indicates that AdaSARAH-M (Algorithm 3) is robust to the mini-batch samples b and b_H.

At the end of this subsection, we discuss the effect of the parameter γ in AdaSARAH-M (Algorithm 3) on $w8a$ and $ijcnn1$. More specifically, we provide the numerical results in Fig. 7.

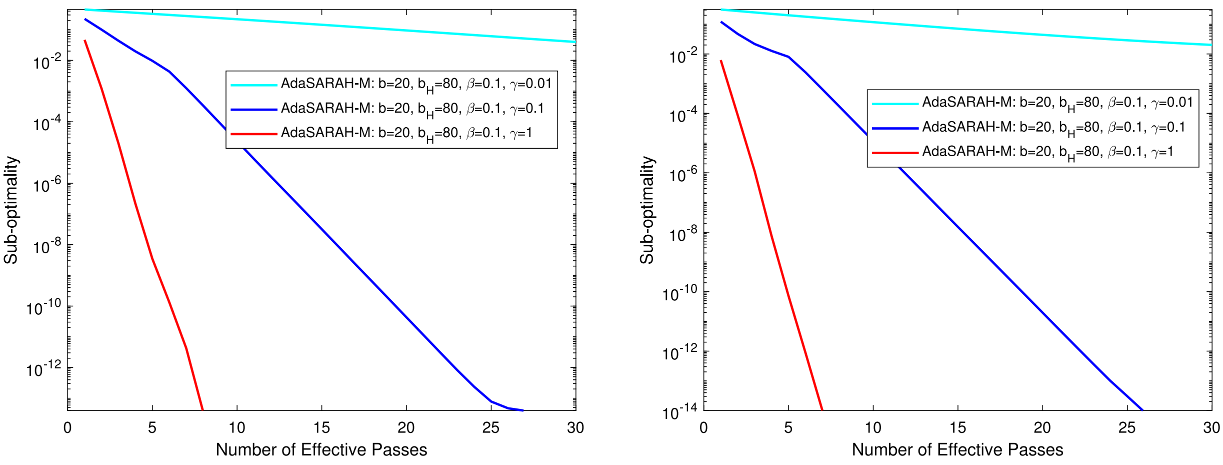

Fig. 7. Performance of AdaSARAH-M with the parameter γ on $w8a$ (left) and $ijcnn1$ (right).

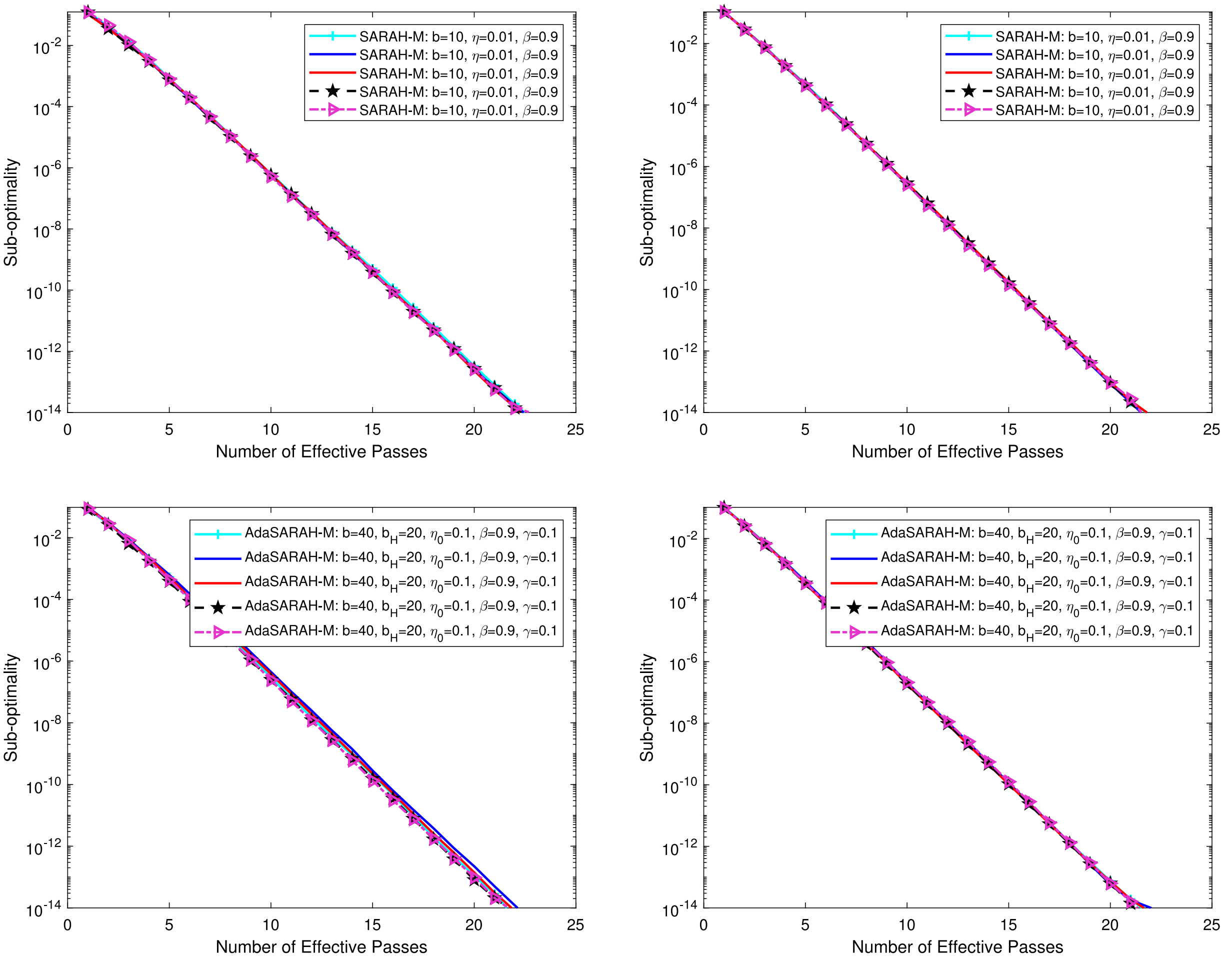

Fig. 8. Performing SARAH-M (the first line) and AdaSARAH-M (the second line) multiple times on the same setting on $a8a$ (left) and $ijcnn1$ (right).

Fig. 7 shows that a small γ makes AdaSARAH-M (Algorithm 3) converge slowly and a large γ makes AdaSARAH-M (Algorithm 3) converge quickly. However, a larger γ will lead to the divergence of the proposed algorithm. Actually, in most cases, we can set $\gamma = 1$ when taking a big mini-batch sample b_H, which reduces the difficulty in setting the crucial parameters for the proposed algorithm.

To confirm the robustness and reliability of the resulting algorithms, we will perform them multiple times on the same parameter settings.

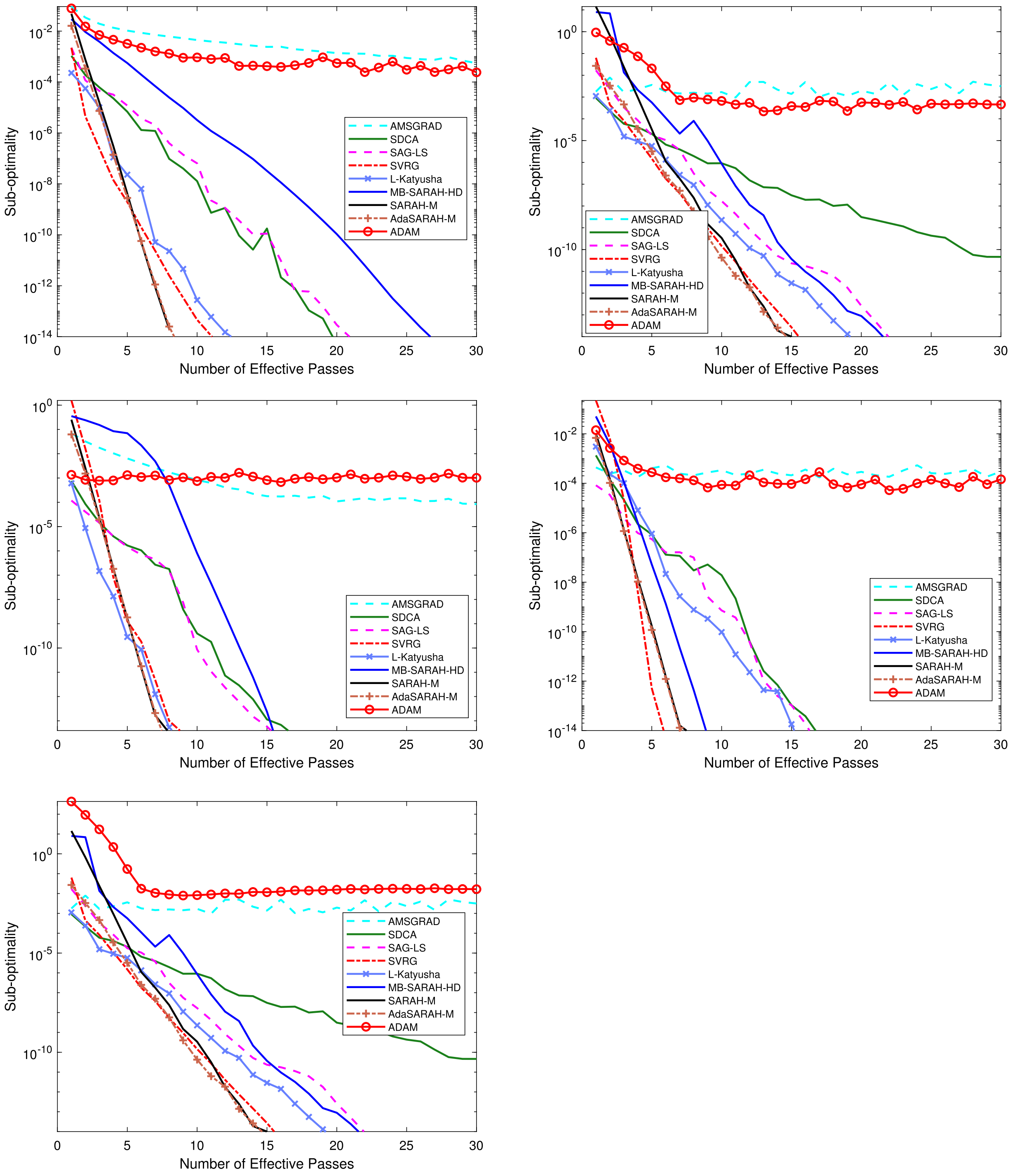

Fig. 9. Comparison of different methods on various data sets: $a8a$ (top left), $MNIST$ (top right), $w8a$ (middle left), $ijcnn1$(middle right) and $news20$ (bottom).

Specifically, we run SARAH-M (Algorithm 2) five times with $b = 10$, $\eta = 0.01$ and $\beta = 0.9$ on $a8a$ and $ijcnn1$. In addition, we also performed AdaSARAH-M (Algorithm 3) five times with $b = 20$, $b_H = 20$, $\eta_0 = 0.1$, $\gamma = 0.1$, and $\beta = 0.9$ on $a8a$ and $ijcnn1$. The numerical results are displayed in Fig. 8.

Fig. 8 confirms the robustness and reliability of SARAH-M (Algorithm 2) and AdaSARAH-M (Algorithm 3), respectively.

6.3. Comparison with other related algorithms

To greatly demonstrate the efficacy of SARAH-M (Algorithm 2) and AdaSARAH-M (Algorithm 3), we compare them with several state-of-the-art algorithms, which include ADAM (Kingma & Ba, 2015), AMSGRAD (A variant of ADAM) (Reddi et al., 2018), SVRG (Johnson & Zhang, 2013), SAG-LS (SAG with line search) (Schmidt et al., 2015), SDCA (Shalev-Shwartz & Zhang, 2013), L-Katyusha (loopless variants of Katyusha) (Kovalev et al., 2020), MB-SARAH-HD (SARAH with the hypergradient descent in the mini-batch setting) (Yang et al., 2020). Notice that we set the parameters for these algorithms as suggested in their original literature.

For example, when executing SVRG, we set $\eta = 0.01$ on $a8a$, $\eta = 8$ on $ijcnn1$, $\eta = 2$ on $w8a$, $\eta = 0.1$ on $MNIST$, and $\eta = 1.2$ on $news20$. When running ADAM and AMSGRAD, as many references done, we set $\beta_1 = 0.9$ and $\beta_2 = 0.99$ for all data sets. For MB-SARAH-HD, we adopt

Table 3
A comparison of total time consumption among SARAH, SARAH-M (Algorithm 2) and AdaSARAH-M (Alorithm 3) on $a8a$ and $ijcnn1$.

Algorithm	Total time on $a8a$ (s)	Total time on $ijcnn1$ (s)
SARAH	3.6415	5.6772
SARAH-M: $b = 5$	2.8248	2.6680
SARAH-M: $b = 10$	2.1126	1.6189
SARAH-M: $b = 20$	1.7262	1.1306
AdaSARAH-M: $b = 5$, $b_H = 20$	2.0088	1.5675
AdaSARAH-M: $b = 10$, $b_H = 20$	1.8829	1.3520
AdaSARAH-M: $b = 20$, $b_H = 20$	1.7238	1.0812

$\eta_0 = 0.001$ and $\beta = 0.01$ on $a8a$, $\eta_0 = 1$ and $\beta = 0.1$ on $ijcnn1$, $\eta_0 = 0.8$ and $\beta = 0.01$ on $MNIST$ and $news20$, and $\eta_0 = 1$ and $\beta = 0.01$ on $w8a$. The parameters of SDCA and L-Katyusha are set as suggested by Kovalev et al. (2020) and Shalev-Shwartz and Zhang (2013) respectively.

Fig. 9 shows that our SARAH-M (Algorithm 2) and AdaSARAH-M (Algorithm 3) algorithms perform well and even converge faster than state-of-the-art stochastic optimization algorithms on different data sets.

7. Conclusion

In this paper, we provided the understanding of how the momentum improves variance reduced stochastic gradient algorithms. Specifically, we first explored the performance of SARAH with the momentum term and developed a novel variance reduced stochastic gradient algorithm, termed as SARAH-M. Secondly, we proposed an adaptive SARAH-M method, referred to as AdaSARAH-M, by incorporating the RBB technique into SARAH-M, which provided a simple way to determine the step size for the original SARAH-M algorithm. We theoretically analyzed the convergence of SARAH-M and AdaSARAH-M when the loss function is strongly convex and showed that it attained a linear rate of convergence. The analysis in complexity of the proposed algorithms indicated that they can outperform modern stochastic optimization algorithms. A range of numerical results, compared with state-of-the-art algorithms for benchmarking machine learning problems verified the effectiveness of the proposed algorithms. Also, the numerical experiments in discussing the effect of different parameters in the proposed algorithms demonstrated that they were robust to several crucial parameters, which made the users feel free to perform the algorithms.

CRediT authorship contribution statement

Zhuang Yang: Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by grants from the National Natural Science Foundation of China under Grant 62302325. This work was also supported by the Natural Science Foundation of Jiangsu Province in China under Grant BK20230485.

Appendix A. Proofs for SARAH-M

A.1. Proof of Lemma 1

Proof. According to Assumption 1, we have

$$\begin{aligned}\mathbb{E}[F(w_{k+1})] &\le \mathbb{E}\Big[F(w_k) + \langle \nabla F(w_k), w_{k+1} - w_k\rangle + \frac{L}{2}\|w_{k+1} - w_k\|^2\Big]\\ &= \mathbb{E}\Big[F(w_k) + \langle \nabla F(w_k), \beta_{k-1}(w_k - w_{k-1}) - \eta G_k\rangle\\ &+ \frac{L}{2}\|\beta_{k-1}(w_k - w_{k-1}) - \eta G_k\|^2\Big]\\ &= \mathbb{E}\Big[\ F(w_k) + \beta_{k-1}\langle \nabla F(w_k), w_k - w_{k-1}\rangle - \eta\langle \nabla F(w_k), G_k\rangle\\ &+ \frac{L}{2}\|\beta_{k-1}(w_k - w_{k-1}) - \eta G_k\|^2\ \Big]\end{aligned} \tag{A.1}$$

where the first equality holds since the fact that $w_{k+1} = v_k - \eta G_k = w_k + \beta_{k-1}(w_k - w_{k-1}) - \eta G_k$.

Further, utilizing the fact that (i) $a^2+b^2 \ge 2ab$, (ii) $(a+b)^2 \le 2(a^2+b^2)$ and (iii) $\langle a, b\rangle = \frac{1}{2}[\|a\|^2 + \|b\|^2 - \|a-b\|^2]$, we have

$$\begin{aligned}\mathbb{E}[F(w_{k+1})] &\overset{(i),(ii)}{\le} \mathbb{E}\Big[\ F(w_k) + \beta_{k-1}\Big[\frac{1}{2}\|\nabla F(w_k)\|^2 + \frac{1}{2}\|w_k - w_{k-1}\|^2\Big]\\ &- \eta\langle \nabla F(w_k), G_k\rangle + \frac{L}{2}\big[2\beta_{k-1}^2\|w_k - w_{k-1}\|^2 + 2\eta^2\|G_k\|^2\big]\ \Big]\\ &\overset{(iii)}{\le} \mathbb{E}\Big[\ F(w_k) + \frac{\beta_{k-1}}{2}\|\nabla F(w_k)\|^2 + \frac{\beta_{k-1}}{2}\|w_k - w_{k-1}\|^2\\ &- \frac{\eta}{2}\big[\|\nabla F(w_k)\|^2 + \|G_k\|^2 - \|\nabla F(w_k) - G_k\|^2\big] + L\eta^2\|G_k\|^2\\ &+ L\beta_{k-1}^2\|w_k - w_{k-1}\|^2\ \Big]\\ &= \mathbb{E}\Big[\ F(w_k) + \Big(\frac{\beta_{k-1}}{2} - \frac{\eta}{2}\Big)\|\nabla F(w_k)\|^2 + \Big(\frac{\beta_{k-1}}{2} + L\beta_{k-1}^2\Big)\\ &\cdot \|w_k - w_{k-1}\|^2 + \frac{\eta}{2}\|\nabla F(w_k) - G_k\|^2 - \Big(\frac{\eta}{2} - L\eta^2\Big)\|G_k\|^2\ \Big]\\ &\le \mathbb{E}\Big[\ F(w_k) + \Big(\frac{\hat{\beta}}{2} - \frac{\eta}{2}\Big)\|\nabla F(w_k)\|^2 + \Big(\frac{\hat{\beta}}{2} + L\beta_{k-1}^2\Big)\ \|\ w_k\\ &- w_{k-1}\ \|^2 + \frac{\eta}{2}\|\nabla F(w_k) - G_k\|^2 - \Big(\frac{\eta}{2} - L\eta^2\Big)\|G_k\|^2\ \Big],\end{aligned} \tag{A.2}$$

where in the last inequality we use $\beta_k \le \hat{\beta}$.

To satisfy the inequality (A.2), it is sufficient to set

$$\begin{aligned}&\mathbb{E}[F(w_{k+1})]\\ &\le \mathbb{E}\Big[F(w_k) + \Big(\frac{\hat{\beta}}{2} - \frac{\eta}{2}\Big)\|\nabla F(w_k)\|^2 + \frac{\eta}{2}\|\nabla F(w_k) - G_k\|^2\\ &- \Big(\frac{\eta}{2} - L\eta^2\Big)\|G_k\|^2\Big].\end{aligned} \tag{A.3}$$

When summing the inequality over $k = 0, 1, \ldots, m$, we have

$$\begin{aligned}&\mathbb{E}[F(w_{k+1})]\\ &\le \mathbb{E}[F(w_0)] - \Big(\frac{\eta}{2} - \frac{\hat{\beta}}{2}\Big)\sum_{k=0}^{m}\mathbb{E}\big[\|\nabla F(w_k)\|^2\big] + \frac{\eta}{2}\sum_{k=0}^{m}\mathbb{E}\ \big[\|\ \nabla F(w_k)\\ &- G_k\ \|^2\big] - \Big(\frac{\eta}{2} - L\eta^2\Big)\sum_{k=0}^{m}\mathbb{E}\big[\|G_k\|^2\big].\end{aligned} \tag{A.4}$$

From (A.4), we infer

$$\begin{aligned}&\mathbb{E}\big[\|\nabla F(w_k)\|^2\big]\\ &\le \frac{2}{\eta - \hat{\beta}}\mathbb{E}[F(w_0) - F(w_{m+1})] + \frac{\eta}{\eta - \hat{\beta}}\sum_{k=0}^{m}\mathbb{E}\big[\|\nabla F(w_k) - G_k\|^2\big]\\ &- \frac{2}{\eta - \hat{\beta}}\Big(\frac{\eta}{2} - L\eta^2\Big)\sum_{k=0}^{m}\mathbb{E}\big[\|G_k\|^2\big] \quad \square\end{aligned} \tag{A.5}$$

A.2. Proof of Theorem 1

Proof. Combining Lemma 2 and the fact that $\|a\|^2 \ge \frac{1}{2}\|b\|^2 - \|b-a\|^2$, we have

$$\begin{aligned}&\mathbb{E}[\|\nabla F(v_k) - G_k\|^2]\\ &\ge \frac{1}{2}\mathbb{E}[\|\nabla F(w_k) - G_k\|^2] - \mathbb{E}[\|\nabla F(w_k) - G_k - \nabla F(v_k) + G_k\|^2]\\ &= \frac{1}{2}\mathbb{E}[\|\nabla F(w_k) - G_k\|^2] - \mathbb{E}[\|\nabla F(w_k) - \nabla F(v_k)\|^2]. \qquad (A.6)\end{aligned}$$

Based on (A.6), to hold the inequality (9) (appearing in Lemma 2), it is enough to set

$$\mathbb{E}[\|\nabla F(w_k) - G_k\|^2] \le \frac{2}{b}\left(\frac{n-b}{n-1}\right)L^2\eta^2\sum_{j=1}^{k}\mathbb{E}[\|G_{j-1}\|^2]. \qquad (A.7)$$

According to the fact that $\|\nabla F(w_0) - G_0\|^2 = 0$, when summing the inequality (A.7) over $k = 0, 1, \ldots, m$, we obtain

$$\begin{aligned}\sum_{k=1}^{m}\mathbb{E}[\|\nabla F(w_k) - G_k\|^2] &\le \frac{2}{b}\left(\frac{n-b}{n-1}\right)L^2\eta^2\big[m\mathbb{E}\big[\|G_0\|^2\big] + (m-1)\\ &\quad\cdot\mathbb{E}\big[\|G_1\|^2\big] + \cdots + \mathbb{E}\big[\|G_{m-1}\|^2\big]\big]. \qquad (A.8)\end{aligned}$$

Further, we have

$$\begin{aligned}&\sum_{k=1}^{m}\mathbb{E}[\|\nabla F(w_k) - G_k\|^2] - \left(\frac{\eta}{2} - L\eta^2\right)\sum_{k=0}^{m}\mathbb{E}\big[\|G_k\|^2\big]\\ &\le \frac{2}{b}\left(\frac{n-b}{n-1}\right)L^2\eta^2\big[m\mathbb{E}\big[\|G_0\|^2\big] + (m-1)\mathbb{E}\big[\|G_1\|^2\big]\\ &+ \cdots + \mathbb{E}\big[\|G_{m-1}\|^2\big]\big] - \left(\frac{\eta}{2} - L\eta^2\right)\sum_{k=0}^{m}\mathbb{E}\big[\|G_k\|^2\big]\\ &\le \left[\frac{2}{b}\left(\frac{n-b}{n-1}\right)L^2\eta^2 m - \left(\frac{\eta}{2} - L\eta^2\right)\right]\sum_{k=0}^{m}\mathbb{E}\big[\|G_k\|^2\big]\\ &\le 0, \qquad (A.9)\end{aligned}$$

where the last inequality holds due to (11) appearing in Theorem 1.

Therefore, from Lemma 1 and $\tilde{w}_s = w_m$, we derive

$$\begin{aligned}\mathbb{E}\big[\|\nabla F(w_m)\|^2\big] &= \frac{1}{m+1}\sum_{k=0}^{m}\mathbb{E}\big[\|\nabla F(w_k)\|^2\big]\\ &\le \frac{1}{m+1}\cdot\frac{2}{\eta-\hat{\beta}}\mathbb{E}[F(w_k) - F(w_*)]. \qquad (A.10)\end{aligned}$$

Here, we have finished the proof of Theorem 1. □

A.3. Proof of Theorem 2

Proof. According to the fact that $w_0 = \tilde{w}_{s-1}$ and $\tilde{w}_s = w_m$, where $s \ge 1$, we infer

$$\begin{aligned}\mathbb{E}[\|\nabla F(\tilde{w}_s)|\tilde{w}_{s-1}\|^2] &= \mathbb{E}[\|\nabla F(\tilde{w}_s)|w_0\|^2]\\ &\le \frac{1}{m+1}\cdot\frac{2}{\eta-\hat{\beta}}\mathbb{E}[F(w_0) - F(w_*)]\\ &\le \frac{1}{\mu(m+1)(\eta-\hat{\beta})}\mathbb{E}[\|F(w_0)\|^2]\\ &= \frac{1}{\mu(m+1)(\eta-\hat{\beta})}\mathbb{E}[\|F(\tilde{w}_{s-1})\|^2]\end{aligned}$$

where the second inequality holds due to the inequality (6).

Finally, by setting $\rho = \frac{1}{\mu(m+1)(\eta-\hat{\beta})}$, we obtain the desired results. □

Appendix B. Proof of AdaSARAH-M

B.1. Proof of Lemma 3

Proof. From Assumption 2, we easily obtain

$$\begin{aligned}\eta &= \frac{\gamma}{b_H}\frac{\|v_k - v_{k-1}\|^2}{|\langle v_k - v_{k-1}, \nabla F_{S_H}(v_k) - \nabla F_{S_H}(v_{k-1})\rangle|}\\ &\le \frac{\gamma}{b_H}\frac{\|v_k - v_{k-1}\|^2}{\mu\|v_k - v_{k-1}\|^2} = \frac{\gamma}{\mu b_H}\end{aligned}$$

In addition, from Assumption 1, we easily obtain

$$\begin{aligned}\eta &= \frac{\gamma}{b_H}\frac{\|v_k - v_{k-1}\|^2}{|\langle v_k - v_{k-1}, \nabla F_{S_H}(v_k) - \nabla F_{S_H}(v_{k-1})\rangle|}\\ &\ge \frac{\gamma}{b_H}\frac{\|v_k - v_{k-1}\|^2}{L\|v_k - v_{k-1}\|^2} = \frac{\gamma}{Lb_H} \quad □\end{aligned}$$

B.2. Proof of Lemma 4

Proof. According to the proof in Lemma 1, we have

$$\begin{aligned}\mathbb{E}[F(w_{k+1})] &\le \mathbb{E}\Big[F(w_k) + \frac{\beta_{k-1}}{2}\|\nabla F(w_k)\|^2 + \frac{\beta_{k-1}}{2}\|w_k - w_{k-1}\|^2\\ &\quad - \frac{\eta_k}{2}\big[\|\nabla F(w_k)\|^2 + \|G_k\|^2 - \|\nabla F(w_k) - G_k\|^2\big] + L\eta_k^2\|G_k\|^2\\ &\quad + L\beta_{k-1}^2\|w_k - w_{k-1}\|^2\Big]\\ &\le \mathbb{E}\Big[F(w_k) + \frac{\hat{\beta}}{2}\|\nabla F(w_k)\|^2 + \frac{\hat{\beta}}{2}\|w_k - w_{k-1}\|^2\\ &\quad - \frac{\gamma}{2\mu b_H}\big[\|\nabla F(w_k)\|^2 + \|G_k\|^2 - \|\nabla F(w_k) - G_k\|^2\big]\\ &\quad + \frac{L\gamma^2}{\mu^2 b_H^2}\|G_k\|^2\\ &\quad + L\hat{\beta}^2\|w_k - w_{k-1}\|^2\Big]\\ &= \mathbb{E}\left[F(w_k) + \left(\frac{\hat{\beta}}{2} - \frac{\gamma}{2\mu b_H}\right)\|\nabla F(w_k)\|^2 + \left(\frac{\hat{\beta}}{2} + L\hat{\beta}^2\right)\right.\\ &\quad\cdot\|w_k - w_{k-1}\|^2 + \frac{\gamma}{2\mu b_H}\|\nabla F(w_k) - G_k\|^2\\ &\quad\left. - \left(\frac{\gamma}{2\mu b_H} - \frac{L\gamma^2}{\mu^2 b_H^2}\right)\|G_k\|^2\right], \qquad (B.1)\end{aligned}$$

where the second equality holds due to Lemma 3 and $\beta_k \le \hat{\beta}$.

To make the inequality (B.1), it is enough to set

$$\begin{aligned}&\mathbb{E}[F(w_{k+1})]\\ &\le \mathbb{E}\Big[F(w_k) + \left(\frac{\hat{\beta}}{2} - \frac{\gamma}{2\mu b_H}\right)\|\nabla F(w_k)\|^2 + \frac{\gamma}{2\mu b_H}\|\nabla F(w_k) - G_k\|^2\\ &- \left(\frac{\gamma}{2\mu b_H} - \frac{L\gamma^2}{\mu^2 b_H^2}\right)\|G_k\|^2\Big]. \qquad (B.2)\end{aligned}$$

When summing the inequality over $k = 0, 1, \ldots, m$, we have

$$\begin{aligned}&\mathbb{E}[F(w_{k+1})]\\ &\le \mathbb{E}[F(w_0)] - \left(\frac{\gamma}{2\mu b_H} - \frac{\hat{\beta}}{2}\right)\sum_{k=0}^{m}\mathbb{E}\big[\|\nabla F(w_k)\|^2\big] + \frac{\gamma}{2\mu b_H}\\ &\cdot\sum_{k=0}^{m}\mathbb{E}[\|\nabla F(w_k) - G_k\|^2] - \left(\frac{\gamma}{2\mu b_H} - \frac{L\gamma^2}{\mu^2 b_H^2}\right)\sum_{k=0}^{m}\mathbb{E}\big[\|G_k\|^2\big]. \qquad (B.3)\end{aligned}$$

From (B.3), we infer

$$\begin{aligned}&\mathbb{E}\big[\|\nabla F(w_k)\|^2\big]\\ &\le \frac{2\mu b_H}{\gamma - \hat{\beta}\mu b_H}\mathbb{E}[F(w_0) - F(w_{m+1})] + \frac{\gamma}{\gamma - \hat{\beta}\mu b_H}\sum_{k=0}^{m}\mathbb{E}\,\big[\|\nabla F(w_k)\\ &- G_k\|^2\big] - \frac{\gamma}{\gamma - \hat{\beta}\mu b_H}\left(1 - \frac{2L\gamma^2}{\mu b_H}\right)\sum_{k=0}^{m}\mathbb{E}\big[\|G_k\|^2\big]. \quad □ \qquad (B.4)\end{aligned}$$

B.3. Proof of Theorem 3

Proof. From the proof of Theorem 1, we ascertain

$$\mathbb{E}[\|\nabla F(w_k) - G_k\|^2] \leq \frac{2L^2 b_H^2}{b\mu^2 b_H^2}\left(\frac{n-b}{n-1}\right)\sum_{j=1}^{k}\mathbb{E}[\|G_{j-1}\|^2]. \tag{B.5}$$

According to the fact $\|\nabla F(w_0) - G_0\|^2 = 0$, when summing the inequality (B.5) over $k = 0, 1, \ldots, m$, we obtain

$$\sum_{k=1}^{m}\mathbb{E}[\|\nabla F(w_k) - G_k\|^2] \leq \frac{2L^2 b_H^2}{b\mu^2 b_H^2}\left(\frac{n-b}{n-1}\right)\left[\, m\mathbb{E}\left[\|G_0\|^2\right] + (m-1) \cdot \mathbb{E}\left[\|G_1\|^2\right] + \cdots + \mathbb{E}\left[\|G_{m-1}\|^2\right]\,\right]. \tag{B.6}$$

Further, we have

$$\begin{aligned}
&\sum_{k=1}^{m}\mathbb{E}[\|\nabla F(w_k) - G_k\|^2] - \left(1 - \frac{2L\gamma^2}{\mu b_H}\right)\sum_{k=0}^{m}\mathbb{E}\left[\|G_k\|^2\right]\\
&\leq \frac{2L^2 b_H^2}{b\mu^2 b_H^2}\left(\frac{n-b}{n-1}\right)\Big[m\mathbb{E}\left[\|G_0\|^2\right] + (m-1)\mathbb{E}\left[\|G_1\|^2\right]\\
&+ \cdots + \mathbb{E}\left[\|G_{m-1}\|^2\right]\Big] - \left(1 - \frac{2L\gamma^2}{\mu b_H}\right)\sum_{k=0}^{m}\mathbb{E}\left[\|G_k\|^2\right]\\
&\leq \left[\frac{2L^2 b_H^2}{b\mu^2 b_H^2}\left(\frac{n-b}{n-1}\right)m - \left(1 - \frac{2L\gamma^2}{\mu b_H}\right)\right]\sum_{k=0}^{m}\mathbb{E}\left[\|G_k\|^2\right]\\
&\leq 0,
\end{aligned} \tag{B.7}$$

where the last inequality holds due to (12) appearing in Theorem 3.

Therefore, from Lemma 3 and $\tilde{w}_s = w_m$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\|\nabla F(w_m)\|^2\right] &= \frac{1}{m+1}\sum_{k=0}^{m}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right]\\
&\leq \frac{1}{m+1} \cdot \frac{2\mu b_H}{\gamma - \hat{\beta}\mu b_H}\mathbb{E}[F(w_k) - F(w_*)]. \quad \square
\end{aligned}$$

References

Allen-Zhu, Z. (2017). Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, *18*, 8194–8244.

Amari, S.-i. (2013). Dreaming of mathematical neuroscience for half a century. *Neural Networks*, *37*, 48–51.

Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, *60*, 223–311.

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

Csiba, D., & Richtárik, P. (2018). Importance sampling for minibatches. *Journal of Machine Learning Research*, *19*, 962–982.

Cutkosky, A., & Orabona, F. (2019). Momentum-based variance reduction in non-convex sgd. In *Proceedings of the 33rd international conference on neural information processing systems* (pp. 15236–15245).

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

Duchi, J. C., Bartlett, P. L., & Wainwright, M. J. (2012). Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, *22*, 674–701.

Fang, C., Li, C. J., Lin, Z., & Zhang, T. (2018). SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in neural information processing systems* (pp. 689–699).

Gidel, G., Berard, H., Vignoud, G., Vincent, P., & Lacoste-Julien, S. (2018). A variational inequality perspective on generative adversarial networks. In *International conference on learning representations*.

Gitman, I., Lang, H., Zhang, P., & Xiao, L. (2019). Understanding the role of momentum in stochastic gradient methods. *Advances in Neural Information Processing Systems*, *32*, 9633–9643.

Higham, N. J., & Strabić, N. (2016). Anderson acceleration of the alternating projections method for computing the nearest correlation matrix. *Numerical Algorithms*, *72*, 1021–1042.

Hou, J., Zeng, X., Wang, J., & Chen, J. (2022). Distributed momentum-based frank-wolfe algorithm for stochastic optimization. *IEEE/CAA Journal of Automatica Sinica*, *10*, 685–699.

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

Kingma, D. P., & Ba, J. (2015). ADAM: A method for stochastic optimization. In *ICLR (Poster)*.

Kovalev, D., Horváth, S., & Richtárik, P. (2020). Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop. In *Algorithmic learning theory* (pp. 451–467). PMLR.

Lan, G. (2020). *First-order and stochastic optimization methods for machine learning*. Springer Nature.

Lei, L., & Jordan, M. (2017). Less than a single pass: Stochastically controlled stochastic gradient. In *Artificial intelligence and statistics* (pp. 148–156). PMLR.

Li, M., Zhang, T., Chen, Y., & Smola, A. J. (2014). Efficient mini-batch training for stochastic optimization. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 661–670).

Liu, Y., Shang, F., & Jiao, L. (2019). Accelerated incremental gradient descent using momentum acceleration with scaling factor. In *International joint conference on artificial intelligence* (pp. 3045–3051).

Loizou, N., & Richtárik, P. (2020). Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods. *Computational Optimization and Applications*, *77*, 653–710.

Ma, J., & Yarats, D. (2018). Quasi-hyperbolic momentum and adam for deep learning. In *International conference on learning representations*.

Mou, W., Li, C. J., Wainwright, M. J., Bartlett, P. L., & Jordan, M. I. (2020). On linear stochastic approximation: Fine-grained polyak-ruppert and non-asymptotic concentration. In *Conference on learning theory* (pp. 2947–2997). PMLR.

Mu, Y., Liu, W., Liu, X., & Fan, W. (2016). Stochastic gradient made stable: A manifold propagation approach for large-scale optimization. *IEEE Transactions on Knowledge and Data Engineering*, *29*, 458–471.

Nesterov, Y. (2004). *Introductory lectures on convex optimization : basic course*. Kluwer Academic.

Neu, G., & Rosasco, L. (2018). Iterate averaging as regularization for stochastic gradient descent. In *Conference on learning theory* (pp. 3222–3242). PMLR.

Nguyen, L. M., van Dijk, M., Phan, D. T., Nguyen, P. H., Weng, T.-W., & Kalagnanam, J. R. (2022). Finite-sum smooth optimization with SARAH. *Computational Optimization and Applications*, *82*, 561–593.

Nguyen, L. M., Liu, J., Scheinberg, K., & Takáč, M. (2017a). SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning-Volume 70* (pp. 2613–2621). JMLR. org.

Nguyen, L. M., Liu, J., Scheinberg, K., & Takáč, M. (2017b). Stochastic recursive gradient algorithm for nonconvex optimization. arXiv preprint arXiv:1705.07261.

Nguyen, L. M., Nguyen, P. H., Richtárik, P., Scheinberg, K., Takáč, M., & van Dijk, M. (2019). New convergence aspects of stochastic gradient algorithms. *Journal of Machine Learning Research*, *20*, 1–49.

Nitanda, A. (2014). Stochastic proximal gradient descent with acceleration techniques. In *Advances in neural information processing systems* (pp. 1574–1582).

Polyak, B. T. (1987). *Optimization software, Introduction to optimization* (p. 1). New York: Inc. Publications Division.

Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, *30*, 838–855.

Reddi, S. J., Kale, S., & Kumar, S. (2018). On the convergence of adam and beyond. In *International conference on learning representations*.

Roux, N. L., Schmidt, M., & Bach, F. R. (2012a). A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in neural information processing systems* (pp. 2663–2671).

Roux, N. L., Schmidt, M., & Bach, F. R. (2012b). A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in neural information processing systems* (pp. 2663–2671).

Ruszczynski, A., & Syski, W. (1983). Stochastic approximation method with gradient averaging for unconstrained problems. *IEEE Transactions on Automatic Control*, *28*, 1097–1105.

Schmidt, M., Babanezhad, R., Ahmed, M. O., Defazio, A., Clifton, A., & Sarkar, A. (2015). Non-uniform stochastic average gradient method for training conditional random fields. In *International conference on artificial intelligence and statistics*.

Scieur, D., Bach, F., & d'Aspremont, A. (2017). Nonlinear acceleration of stochastic algorithms. *Advances in Neural Information Processing Systems*, *30*, 3982–3991.

Scieur, D., d'Aspremont, A., & Bach, F. (2020). Regularized nonlinear acceleration. *Mathematical Programming*, *179*, 47–83.

Sebbouh, O., Gower, R. M., & Defazio, A. (2021). Almost sure convergence rates for stochastic gradient descent and stochastic heavy ball. In *Conference on learning theory* (pp. 3935–3971). PMLR.

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

Shang, F., Jiao, L., Zhou, K., Cheng, J., Ren, Y., & Jin, Y. (2018). Asvrg: Accelerated proximal svrg. In *Asian conference on machine learning* (pp. 815–830). PMLR.

Sidi, A. (2003). *Practical extrapolation methods: theory and applications, Vol. 10*. Cambridge University Press.

Spall, J. C. (2012). Stochastic optimization. In *Handbook of computational statistics* (pp. 173–201). Springer.

Toth, A., Ellis, J. A., Evans, T., Hamilton, S., Kelley, C., Pawlowski, R., & Slattery, S. (2017). Local improvement results for anderson acceleration with inaccurate function evaluations. *SIAM Journal on Scientific Computing*, *39*, S47–S65.

Toth, A., & Kelley, C. (2015). Convergence analysis for anderson acceleration. *SIAM Journal on Numerical Analysis*, *53*, 805–819.

Tran, T. H., Nguyen, L. M., & Tran-Dinh, Q. (2021). SMG: A shuffling gradient-based method with momentum. In *International conference on machine learning* (pp. 10379–10389). PMLR.

Wang, Z., Ji, K., Zhou, Y., & Tarokh, V. (2019). Spiderboost and momentum: Faster variance reduction algorithms. *Advances in Neural Information Processing Systems*, *32*, 2406–2416.

Wang, B., Nguyen, T., Sun, A. L., Baraniuk, R. G., & Osher, S. J. (2022). Scheduled restart momentum for accelerated stochastic gradient descent. *SIAM Journal on Imaging Sciences*, *15*, 738–761.

Wei, F., Bao, C., & Liu, Y. (2021). Stochastic anderson mixing for nonconvex stochastic optimization. *Advances in Neural Information Processing Systems*, *34*.

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

Xie, J., Ma, Z., Xue, G., Sun, J., Zheng, Y., & Guo, J. (2021). Ds-ui: Dual-supervised mixture of gaussian mixture models for uncertainty inference in image recognition. *IEEE Transactions on Image Processing*, *30*, 9208–9219.

Xin, R., Kar, S., & Khan, U. A. (2020). Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence. *IEEE Signal Processing Magazine*, *37*, 102–113.

Xu, Y., & Xu, Y. (2023). Momentum-based variance-reduced proximal stochastic gradient method for composite nonconvex stochastic optimization. *Journal of Optimization Theory and Applications*, *196*, 266–297.

Xu, Y., Yuan, Z., Yang, R., & Yang, T. (2019). On the convergence of (stochastic) gradient descent with extrapolation for non-convex minimization. In *International joint conference on artificial intelligence* (pp. 4003–4009).

Yang, Z., Chen, Z., & Wang, C. (2020). An accelerated stochastic variance-reduced method for machine learning problems. *Knowledge-Based Systems*, *198*, Article 105941.

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

Yasuda, S., Mahboubi, S., Indrapriyadarsini, S., Ninomiya, H., & Asai, H. (2019). A stochastic variance reduced nesterov's accelerated quasi-newton method. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)* (pp. 1874–1879). IEEE.

Zhou, K., Ding, Q., Shang, F., Cheng, J., Li, D., & Luo, Z. (2019). Direct acceleration of SAGA using sampled negative momentum. In *International conference on artificial intelligence and statistics* (pp. 1602–1610).