Signal Processing 159 (2019) 171–186

Contents lists available at ScienceDirect

Signal Processing

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

Accelerated stochastic gradient descent with step size selection rules

Zhuang Yang[a,b], Cheng Wang[a,*], Zhemin Zhang[a], Jonathan Li[a,c]

[a] *Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, FJ 361005, China*
[b] *School of Electronics and Communication Engineering, Sun Yat-Sen University, Guangzhou 510275, China*
[c] *Department of Geography and Environmental Management, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

ARTICLE INFO

Article history:
Received 24 May 2018
Revised 5 February 2019
Accepted 7 February 2019
Available online 7 February 2019

Keywords:
Empirical risk minimization
Accelerated proximal gradient method
Mini-batches
Barzilai-Borwein method
Variance reduction
Convex optimization

ABSTRACT

Accelerated stochastic gradient descent (ASGD) methods, which incorporate accelerated proximal gradient (APG) and stochastic gradient (SG), have received considerable attention recently for solving regularized risk minimization problems in signal/image processing, statistics and machine learning. However, there has been a paucity of practical guidance proposed for resolving one of the major issues in ASGD: how to choose an appropriate step size. To solve this problem, we propose to use the Barzilai-Borwein (BB) method to automatically compute step size for the accelerated mini-batch Prox-SVRG (Acc-Prox-SVRG) method (the state of the art ASGD method), thereby obtaining a new accelerated method: Acc-Prox-SVRG-BB. We prove the convergence of Acc-Prox-SVRG-BB and show that its complexity is comparable with the best known stochastic gradient methods. In addition, we incorporate Beck and Teboulle's APG (FISTA) and Prox-SVRG in a mini-batch setting and obtain another new accelerated gradient descent method, FISTA-Prox-SVRG, which requires the selection of fewer unknown parameters than those required in Acc-Prox-SVRG. Finally, we introduce the BB method into FISTA-Prox-SVRG to further show the efficacy of the BB method. Numerical results demonstrate the advantage of our algorithms.

1. Introduction

Many optimization problems that arise in signal/image processing, statistics and machine learning can be formulated as stochastic composite optimization (SCO) problems where the expectation of stochastic loss function plus a possibly non-smooth regularization term is minimized. Instances include support vector machines [1–3], sparse, group sparse linear regression [4,5] and sparse Logistic regression [6,7].

Accelerated proximal gradient (APG) [8–11] and stochastic gradient (SG) [12–15] methods are at the heart of dealing with the SCO problems. Recently, some researchers considered combining the accelerated gradient methods with the stochastic gradient methods to generate a series of celebrated works. For instance, by properly modifying the Nesterov optimal smooth method [16], Ghadimi et al. [17] developed a generic accelerated stochastic approximation (AC-SA) algorithmic framework and showed that AC-SA can be specialized to yield optimal or nearly optimal methods for different classes of SCO problems. Shalev-Shwartz et al. [18] introduced the Nesterov's acceleration method into the proximal stochastic dual coordinate ascent (Prox-SDCA) method for regularized loss minimization. Moreover, they show that this method, using the accelerated procedure, improves the dependence on the condition number. For the SCO problems, Nitanda [19] proposed the accelerated mini-batch Prox-SVRG (Acc-Prox-SVRG) method which incorporates the Nesterov's acceleration method [16] and the proximal proximal stochastic variance reduction gradient (Prox-SVRG) method [20] in the mini-batch setting. He proved that the overall complexity of Acc-Prox-SVRG, with an appropriate mini-batch size, is more efficient than both the Prox-SVRG and Nesterov's acceleration methods. Furthermore, Nitanta [21] proposed using another accelerated gradient method [22], similar to the Nesterov's acceleration method combined with Prox-SVRG in a mini-batch setting, to obtain a new accelerated stochastic gradient method, the accelerated efficient mini-batch SVRG (AMSVRG) method. He also showed that AMSVRG achieves a fast convergence complexity for general convex and optimal strongly convex problems. Other ASGD methods for solving the SCO problems can be found in [23–25] and the references therein.

Because of the selection of noisy gradients, the convergence of stochastic gradient (SG) methods [26–29] is often guaranteed assuming that step size is diminishing but not too rapidly. A diminishing step size sequence results in a slow convergence rate for stochastic gradient methods. Even for strongly convex and smooth problems, the convergence rate of SG is sub-linear [30]. Some state

[*] Corresponding author.
E-mail address: cwang@xmu.edu.cn (C. Wang).

https://doi.org/10.1016/j.sigpro.2019.02.010
0165-1684/© 2019 Published by Elsevier B.V.

of the art SG methods (e.g. SAG [31], SAGA [32], SDCA [33], SVRG [34], S2GD [35], mS2GD [36] and Acc-Prox-SVRG) converge linearly with a constant step size, chosen from multiple pre-prepared step size which is time consuming. As variants of the SG methods, the ASGD methods [17,18,21,23,25] also employ a diminishing step size, or a best tuned step size. Specifically, there is no guidance for the specific choice of the step size sequence.

To solve this shortcoming associated with stochastic optimization, Yousefian et al. [37] proposed two adaptive steplength schemes, recursive steplength stochastic approximation (RSA) scheme and cascading steplength stochastic approximation (CSA) scheme, and theoretically analyzed the convergence of two new iteration schemes for strongly convex differentiable stochastic optimization problems. Mahsereci et al. [38], by performing a line search to obtain step size for univariate optimization objectives in a Gaussian process, arrived at a lightweight "black-box" algorithm that exposes no parameter to the user. Tan et al. [39] employed the Barzilai-Borwein (BB) method to compute the step size for stochastic gradient descent (SGD) methods and its variants, thereby leading to two new methods: SVRG-BB and SGD-BB. Especially, they proved that SVRG-BB has a linear convergence rate for strongly convex objective functions. Motivated by this work, De et al. [40] considered a "big batch" for SGD and employed backtracking line search and BB methods to compute step size for their proposed methods. Specifically, they showed that, on a range of convex problems, using an adaptive step size method based on the BB curvature estimate empirically performs better than the backtracking line search. In addition, Yang et al. [41] proposed a random BB (RBB) method to compute step size for mini-batch algorithms. They combined the RBB method with mS2GD and Acc-Prox-SVRG to obtain two algorithms: mS2GD-RBB and Acc-Prox-SVRG-RBB. However, the authors did not provide the details of Acc-Prox-SVRG-RBB, but showed just the convergence of mS2GD-RBB for strongly convex objective functions.

The primary contributions of our work are as follows:

(i) We propose a new ASGD method, Acc-Prox-SVRG-BB, which uses the BB method to automatically compute step size for Acc-Prox-SVRG. We prove the convergence of Acc-Prox-SVRG-BB and show that its complexity achieves the same level as the best known stochastic gradient methods.

(ii) To reduce the difficulty of choosing the parameters in Acc-Prox-SVRG, we propose a new method, FISTA-Prox-SVRG, which incorporates Beck and Teboulle's APG (a.k.a FISTA) and Prox-SVRG in a mini-batch setting.

(iii) To further validate the effectiveness of the BB method in the ASGD methods, we combine the BB method with FISTA-Prox-SVRG. Numerical experiments show the efficacy of our proposed methods.

The remainder of this paper is organized as follows. In Section 2, we give the problem formulation and background. In Section 3, we introduce the details of Acc-Prox-SVRG-BB. In Section 4, we analyze the convergence of our Acc-Prox-SVRG-BB method. In Section 5, we propose the FISTA-Prox-SVRG algorithm and combine the BB method with FISTA-Prox-SVRG. In Section 6, we present our numerical results. Finally, in Section 7 we conclude the paper.

2. Problem formulation and background

We consider the following SCO problems,

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

where each $f_i(w)$ is smooth, and $R(w)$ is possibly non-smooth but allows a simple proximal mapping. For convenience, we set $F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$.

The results presented in this paper are established under the following assumptions:

Assumption 1. Each gradient of the convex function, $f_i(w)$, in (1) is differentiable and Lipschitz continuous with a positive constant L, which means that for all w and v in $\mathbb{R}^d$, we have

$$\|\nabla f_i(w) - \nabla f_i(v)\|_2 \le L\|w - v\|_2. \tag{2}$$

From Assumption 1, we readily ascertain that the gradient of the average function, $F(w)$, is also Lipschitz continuous, i.e., there exists a positive constant, L, such that for all w, $v \in \mathbb{R}^d$, it holds that

$$\|\nabla F(w) - \nabla F(v)\|_2 \le L\|w - v\|_2. \tag{3}$$

Assumption 2. $P(w)$ is μ-strongly convex, i.e. there exists $\mu > 0$ such that for all w, $v \in \mathbb{R}^d$,

$$(\nabla P(w) - \nabla P(v))^T (w - v) \ge \mu \parallel w - v \parallel_2^2, \tag{4}$$

or equivalently

$$P(w) \ge P(v) + \nabla P(v)^T (w - v) + \frac{\mu}{2}\|w - v\|_2^2. \tag{5}$$

Also, we need the objective function, $P(w)$, to be Lipschitz continuous. In addition, the strong convexity parameter, μ, is often less than the Lipshcitz constant, L, i.e., $\mu < L$ and the ratio L/μ, denoted as κ, is often called the condition number of problem (1).

Assumption 3. The regularization function $R(w)$ is a lower semi-continuous proper convex function; however, it can be non-differentiable or non-continuous.

3. Algorithm

In this section, we introduce the Acc-Prox-SVRG-BB method. We first introduce the BB step size and the Acc-Prox-SVRG method.

3.1. Barzilai-Borwein step size

Suppose we want to solve the unconstrained minimization problem

$$\min_{w\in\mathbb{R}^d} F(w), \tag{6}$$

where F is differentiable. The Newton-iteration is

$$w_{k+1} = w_k - H_k^{-1}\nabla F(w_k), \tag{7}$$

where $\nabla F(w_k)$ denotes the gradient of $F(w)$ at w_k and H_k is an approximation of the Hessian matrix of $F(w)$ at w_k. Now approximating the Hessian matrix H_k by $H_k = (1/\eta_k)I$ with $\eta_k > 0$ and substituting (7), we have

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

Introduce $H_k = (1/\eta_k)I$ into the secant equation, $H_k s_k = y_k$, where $s_k = w_k - w_{k-1}$ and $y_k = \nabla F(w_k) - \nabla F(w_{k-1})$ for $k \ge 1$. Hence, by minimizing the residual of the secant equation, i.e., $\|(1/\eta_k)s_k - y_k\|_2^2$, we obtain the following choice for η_k:

$$\eta_k = \|s_k\|_2^2/(s_k^T y_k). \tag{9}$$

The Eq. (9) is often called the BB step size, which was proposed by Barzilai and Borwein [42] to solve unconstrained optimization problems. Due to its simplicity and numerical efficiency, much attetion has been focused on the BB method and it has been extended to many applications, such as image processing [43], compressed sensing [44] and sparse reconstruction [45].

3.2. Accelerated mini-Batch prox-SVRG

A standard way to solve problem (1) is the proximal stochastic gradient (Prox-SG) method in which the iteration scheme is:

$$w_{k+1} = \text{prox}_{\eta_k R}(w_k - \eta_k \nabla f_i(w_k)), \tag{10}$$

where $\nabla f_i(w_k)$ represents the gradient of the i-th component function at w_k and "prox" defines the proximal mapping, defined as follows:

$$\text{prox}_{\eta R}(z) \overset{\text{def}}{=} \arg\min_{w\in\mathbb{R}^d}\left\{\frac{1}{2}\|w-z\|_2^2 + \eta R(w)\right\},$$

Compared with Prox-SG, Acc-Prox-SVRG uses the variance reduction technique of SVRG [20], which allows the selection of a large step size and results in a rapid convergence. During each stage, this method performs Nesterov's acceleration method-like iterations and employs the following direction with mini-batch instead of a single gradient in (10):

$$G_k = \nabla F_{I_k}(v_k) - \nabla F_{I_k}(\widetilde{w}) + \nabla F(\widetilde{w}), \tag{11}$$

where $I_k = \{i_1, \ldots, i_b\}$ is a randomly chosen size b subset of $\{1, \ldots, n\}$ and $\nabla F_{I_k} = \frac{1}{b}\Sigma_{j=1}^{b}\nabla f_{i_j}$. At the beginning of each stage,the initial point w_1 is set to be $\widetilde{w}$, and at the end of stage, $\widetilde{w}$ is updated. Moreover, conditioned on v_k, $\mathbb{E}[G_k] = \nabla F(v_k)$, so that G_k is also an unbiased estimator.

3.3. Acc-Prox-SVRG with BB step size

According to the above descriptions, we now give the details of Acc-Prox-SVRG-BB in Algorithm 1.

Algorithm 1 Acc-Prox-SVRG-BB.

```
Input: update frequency $m$; mini-batch size $b \in [n]$; initial point $\widetilde{w}_1$; initial step size $\eta_1$
for $s = 1, 2, \ldots$ do
    Set $\widetilde{w} = \widetilde{w}_s$
    Compute and store $g_s = (1/n)\sum_{i=1}^{n}\nabla f_i(\widetilde{w})$, $g_s^{'} = \partial P(\widetilde{w})$
    if $s > 1$ then
        $$\eta_s = \frac{\delta}{m}\|\widetilde{w}_s - \widetilde{w}_{s-1}\|_2^2/(\widetilde{w}_s - \widetilde{w}_{s-1})^T(g_s^{'} - g_{s-1}^{'}) \tag{12}$$
    end if
    Set $v_1 = w_1 = \widetilde{w}$
    for $k = 1, \ldots, m$ do
        Randomly pick subset $I_k \in \{1, \ldots, n\}$ of size $b$
        Compute a stochastic estimate of $\nabla F(w_k)$;
            $G_k = \nabla F_{I_k}(v_k) - \nabla F_{I_k}(\widetilde{w}) + g_s$
        $w_{k+1} = prox_{\eta_s R}(v_k - \eta_s G_k)$
        $v_{k+1} = w_{k+1} + \beta_k(w_{k+1} - w_k)$
    end for
    Set $\widetilde{w}_{s+1} = w_{m+1}$
end for
```

Note that, in Algorithm 1, we use the sub-gradient of the nonsmooth convex objective function, $P(w)$, to compute the step size[1] In addition, to ensure convergence of Acc-Prox-SVRG-BB, we multiply the parameter δ, where values of δ are from the range (0, 1]. If we set $\eta_s = \eta$, then the Acc-Prox-SVRG-BB method reduces to the original Acc-Prox-SVRG method. Moreover, for the first epoch of Acc-Prox-SVRG-BB, we set an initial step size η_1. However, numerical experiments showed that the performance of our proposed method is insensitive to the choice of initial step size, η_1.

4. Convergence analysis

For clarity, we first show the definition of estimation sequence. From [16], we have

Definition 1. A pair of sequence $\{\Phi_k(w)\}_{k=0}^{\infty}$ and $\{\lambda_k\}_{k=0}^{\infty}$, $\lambda_k \geq 0$ is called an estimate sequence of function $f(w)$ if $\lambda_k \to 0$. For any $w \in \mathbb{R}^d$ and all $k \geq 0$ we have

$$\Phi_k \leq (1-\lambda_k)f(w) + \lambda\Phi_0(w) \tag{13}$$

According to [16], to satisfy (13), we can set

$$\lambda_{k+1} = (1-\alpha_k)\lambda_k,$$

$$\Phi_{k+1}(w) = (1-\alpha_k)\Phi_k(w) + \alpha_k\Big(f(v_k) + (\nabla f(v_k), w - v_k) + \frac{\mu}{2}\|w - v_k\|_2^2\Big),$$

where (i) f is a strongly convex; (ii) $\Phi_0(w)$ is an arbitrary function on $\mathbb{R}^d$; (iii) $\{v_k\}_{k=0}^{\infty}$ is an arbitrary sequence on $\mathbb{R}^d$; (iv) $\{\alpha_k\}_{k=0}^{\infty}$: $\alpha_k \in (0, 1)$, $\sum_{k=0}^{\infty}\alpha_k = \infty$; (v) $\lambda_0 = 1$. It is easily proved that (13) is satisfied when the above two iteration schemes are used.

The iteration schemes of $\{\lambda_k\}$ and $\{\Phi_k\}$ provide some rules for updating the estimate sequence by a recursion. Actually, when we choose, Φ_0, as a simple quadratic function, the estimate sequence $\{\Phi_k(w)\}$ is simplified as a quadratic function sequence set. For clarity, we offer an explanation.

From Lemma 2.2.3 in [16], we know that when setting $\Phi_0 = \Phi_0^* + \frac{\mu}{2}\|w - z_0\|_2^2$, we have $\Phi_k = \Phi_k^* + \frac{\mu}{2}\|w - z_k\|_2^2$, where $\Phi_k^* = \min_{w\in\mathbb{R}^d}\ \Phi_k(w)$ and $z_k = \arg\min_{w\in\mathbb{R}^d}\ \Phi_k(w)$.

Now, we briefly prove Lemma 2.2.3 in [16]. Note that $\nabla^2\Phi_0(w) = \mu I$, where I is an identity matrix. Let us prove that $\nabla^2\Phi_k(w) = \mu I$ fore all $k \geq 0$. Actually, if that is true for some k, then, from the iteration scheme of $\{\Phi_k\}$, we have

$$\nabla^2\Phi_{k+1}(w) = (1-\alpha_k)\nabla^2\Phi_k(w) + \mu\alpha_k I \equiv \mu I$$

This finishes the proof of Lemma 2.2.3 in [16].

Under above assumptions and definition, we analyze the convergence of Acc-Prox-SVRG-BB described in Algorithm 1 and provide some notations and definitions. By the definition of a proximity operator, the iteration scheme on $\{w_k\}$ can be rewritten as:

$$w_{k+1} = v_k - \eta_k(G_k + \xi_k), \tag{14}$$

where $\xi_k \in \partial R(w_{k+1})$.

We now define the estimate sequence $\Phi_k(w)$ (k=1, 2, ..., m+1) by

$$\Phi_1(w) = P(w_1) + \frac{\mu}{2}\|w - w_1\|_2^2,$$

$$\Phi_{k+1}(w) = (1-\alpha_k)\Phi_k(w) + \alpha_k\Big(F_{I_k}(v_k) + (G_k, w - v_k) + \frac{\mu}{2}\|w - v_k\|_2^2 + R(w_{k+1}) + (\xi_k, w - w_{k+1})\Big),\ for\quad k \geq 1$$

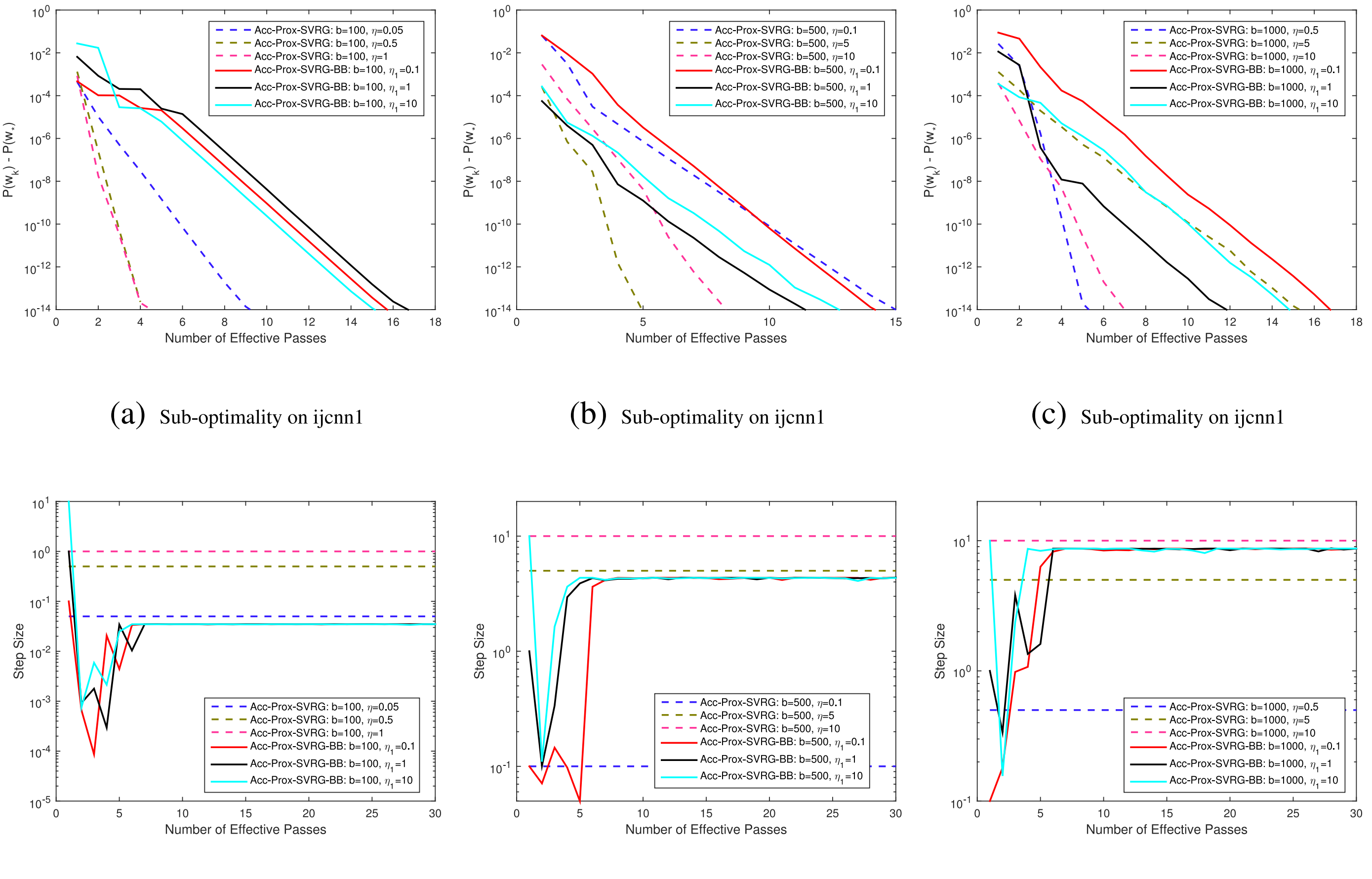
We set $\alpha_k = \sqrt{\frac{\mu\delta}{Cm}}$, where $\mu < C < L$. Hence, we have the following estimate sequence iteration scheme:

$$\Phi_{k+1}(w) = \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\Phi_k(w) + \sqrt{\frac{\mu\delta}{Cm}}\Big(F_{I_k}(v_k) + (G_k, w - v_k) + \frac{\mu}{2}\|w - v_k\|_2^2 + R(w_{k+1}) + (\xi_k, w - w_{k+1})\Big),\quad for\quad k \geq 1$$

We set

$$\Phi_k^* = min_{w\in\mathbb{R}^d}\Phi_k(w) \qquad and \qquad z_k = argmin_{w\in\mathbb{R}^d}\Phi_k(w)$$

[1] We adopt the definition of sub-gradient from [16]. Namely, if P is a convex function, then a vector g is called the sub-gradient of the function P at the point $w_0 \in dom\,R$ if, for any $w \in dom\,P$, we have that $P(w) \geq P(w_0) + \langle g, w - w_0\rangle$.



(a) Sub-optimality on ijcnn1

(b) Sub-optimality on ijcnn1

(c) Sub-optimality on ijcnn1

(d) Step size on ijcnn1

(e) Step size on ijcnn1

(f) Step size on ijcnn1

Fig. 1. Comparison of Acc-Prox-SVRG-BB and Acc-Prox-SVRG with fixed step sizes on *ijcnn*1. The dashed lines for Acc-Prox-SVRG with fixed step size η given in the legend. The solid line stand for Acc-Prox-SVRG-BB with different initial step size η_1 given in the legend.

Since $\nabla^2\Phi_k(w)=\mu I_n$, it follows that for $\forall w\in\mathbb{R}^d$,

$$\Phi_k(w)=\Phi_k^*+\frac{\mu}{2}\|w-z_k\|_2^2, \tag{15}$$

The following lemma is the key to the analysis of our method. The proof of the lemmas and theorem are in Appendix A.

Lemma 1. *Suppose that Assumption 1 and Assumption 2 hold, then the boundary of η_k computed in Algorithm 1 is $\left[\frac{\delta}{Lm},\frac{\delta}{\mu m}\right]$.*

For convenience, but without loss of generality, we set $\eta_k=\frac{\delta}{Cm}$ in our convergence analysis, where $\mu<C<L$.

Lemma 2. *Considering Acc-Prox-SVRG-BB in Algorithm 1 under Assumption 1, Assumption 2, Assumption 3 and Lemma 1. If $m>\frac{2L\delta}{C}$, then for $k\geq 1$ we have*

$$\mathbb{E}[\Phi_k(w)]\leq P(w)+\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)^{k-1}(\Phi_1-P)(w)\quad and \tag{16}$$

$$\mathbb{E}[P(w_k)]\leq\mathbb{E}\left[\Phi_k^*+\sum_{l=1}^{k-1}\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)^{k-1-l}\left\{-\frac{\mu}{2}\frac{1-\frac{\mu\delta}{Cm}}{\sqrt{\frac{\mu\delta}{Cm}}}\|w_l-v_l\|_2^2+\frac{\delta}{Cm}\|\nabla F(v_l)-G_l\|_2^2\right\}\right] \tag{17}$$

From Lemma 1, we know that the step sizes, η_k, vary in $[\frac{\delta}{Lm},\frac{\delta}{\mu m}]$. In our proof, when we set $\eta_k=\frac{\delta}{Cm}$, where $\mu<C<L$, we have $\frac{\delta}{Lm}<\frac{\delta}{Cm}<\frac{\delta}{\mu m}$. Although the step sizes, η_k, are different at each iteration, we always find a constant, C, that satisfies (16) and (17) in Lemma 2. Actually, when we take $C<L$, the inequalities (16) and (17) always hold.

Lemma 3. *For $k\geq 1$ we have*

$$z_{k+1}=\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)z_k+\sqrt{\frac{\mu\delta}{Cm}}v_k-\sqrt{\frac{\delta}{\mu Cm}}(G_k+\xi_k)\quad and \tag{18}$$

$$z_k-v_k=\sqrt{\frac{Cm}{\mu\delta}}(v_k-w_k) \tag{19}$$

Moreover, from [19], we have the following two lemmas:

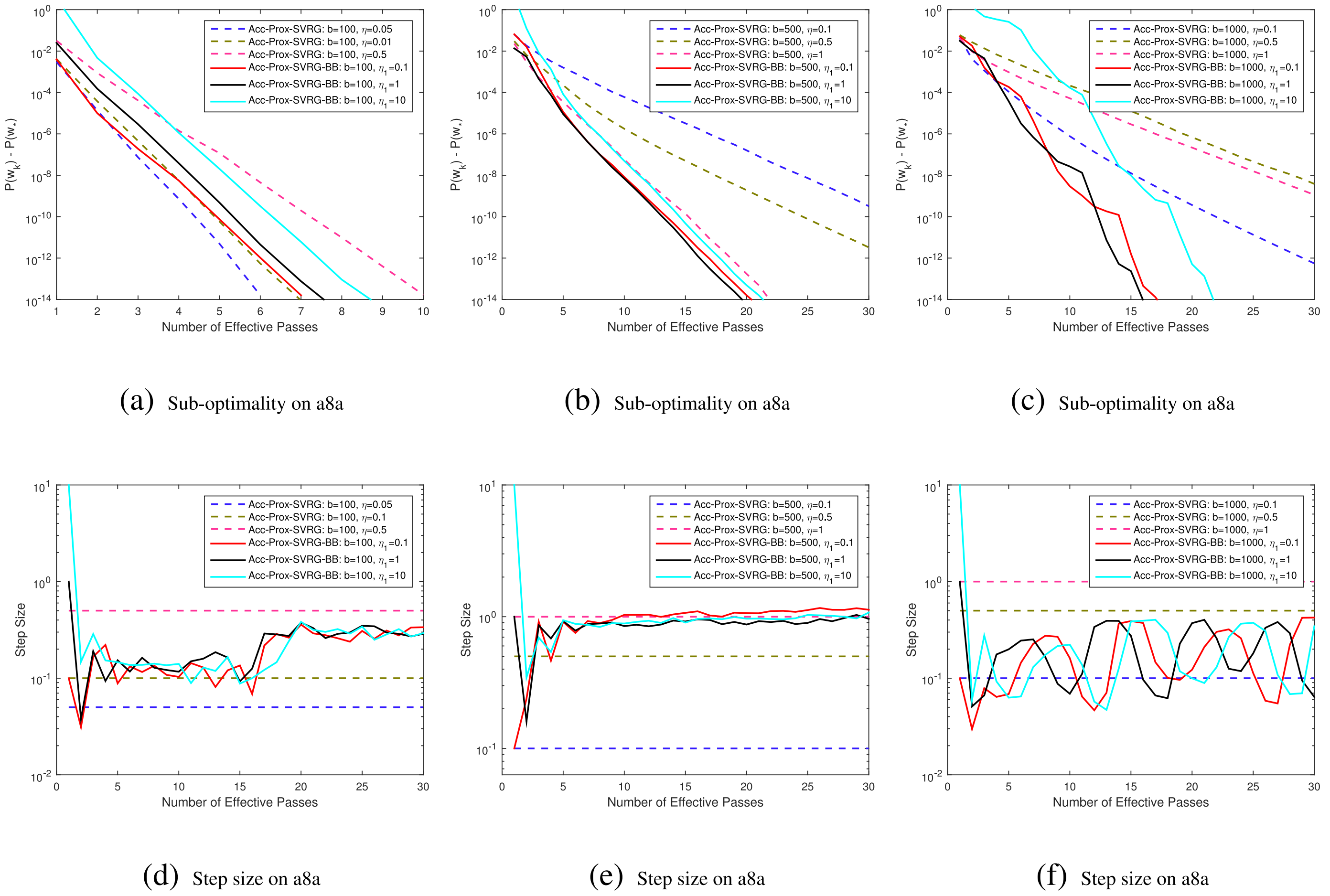
Lemma 4. *For $k\geq 1$ we have*

$$\langle\nabla F(v_k)+\xi_k,G_k+\xi_k\rangle=\frac{1}{2}(\|\nabla F(v_k)+\xi_k\|_2^2+\|G_k+\xi_k\|_2^2-\|\nabla F(v_k)-G_k\|_2^2) \tag{20}$$

$$\|G_k+\xi_k\|_2^2=2(\|\nabla F(v_k)+\xi_k\|_2^2+\|\nabla F(v_k)-G_k\|_2^2) \tag{21}$$

$$\|\nabla F(v_k)+\xi_k\|_2^2=2(\|G_k+\xi_k\|_2^2+\|\nabla F(v_k)-G_k\|_2^2) \tag{22}$$

In the following, we show the boundary of the modified stochastic gradient G_k:



(a) Sub-optimality on a8a

(b) Sub-optimality on a8a

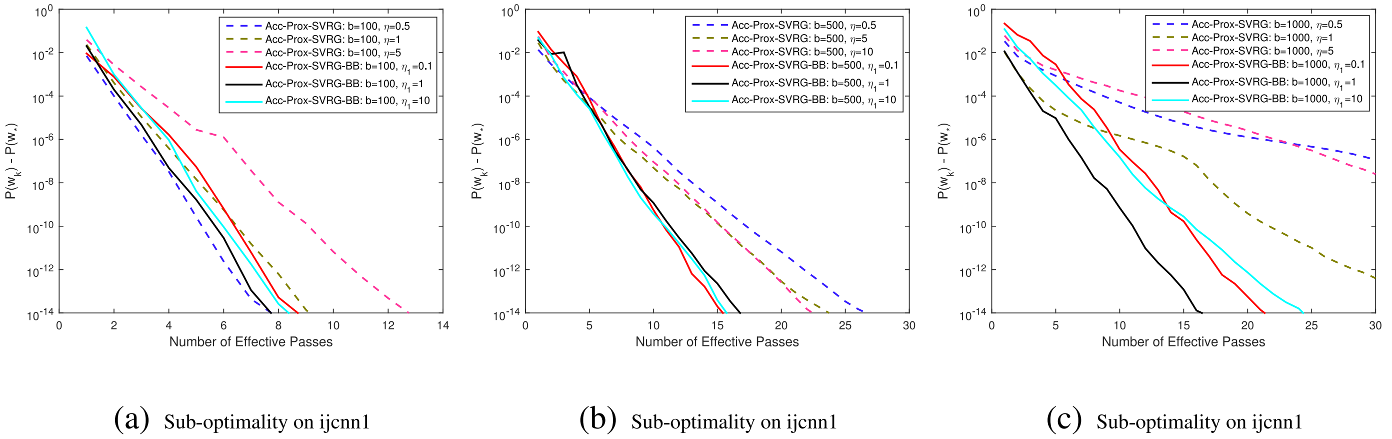
(c) Sub-optimality on a8a

(d) Step size on a8a

(e) Step size on a8a

(f) Step size on a8a

Fig. 2. Comparison of Acc-Prox-SVRG-BB and Acc-Prox-SVRG with fixed step sizes on a8a. The dashed lines for Acc-Prox-SVRG with fixed step size η_k given in the legend. The solid line stand for Acc-Prox-SVRG-BB with different η_0 given in the legend.



(a) Sub-optimality on ijcnn1

(b) Sub-optimality on ijcnn1

(c) Sub-optimality on ijcnn1

Fig. 3. Comparison of Acc-Prox-SVRG-BB and Acc-Prox-SVRG with regularization parameters $\lambda_1 = 10^{-4}$ and $\lambda_2 = 10^{-4}$ on *ijcnn*1. The dashed lines for Acc-Prox-SVRG with fixed step size η given in the legend. The solid line stand for Acc-Prox-SVRG-BB with different initial step size η_1 given in the legend.

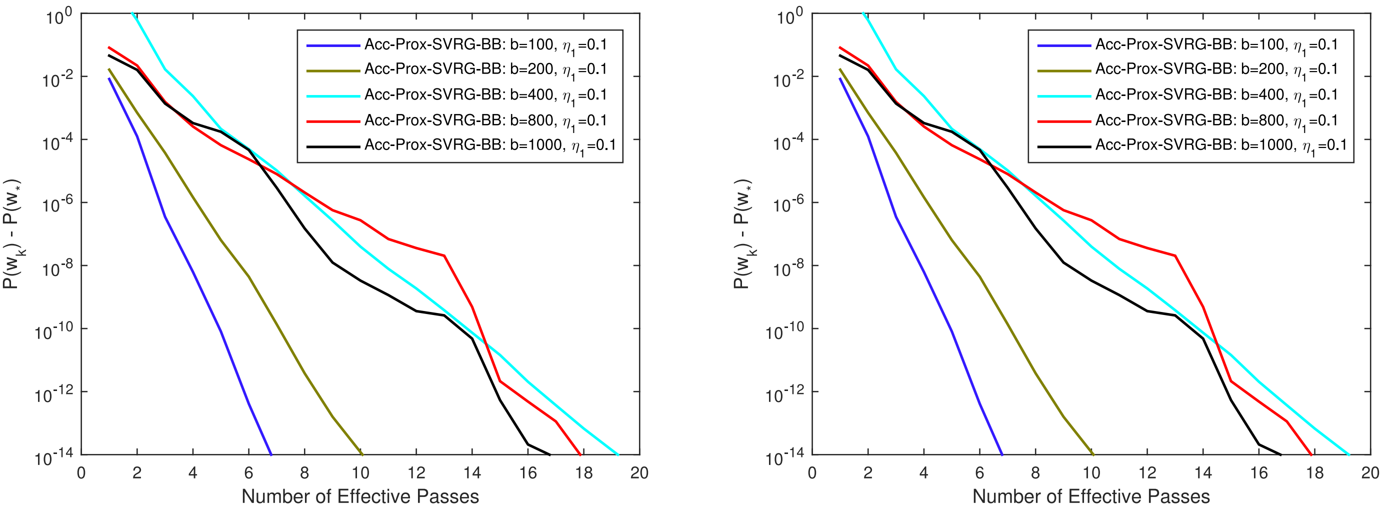
Lemma 5. *Suppose that Assumption 1 holds, and let* $w_* = \arg\min_{w\in\mathbb{R}^d} P(w)$. *If* G_k *is defined as in (11), then*

$$\mathbb{E}\|G_k - \nabla F(v_k)\|_2^2 \le \alpha(b)(2L^2\|v_k - w_k\|_2^2 + 8L(P(w_k) - P(w_*) + P(\widetilde{w}) - P(w_*))), \tag{23}$$

where $\alpha(b) = \frac{n-b}{b(n-1)}$.

The following theorem establishes convergence in expectation of Acc-Prox-SVRG-BB for non-smooth and strongly convex functions.

Theorem 1. *Suppose that Assumption 1, Assumption 2, and Assumption 3 hold. Let* $m \ge \max\left\{\frac{2\delta L}{C}\sqrt[3]{\frac{2L\alpha(b)^2}{\mu}}, \frac{2L\delta}{C}\right\}$ *and* $0 < \delta \le 1$.



(a) Sub-optimality on a8a

(b) Sub-optimality on ijcnn1

Fig. 4. Performance of Acc-Prox-SVRG-BB with different batch size (parameter b) on a8a (left) and ijcnn1 (right).

Then we have

$$
\begin{aligned}
&\mathbb{E}[P(\widetilde{w}_{s+1}) - P(w_*)] \\
&\leq \left(\left(1-\sqrt{\frac{\mu\delta}{Cm}}+\alpha(b)\cdot\frac{8\delta L}{Cm}\right)^m + \frac{1}{\frac{1}{8L}\sqrt{\frac{\mu Cm}{\delta}}-\alpha(b)}\right) \\
&\cdot\left(2+\frac{8L\delta}{\mu Cm}\right)(P(\widetilde{w}_s)-P(w_*)). \qquad (24)
\end{aligned}
$$

Based on Algorithm 1 and Theorem 1, we show the complexity of Acc-Prox-SVRG-BB. For clarity, we give a brief derivation of the complexity.

According to Theorem 1, we have $\mathbb{E}[P(\widetilde{w}_{s+1}) - P(w_*)] \leq \rho^s[P(\widetilde{w}_0) - P(w_*)]$, where

$$
\rho = \left(\left(1-\sqrt{\frac{\mu\delta}{Cm}}+\alpha(b)\cdot\frac{8\delta L}{Cm}\right)^m + \frac{1}{\frac{1}{8L}\sqrt{\frac{\mu Cm}{\delta}}-\alpha(b)}\right)\left(2+\frac{8L\delta}{\mu Cm}\right).
$$

Actually, by choosing m large enough, we can make parameter, ρ, arbitrarily small. Thereby, to satisfy $\mathbb{E}[P(\widetilde{w}_{s+1}) - P(w_*)] \leq \varepsilon$, the number of outer loops, s, must satisfy $s = O(\log(1/\varepsilon))$.

In addition, each stage requires $n+2m$ component gradient evaluations. From the proof of the second inequality of Lemma 2 (a.k.a. (17)) in Appendix A.1, we have

$$
\begin{aligned}
&\mathbb{E}_{I_k}[P(w_{k+1}) - \Phi^*_{k+1}] \\
&\overset{(20)}{\leq} \mathbb{E}_{I_k}\Bigg[\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)(-\Phi^*_k + P(w_k)) - \frac{\mu}{2}\left(1-\frac{\mu\delta}{Cm}\right) \\
&\cdot\sqrt{\frac{Cm}{\mu\delta}}\|w_k - v_k\|_2^2 - \frac{\delta}{2Cm}\|\nabla F(v_k)+\xi_k\|_2^2 \\
&+ \frac{\delta}{2Cm}\frac{L\delta}{Cm}\|G_k+\xi_k\|_2^2 + \frac{\delta}{2Cm}\|\nabla F(v_k) - G_k\|_2^2\Bigg] \\
&\overset{(21), m>\frac{2L\delta}{C}}{\leq} \mathbb{E}_{I_k}\Bigg[\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)(-\Phi^*_k + P(w_k)) \\
&- \frac{\mu}{2}\left(1-\frac{\mu\delta}{Cm}\right)\sqrt{\frac{Cm}{\mu\delta}}\|w_k - v_k\|_2^2 + \frac{\delta}{Cm}\|\nabla F(v_k) - G_k\|_2^2\Bigg].
\end{aligned}
$$

Table 1
Comparison of the overall complexity suitable for solving Eq. (1).

Algorithm	Complexity	Fixed Step Size?
FISTA	$O(n\sqrt{\kappa}\log(1/\varepsilon))$	No
SAG	$O((n+\kappa)\log(1/\varepsilon)))$	Yes
SDCA	$O((n+\kappa)\log(1/\varepsilon)))$	Yes
SVRG/Prox-SVRG	$O((n+\kappa)\log(1/\varepsilon)))$	Yes
Acc-Prox-SVRG	$O(n+\min\{\kappa, n\sqrt{\kappa}\})\log(1/\varepsilon))$	Yes
Acc-Prox-SVRG-BB	$O((n+\max\{\sqrt[3]{\kappa\alpha(b)^2}, L\delta/C\})\log(1/\varepsilon))$	No

To ensure that the last inequality is true, we require $\frac{\delta}{Cm} \leq \frac{1}{2L}$. Thereby, $m \geq \frac{2L\delta}{C}$.

Moreover, from the beginning of proof of Theorem 1 in Appendix A.2,

$$
\begin{aligned}
&\mathbb{E}[P(w_k) - P(w_*)] \\
&\leq \left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)^{k-1}(\Phi_1 - P)(w_*) + \mathbb{E}\Bigg[\sum_{l=1}^{k-1}\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)^{k-1-l} \\
&\Bigg\{\left(-\frac{\mu}{2}\frac{1-\mu\delta/Cm}{\sqrt{\mu\delta/Cm}}+\alpha(b)\cdot\frac{2\delta L^2}{Cm}\right)\|v_l - w_l\|_2^2 + \alpha(b) \\
&\cdot\frac{8\delta L}{Cm}(P(w_k) - P(w_*) + P(\widetilde{w}) - P(w_*))\Bigg\}\Bigg]
\end{aligned}
$$

To omit the term, $\left(-\frac{\mu}{2}\frac{1-\mu\delta/Cm}{\sqrt{\mu\delta/Cm}}+\alpha(b)\cdot\frac{2\delta L^2}{Cm}\right)\|v_l - w_l\|_2^2$, we set $-\frac{\mu}{2}\frac{1-\mu\delta/Cm}{\sqrt{\mu\delta/Cm}}+\alpha(b)\cdot\frac{2\delta L^2}{Cm} \leq 0$. Thereby, $m \geq \frac{2\delta L}{C}\sqrt[3]{\frac{2L\alpha(b)^2}{\mu}}$.

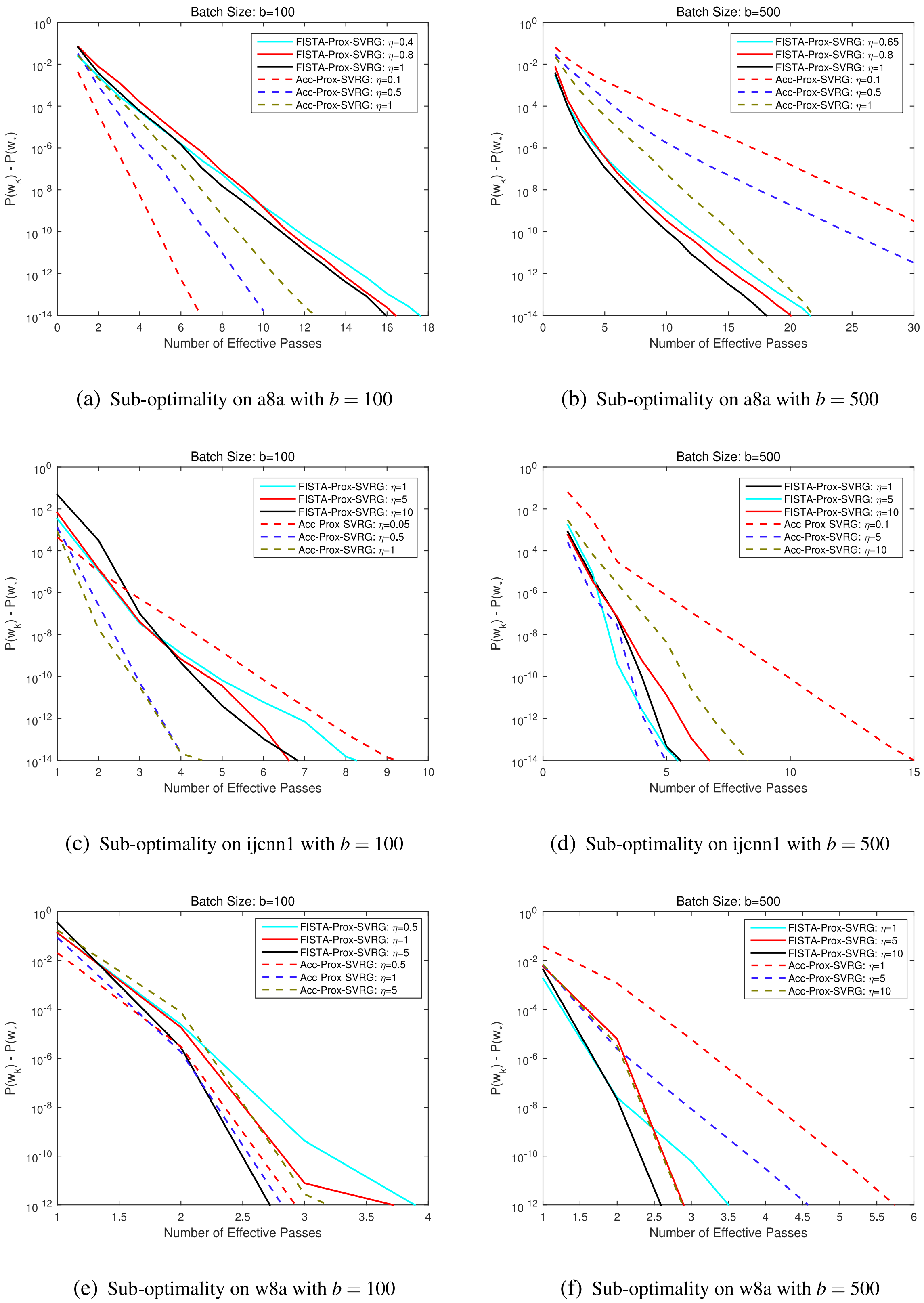
Hence, we ascertain that Acc-Prox-SVRG-BB achieves the following overall complexity (total number of component gradient evaluations to find an ε-accurate solution):

$$
O\left(\left(n+\max\left\{\sqrt[3]{\kappa\alpha(b)^2}, L\delta/C\right\}\right)(\log(1/\varepsilon))\right), \qquad (25)
$$

which recovers the faster rate as Prox-SVRG, Prox-SAG, etc.

For convenience, we show, in the Table 1, the complexities of the state-of-the-art methods.

Note that, in general, $n+\kappa \ll n\sqrt{\kappa}$, which means that state-of-the-art stochastic gradient methods converge at a faster rate than do the classical deterministic methods.



(a) Sub-optimality on a8a with $b = 100$

(b) Sub-optimality on a8a with $b = 500$

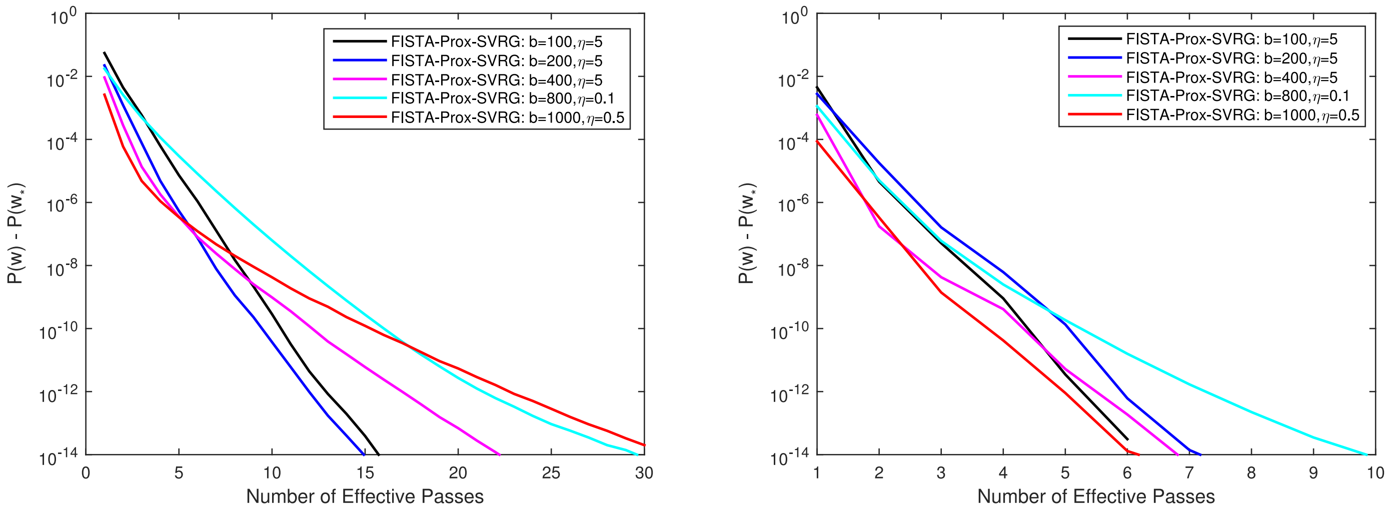
(c) Sub-optimality on ijcnn1 with $b = 100$

(d) Sub-optimality on ijcnn1 with $b = 500$

(e) Sub-optimality on w8a with $b = 100$

(f) Sub-optimality on w8a with $b = 500$

Fig. 5. Comparison of FISTA-Prox-SVRG and Acc-Prox-SVRG on *a8a* (top), *ijcnn*1 (middle) and *w8a*(bottom). The dashed lines correspond to Acc-Prox-SVRG. The solid lines stand for FISTA-Prox-SVRG.



(a) Sub-optimality on a8a

(b) Sub-optimality on ijcnn1

Fig. 6. Performance of FISTA-Prox-SVRG with different batch size (parameter b) on a8a (left) and ijcnn1 (right).

5. FISTA-Prox-SVRG Algorithm

From Algorithm 1, we see that the parameter, β_k, must be chosen in Acc-Prox-SVRG-BB. In general, the parameter, β_k, depends on Lipschitz constant L and strong convexity parameter μ. In the case of an unknown Lipschitz constant, we estimate the optimal step size by using backtracking [8], however, estimating the strong convexity parameter is much more challenging. Influenced by this gap, we incorporate FISTA and Prox-SVRG in the mini-batch setting, thereby generating another new ASGD method, FISTA-Prox-SVRG. Before presenting FISTA-Prox-SVRG, we first give a brief introduction about the FISTA iteration scheme.

For any initial point $v_1 = w_1 \in \mathbb{R}^d$ and $t_1 = 1$, FISTA consists of the following steps for solving problem (1):

$$v_{k+1} = w_{k+1} + \frac{t_k - 1}{t_{k+1}}(w_{k+1} - w_k), \tag{26}$$

$$w_{k+1} = \text{prox}_{\eta R}(v_k - \eta G_k), \tag{27}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}. \tag{28}$$

Beck et al. [8] pointed out that there are two main differences between FISTA and the Nesterov's acceleration method [10]. Actually, comparing the FISTA iteration scheme and the Nesterov's acceleration scheme in the inner iteration of Algorithm 1, we clearly see that FISTA replaces parameter β_k with $\frac{t_k-1}{t_{k+1}}$, where the parameter t_k is easy to access as shown in the above iteration shceme.

We now describe the FISTA-Prox-SVRG method in Algorithm 2.

From Algorithm 2, we see that a constant step size was used in FISTA-Prox-SVRG.

To conclude this section, we apply the BB method to FISTA-Prox-SVRG. For convenience, we call the new method FISTA-Prox-SVRG-BB. Here, we do not show the details of FISTA-Prox-SVRG-BB, because, they can easily be obtained by replacing a constant step size in Algorithm 2 with Eq. (12), which is similar to Acc-Prox-SVRG-BB (Algorithm 1).

Algorithm 2 FISTA-Prox-SVRG.

```
Input: update sequence $m$; step size $\eta$; mini-batch size $b \in [n]$; initial point $\tilde{w}_1$
for $s = 1, 2, \ldots$ do
    Set $\tilde{w} = \tilde{w}_s$, $t_1 = 1$
    Compute and store $g_s = (1/n)\sum_{i=1}^{n} \nabla f_i(\tilde{w})$
    Set $v_1 = w_1 = \tilde{w}$
    for $k = 1, \ldots, m$ do
        Randomly pick subset $I_k \in \{1, \ldots, n\}$ of size $b$
        Compute a stochastic estimate of $\nabla F(w_k)$;
            $G_k = \nabla F_{I_k}(v_k) - \nabla F_{I_k}(\tilde{w}) + g_s$
        $w_{k+1} = prox_{\eta R}(v_k - \eta G_k)$
        $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$.
        $v_{k+1} = w_{k+1} + \frac{t_k-1}{t_{k+1}}(w_{k+1} - w_k)$
    end for
    Set $\tilde{w}_{s+1} = w_{m+1}$
end for
```

6. Numerical experiments

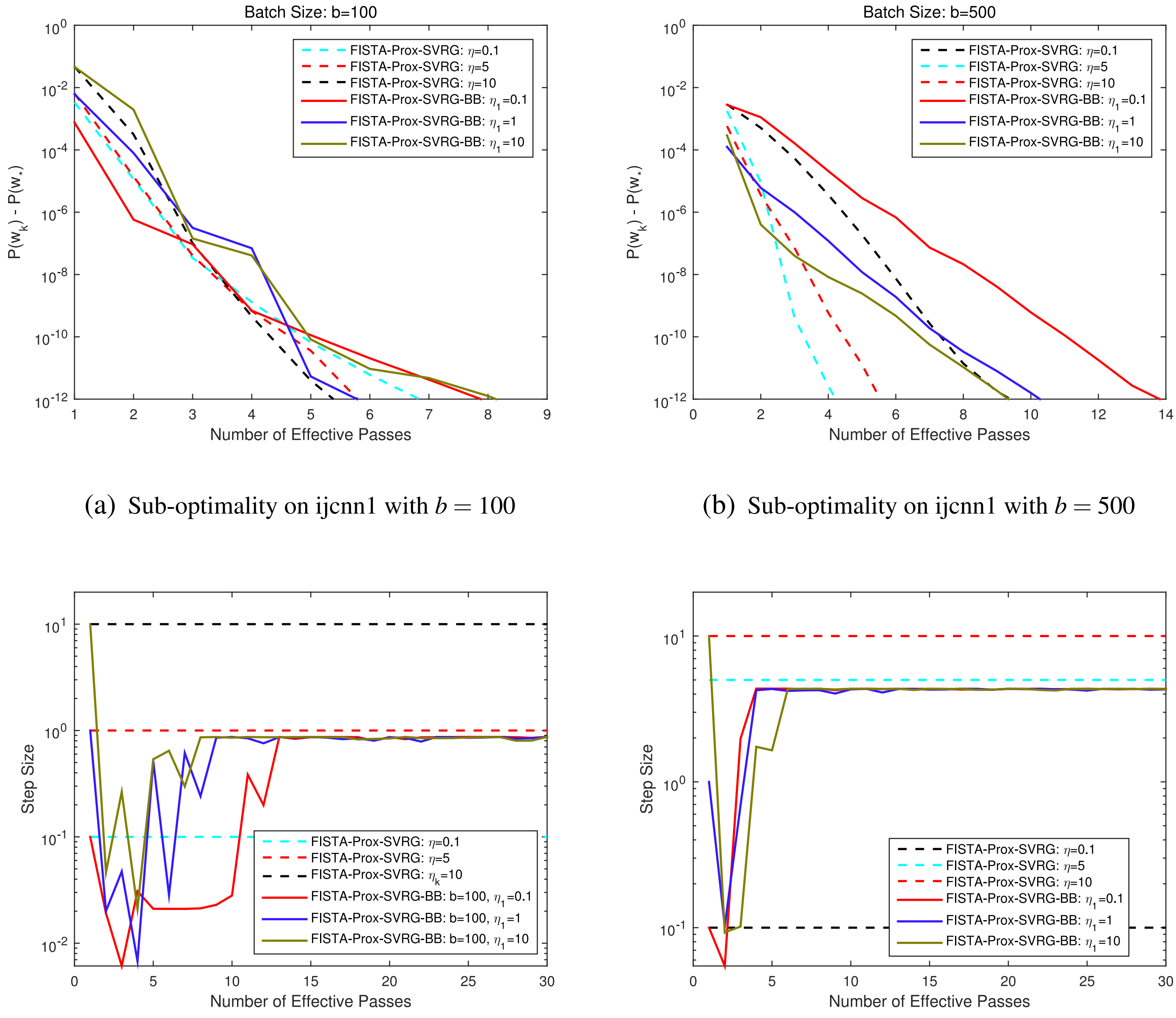
In this section, we discuss the numerical experiments that were conducted to illustrate the properties and performance of Acc-Prox-SVRG-BB, FISTA-Prox-SVRG and FISTA-Prox-SVRG-BB.

In sub-Section 6.1, we study the properties of Acc-Prox-SVRG-BB. In sub-Section 6.2 and sub-Section 6.3, we show the performance of FISTA-Prox-SVRG and FISTA-Prox-SVRG-BB, respectively. Finally, in sub-Section 6.4, we compare Acc-Prox-SVRG-BB, FISTA-Prox-SVRG and FISTA-Prox-SVRG-BB with other related methods.

We discuss the experiments that were performed with $R(x) = \frac{\lambda_2}{2}\|w\|_2^2 + \lambda_1\|w\|_1$ and the logistic loss function f_i given as follows:

$$f_i(w) = \log(1 + \exp(-b_i a_i^T w)). \tag{29}$$

These functions are usually used in machine learning, with $(a_i, b_i) \in \mathbb{R}^d \times \{+1, -1\}$, $i = 1, \ldots, n$, being a training data set of example-label pairs. The resulting optimization problem (1) takes



(a) Sub-optimality on ijcnn1 with $b = 100$

(b) Sub-optimality on ijcnn1 with $b = 500$

(c) Step size on ijcnn1 with $b = 100$

(d) Step size on ijcnn1 with $b = 500$

Fig. 7. The performance of FISTA-Prox-SVRG-BB on ijcnn1 with batch size $b = 100$ or 500.

the form

$$\min_{w\in\mathbb{R}^d} \quad P(w) := \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-b_i a_i^T w)) + \frac{\lambda_2}{2}\|w\|_2^2 + \lambda_1\|w\|_1, \tag{30}$$

and is employed in machine learning for binary classification.

Three publicly available data sets (*a8a, w8a* and *ijcnn1*)[2] were used in the experiments. Detailed information for the data sets is given in Table 2.

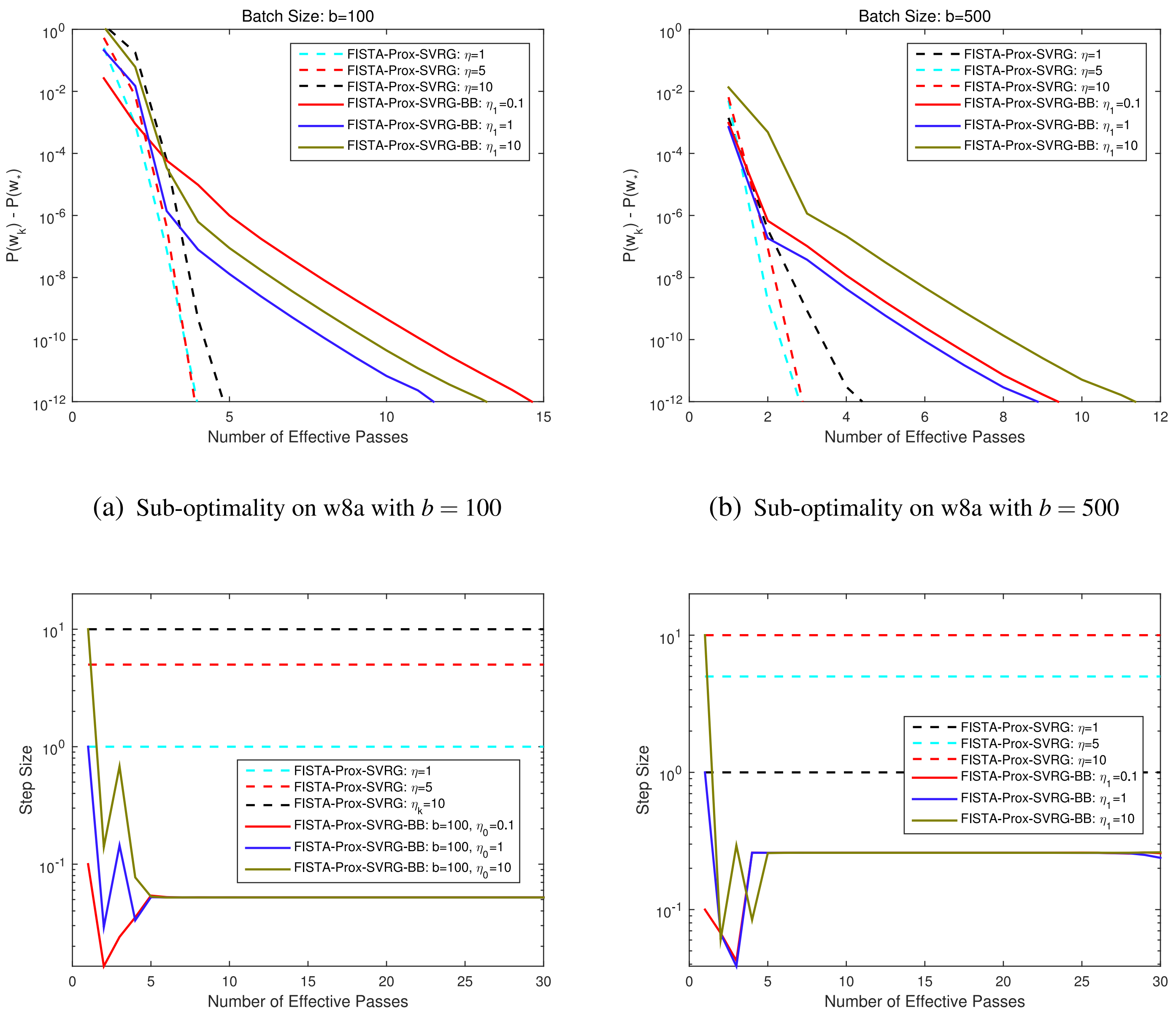
Table 2
Details of the data sets in our experiments.

Dataset	n	d	λ_1	λ_2
a8a	22,696	123	10^{-5}	10^{-2}
w8a	49,749	300	10^{-2}	10^{-4}
ijcnn1	49,990	22	10^{-2}	10^{-4}

6.1. Performance of acc-Prox-SVRG-BB

In this section, we show the performance of Acc-Prox-SVRG-BB conducted using *a8a* and *ijcnn1*. Figs. 1 and 2 display the compared results between Acc-Prox-SVRG-BB and Acc-Prox-SVRG. In Figs. 1 and 2, the *x*-axis represents the number of effective passes over the data, where each effective pass evaluates *n* component gradients. In Fig. 3(a) through Fig. 3(c) and Fig. 2(a) through

[2] *a8a, w8a* and *ijcnn1* are available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.



(a) Sub-optimality on w8a with $b = 100$

(b) Sub-optimality on w8a with $b = 500$

(c) Step size on w8a with $b = 100$

(d) Step size on w8a with $b = 500$

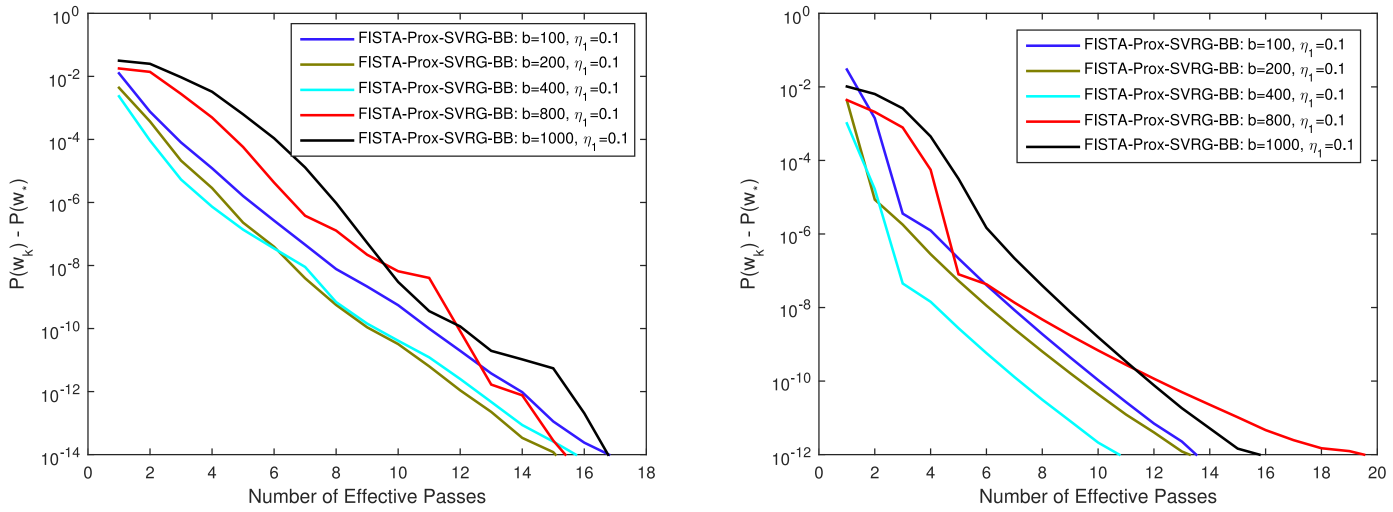
Fig. 8. The performance of FISTA-Prox-SVRG-BB on w8a with batch size $b = 100$ or 500.

Fig. 2(c), the y-axis denotes the sub-optimality: $P(w_k) - P(w_*)$. Here, w_* is obtained by running Acc-Prox-SVRG with the best-tuned step size until it converges. In Fig. 1(d) through Fig. 1(f) and Fig. 2(d) through Fig. 2(f), the y-axis denotes the corresponding step sizes η_k. Moreover, in Figs. 1 and 2, the dashed lines represent Acc-Prox-SVRG with fixed step size given in the legends of the figures. The solid lines correspond to Acc-Prox-SVRG-BB with different initial step size η_1. Finally, we show the performance of Acc-Prox-SVRG-BB with different batch size in Fig. 4.

As suggested in [19], we set $m = \tau b$ ($\tau \in$ {0.1, 1.0, 10}) and $\beta_k = \frac{b-2}{b+2}$ varying b in the set {100, 500, 1000}. Also, in all our experiments, we ran Acc-Prox-SVRG using values of η selected from the range {0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0}. In addition, for Acc-Prox-SVRG-BB, we selected the parameter, m, from (0,1]. Actually, if the parameter, m, is not large, then we set $\delta = 1$; otherwise, we set $\delta = 0.1$.

As shown in Fig. 3(a) through Fig. 3(c) and Fig. 2(a) through Fig. 2(c), Acc-Prox-SVRG-BB always achieves the same level of sub-optimality as Acc-Prox-SVRG with the best tuned step size. Even Acc-Prox-SVRG-BB achieves better performance than Acc-Prox-SVRG with the best tuned step size. Moreover, as seen in Fig. 1(d) through Fig. 1(f) and Fig. 2(d) through Fig. 2(f), the step size of Acc-Prox-SVRG-BB gradually converges to a constant step size.

In Fig. 3(a) through Fig. 3(c), we see that Acc-Prox-SVRG requires fewer number of effective passes than Acc-Prox-SVRG-BB to obtain optimal solution of Problem (1). Actually, our proposed methods and the existing methods (Acc-Prox-SVRG, Prox-SVRG and mS2GD etc.) were often considered for convex problems. The performance of these methods is influenced by the regularization terms. For example, when setting regularization parameters, $\lambda_1 = 10^{-4}$ and $\lambda_2 = 10^{-4}$, rather than the parameters $\lambda_1 = 10^{-2}$ and



(a) Sub-optimality on a8a

(b) Sub-optimality on w8a

Fig. 9. Performance of FISTA-Prox-SVRG-BB with different batch size (parameter b) on a8a (left) and w8a (right).

$\lambda_2 = 10^{-4}$ for ijcnn1, the performance of Acc-Prox-SVRG-BB and Acc-Prox-SVRG are shown in Fig. 3.

As seen in Fig. 3, under $\lambda_1 = 10^{-4}$ and $\lambda_2 = 10^{-4}$, our Acc-Prox-SVRG-BB method performs better than does Acc-Prox-SVRG.

Our motivation is to provide an easy way to compute step size for the Accelerated Stochastic Gradient Descent (ASGD) methods. With a different background, our proposed method achieves results comparable to the other state-of-the-art methods.

At the end of this section, the results of choosing different batch size, b, for Acc-Prox-SVRG-BB are shown in Fig. 4

In 4(a), it is seen that Acc-Prox-SVRG-BB performs better with a small batch size. In 4(b), it is seen that, even with a large batch size, Acc-Prox-SVRG-BB also performs better. We think that BB step size greatly improves the performance of the original Acc-Prox-SVRG method and makes the new method perform better with any mini-batch size. Hence, we conclude that using BB step size in Acc-Prox-SVRG makes it easier to choose the parameters.

6.2. Performance of FISTA-Prox-SVRG

In this section, to confirm the effectiveness of FISTA-Prox-SVRG, we conducted experiments using *a8a, w8a* and *ijcnn*1. Especially, we show the comparison results between FISTA-Prox-SVRG and Acc-Prox-SVRG. We ran FISTA-Prox-SVRG using the values of η selected from the range [0.01,10]. We chose the three best step sizes, η, in each mini-batch setting as shown in the legends of Fig. 5. The parameters for Acc-Prox-SVRG are determined in the same manner as in the above sub-section. Also, we show the performance of FISTA-Prox-SVRG with different batch size in Fig. 6.

Fig. 5 shows that FISTA-Prox-SVRG achieves the same level of sub-optimality as Acc-Prox-SVRG, and even can achieve better performance than Acc-Prox-SVRG. It seems that FISTA-Prox-SVRG is not always achieving better performance than Acc-Prox-SVRG. Because step size is manually selected when running FISTA-Prox-SVRG and Acc-Prox-SVRG, as much as possible, we choose only the best step size. Step size greatly affects the performance of FISTA-Prox-SVRG and Acc-Prox-SVRG. We cannot obtain, by hand, the so called best step size for two algorithms. Therefore, we try our best to choose the step size that makes the two algorithms perform better. Of course, there is no theoretical guarantee that the step size is the best. Moreover, although we cannot guarantee achieving the best performance for each data set, we achieve better performance on most data sets.

Actually, the reason for proposing FISTA-Prox-SVRG is to further reduce the difficulty of choosing parameters in Acc-Prox-SVRG. To further reduce the human factors, we propose using the BB step size in FISTA-Prox-SVRG, thereby obtaining FISTA-Prox-SVRG-BB.

Here, we show the performance of FISTA-Prox-SVRG with different batch size, b in Fig. 6. We ran FISTA-Prox-SVRG using values of η in the range {0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0}; we choose the best η in each mini-batch setting.

Fig. 6 shows that, with an appropriate step size, FISTA-Prox-SVRG performs better with any mini-batch size.

6.3. Performance of FISTA-Prox-SVRG-BB

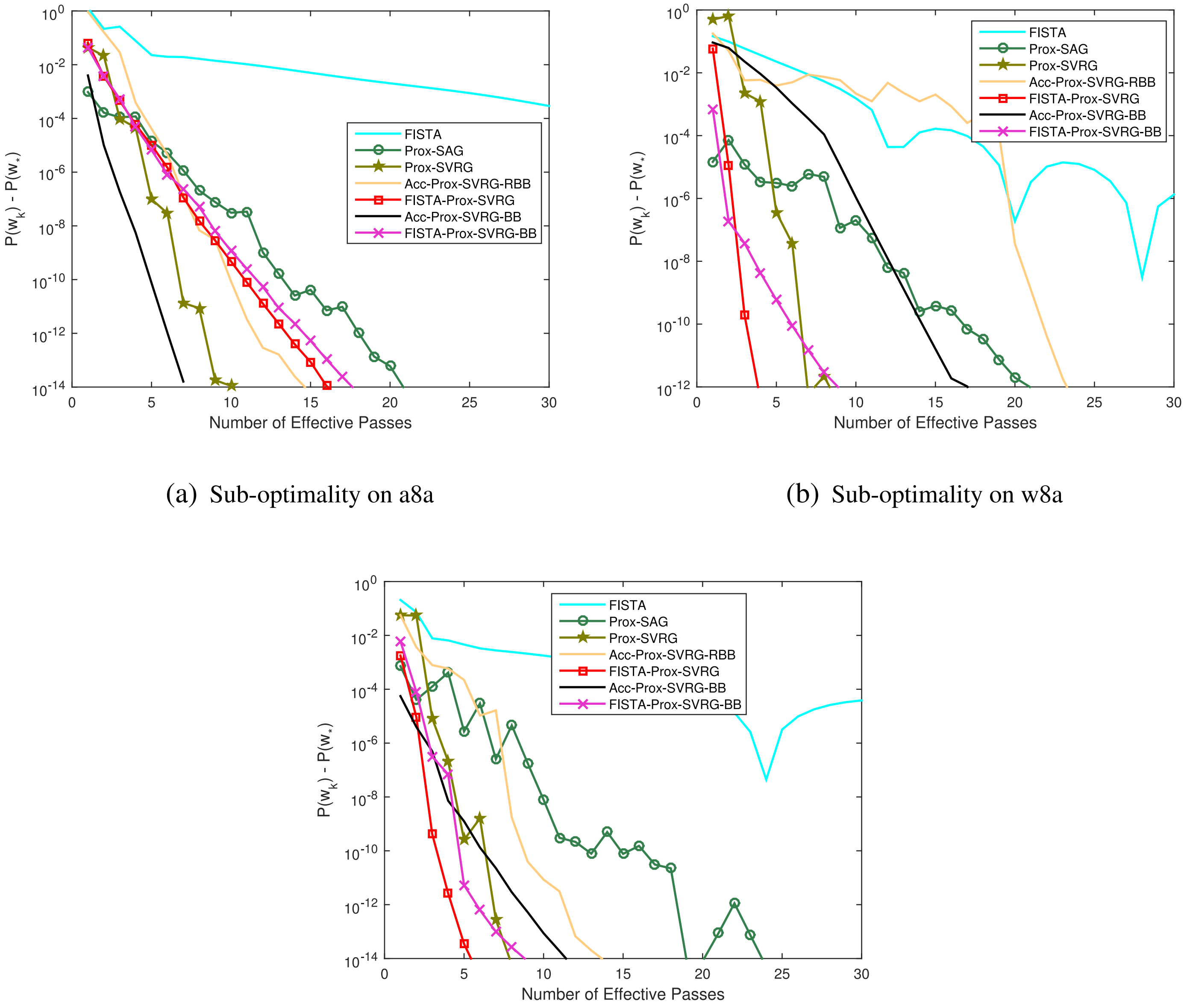
To further validate the effectiveness of the BB method, we introduce the BB method into FISTA-Prox-SVRG and show the results in Figs. 7 and 8. The experiments were conducted using *ijcnn*1 and *w8a*.The dashed lines represent FISTA-Prox-SVRG. The solid lines represent FISTA-Prox-SVRG-BB. We ran FISTA-Prox-SVRG-BB with three different initial step sizes (the details are shown in the legends of Figs. 7 and 8). The parameters for FISTA-Prox-SVRG are determined in the same manner as in the above section. Also, we use the same manner as sub-Section 6.1 to define the parameter δ. Finally, we show the performance of FISTA-Prox-SVRG-BB with different batch size in Fig. 9.

As seen from Fig. 7(a) and (b), Fig. 8(a) and (b), FISTA-Prox-SVRG-BB always achieves the same level of sub-optimality as FISTA-Prox-SVRG. Also, in Fig. 7(c) and (d), Fig. 8(c) and (d), the step sizes of FISTA-Prox-SVRG-BB converge to a constant after 5 or 10 iteration steps.

Also, we show the performance of FISTA-Prox-SVRG-BB with different batch size in Fig. 9. We ran FISTA-Prox-SVRG-BB with initial step size, $\eta_1 = 0.1$.

Fig. 9(a) shows that FISTA-Prox-SVRG-BB performs better with any mini-batch size. Fig. 9(b) also implies that FISTA-Prox-SVRG-BB performs better with any mini-batch size.

Both Figs. 4 and 9 indicate that when the new method, which combines the ASGD method and BB step size, performs better under most mini-batch size situations.



(a) Sub-optimality on a8a

(b) Sub-optimality on w8a

(c) Step size on ijcnn1

Fig. 10. Comparison of different methods on *a8a*, *w8a* and *ijcnn1*.

6.4. Comparison with related methods

In this section, we compare our proposed methods (Acc-Prox-SVRG-BB, FISTA-Prox-SVRG and FISTA-Prox-SVRG-BB) with the following methods:

(a) **FISTA:** FISTA is a version of the fast iterative shrinkage-thresholding algorithms [8] for solving linear inverse problems. In FISTA, a backtracking step size is employed.
(b) **Prox-SAG:** Prox-SAG is a proximal version of the SAG method [31]. In Prox-SAG, as suggested in [46], we also used the standard backtracking line search to obtain the step size.
(c) **Prox-SVRG:** Prox-SVRG is a proximal version of the stochastic variance-reduction gradient method. We ran Prox-SVRG with a constant step size. In addition, as suggested in [20], we employed $m = 2n$ between full gradient evaluations for Prox-SVRG.
(d) **Acc-Prox-SVRG-RBB:** The Acc-Prox-SVRG method incorporates the RBB method into Acc-Prox-SVRG. In addition, for Acc-Prox-SVRG-BB, we chose the best batch size b_1 and b_2 on *a8a, w8a* and *ijcnn1*.

Fig. 10 shows that on the three different data sets, our proposed methods achieve better performance than the competing methods.

7. Conclusion

This paper considered a shortcoming associated with the ASGD methods for step size choice. Specifically, there is little practical guidance provided for good choices. We proposed using the BB method to automatically compute step size for the most advanced method, Acc-Prox-SVRG and obtained a new method: Acc-Prox-SVRG-BB. We provided a detailed convergence analysis, showing that Acc-Prox-SVRG-BB achieves the same level of complexity as the best known stochastic gradient-based methods. In addition, to further reduce the difficulty of the parameters choices of Acc-Prox-SVRG, we incorporated FISTA and Prox-SVRG in a mini-batch setting and proposed a new method, FISTA-Prox-SVRG. Finally, to further show the effectiveness of the BB method, we introduced it into FISTA-Prox-SVRG. Numerical results show that our proposed methods achieve better performance than the competing methods.

Conflict of interest

None.

Acknowledgments

This work was supported by the Natural Science Foundation of China under Grant U1605254, and the funding from Fujian Research Institute on Urban Traffic Big Data.

Appendix A

A1. Proof of lemmas

Now, we give the proof of Lemma 1, Lemma 2 and Lemma 3.
Proof of Lemma 1.

Proof. From the definition of η_s in Algorithm 1 and combining the convexity of objective function (Assumption 2), we have

$$\eta_s = \frac{\delta}{m}\frac{\|\widetilde{w}_s - \widetilde{w}_{s-1}\|_2^2}{(\widetilde{w}_s - \widetilde{w}_{s-1})^T(g_s^{'} - g_{s-1}^{'})}$$
$$\leq \frac{\delta}{m}\frac{\|\widetilde{w}_s - \widetilde{w}_{s-1}\|_2^2}{\mu\|\widetilde{w}_s - \widetilde{w}_{s-1}\|_2^2} = \frac{\delta}{\mu m}$$

In addition, using the Lipstchitz properties of objective function, we have

$$\eta_s = \frac{\delta}{m}\frac{\|\widetilde{w}_s - \widetilde{w}_{s-1}\|_2^2}{(\widetilde{w}_s - \widetilde{w}_{s-1})^T(g_s^{'} - g_{s-1}^{'})}$$
$$\geq \frac{\delta}{m}\frac{\|\widetilde{w}_s - \widetilde{w}_{s-1}\|_2^2}{L\|\widetilde{w}_s - \widetilde{w}_{s-1}\|_2^2} = \frac{\delta}{\mathrm{Lm}}$$

where in the first inequality we also use the Cauchy-Schwartz inequality. □

Proof of Lemma 2.

Proof. We first prove (16) of Lemma 2 by induction. It is true for $k = 1$. We assume that it is true for k, then from the definition of estimate sequence, we have

$$\mathbb{E}[\Phi_{k+1}(w)] = \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\mathbb{E}[\Phi_k(w)] + \sqrt{\frac{\mu\delta}{Cm}}\mathbb{E}\Big[F_{I_k}(v_k)$$
$$+ (G_k, w - v_k) + \frac{\mu}{2}\|w - v_k\|^2 + R(w_{k+1}) + (\xi_k, w - w_{k+1})\Big]$$
$$\leq \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\left[P(w) + \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)^{k-1}(\Phi_1 - P)(w)\right]$$
$$+ \sqrt{\frac{\mu\delta}{Cm}}\mathbb{E}\Big[F_{I_k}(v_k) + (G_k, w - v_k) + \frac{\mu}{2}\|w - v_k\|^2 + R(w_{k+1})$$
$$+ (\xi_k, w - w_{k+1})\Big]$$
$$= \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)P(w) + \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)^k(\Phi_1 - P)(w)$$
$$+ \sqrt{\frac{\mu\delta}{Cm}}\Big(F(v_k) + (\nabla F(v_k), w - v_k) + \frac{\mu}{2}\|w - v_k\|^2 + R(w_{k+1})$$
$$+ (\xi_k, w - w_{k+1})\Big)$$
$$\leq \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)P(w) + \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)^k(\Phi_1 - P)(w)$$
$$+ \sqrt{\frac{\mu\delta}{Cm}}(F(w) + R(w))$$
$$= P(w) + \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)^k(\Phi_1 - P)(w),$$

where in the first inequality we used induction hypothesis and in the last equality we used the convexity of $F(w)$ and $R(w)$. In the following, we will finish the proof of (17) of Lemma 2 □

Proof of Lemma 3.

Proof. From the definition of estimate sequence and (1), we have that for any $k \geq 1$

$$\Phi_{k+1}^* + \frac{\mu}{2}\|w - z_{k+1}\|^2 = \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\left(\frac{\mu}{2}\|w - z_k\|^2 + \Phi_k^*\right)$$
$$+ \sqrt{\frac{\mu\delta}{Cm}}\Big(\nabla F_{I_k}(v_k) + (G_k, w - v_k) + \frac{\mu}{2}\|w - v_k\|^2 + R(w_{k+1})$$
$$+ (\xi_k, w - w_{k+1})\Big)$$

By differentiating this equality at v_k, we have

$$\mu(v_k - z_{k+1}) = \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\mu(v_k - z_k) + \sqrt{\frac{\mu\delta}{Cm}}(G_k + \xi_k).$$

Hence, we have

$$z_{k+1} = \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)z_k + \sqrt{\frac{\mu\delta}{Cm}}v_k - \sqrt{\frac{\delta}{\mu Cm}}(G_k + \xi_k)$$

Here, we finish the proof of (18) of Lemma 3. Next, we prove (19) of Lemma 3 by induction. It is true for $k = 1$. If we assume it is true for k, then according to (18) we have

$$z_{k+1} - v_{k+1} = \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)z_k + \sqrt{\frac{\mu\delta}{Cm}}v_k$$
$$- \sqrt{\frac{\delta}{\mu Cm}}(G_k + \xi_k) - v_{k+1}$$
$$\overset{(19)}{=} \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\left(v_k + \sqrt{\frac{Cm}{\mu\delta}}(v_k - w_k)\right) + \sqrt{\frac{\mu\delta}{Cm}}v_k$$
$$- \sqrt{\frac{\delta}{\mu Cm}}(G_k + \xi_k) - v_{k+1}$$
$$= \sqrt{\frac{Cm}{\mu\delta}}v_k - \sqrt{\frac{\delta}{\mu Cm}}(G_k + \xi_k) - \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\sqrt{\frac{Cm}{\mu\delta}}w_k - v_{k+1}$$
$$= \sqrt{\frac{Cm}{\mu\delta}}\left(v_k - \frac{\delta}{Cm}(G_k + \xi_k)\right) - \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\sqrt{\frac{Cm}{\mu\delta}}w_k - v_{k+1}$$
$$= \sqrt{\frac{Cm}{\mu\delta}}w_{k+1} - \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\sqrt{\frac{Cm}{\mu\delta}}w_k - v_{k+1}$$

From the update rule of v_{k+1} computed in Algorithm 1, we have

$$-\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\sqrt{\frac{Cm}{\mu\delta}}w_k = -\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\sqrt{\frac{Cm}{\mu\delta}}\left[w_{k+1}-\frac{1}{\beta_k}(v_{k+1}-w_{k+1})\right]$$
$$= (-2)\sqrt{\frac{Cm}{\mu\delta}}w_{k+1}+\left(1+\sqrt{\frac{Cm}{\mu\delta}}\right)v_{k+1},$$

where we set $\beta_k = \frac{\sqrt{Cm}-\sqrt{\mu\delta}}{\sqrt{Cm}+\sqrt{\mu\delta}}$. Hence, we obtain

$$z_{k+1}-v_{k+1} = \sqrt{\frac{Cm}{\mu\delta}}(v_{k+1}-w_{k+1})$$

□

Finally based on Lemma 3 and Lemma 4, we prove (17) of Lemma 2 by induction.

Proof. From the definition of Φ_1, $\Phi_1^* = P(w_1)$. If we assume (17) is true for k, then using (18), we obtain

$$\|v_k - z_{k+1}\|_2^2 = \left\| v_k - \left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)z_k - \sqrt{\frac{\mu\delta}{Cm}}v_k + \sqrt{\frac{\delta}{\mu Cm}}(G_k+\xi_k)\right\|_2^2$$
$$= \left\|\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)(v_k - z_k) + \sqrt{\frac{\delta}{\mu Cm}}(G_k+\xi_k)\right\|_2^2$$
$$= \left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)^2\|v_k - z_k\|_2^2 + 2\sqrt{\frac{\delta}{\mu Cm}}\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)(v_k - z_k, G_k+\xi_k)$$
$$+\frac{\delta}{\mu Cm}\|G_k+\xi_k\|_2^2$$

From the above equation and (15) with $w = v_k$, we have

$$\Phi_{k+1}(v_k) = \Phi_{k+1}^* + \frac{\mu}{2}\left\{\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)^2\|v_k - z_k\|_2^2 + 2\sqrt{\frac{\delta}{\mu Cm}}\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\right.$$
$$\left.\cdot(v_k - z_k, G_k+\xi_k) + \frac{\delta}{\mu Cm}\|G_k+\xi_k\|_2^2\right\}$$

On the other hand, from the definition of the estimate sequence and (15),

$$\Phi_{k+1}(v_k) = \left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\left(\Phi_k^* + \frac{\mu}{2}\|v_k - z_k\|_2^2\right) + \sqrt{\frac{\mu\delta}{Cm}}\left(F_{I_k}(v_k)\right.$$
$$\left.+R(w_{k+1}) + (\xi_k, v_k - w_{k+1})\right)$$

Therefore, from the above two equations, we have

$$\Phi_{k+1}^* = \left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\Phi_k^* + \frac{\mu}{2}\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\sqrt{\frac{\mu\delta}{Cm}}\|v_k - z_k\|_2^2$$
$$+\sqrt{\frac{\mu\delta}{Cm}}(F_{I_k}(v_k) + R(w_{k+1}) + (\xi_k, v_k - w_{k+1}))$$
$$-\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\sqrt{\frac{\mu\delta}{Cm}}(v_k - z_k, G_k+\xi_k) - \frac{\delta}{2Cm}\|G_k+\xi_k\|_2^2. \tag{A.1}$$

Since $F(w)$ is Lipschitz smooth, we bound $f(w_{k+1})$ as follows:

$$P(w_{k+1}) \le F(v_k) + (\nabla F(v_k), w_{k+1} - v_k) + \frac{L}{2}\|w_{k+1} - v_k\|_2^2 + R(w_{k+1}). \tag{A.2}$$

Using (A.1), (A.2), (19), and $w_{k+1} = v_k - \frac{\delta}{Cm}(G_k+\xi_k)$

$$\mathbb{E}_{I_k}[P(w_{k+1}) - \Phi_{k+1}^*] \overset{(A.1),(A.2)}{\le} \mathbb{E}_{I_k}\left[\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)(-\Phi_k^* + F(v_k) + R(w_{k+1})) + (\nabla F(v_k), w_{k+1} - v_k)\right.$$

$$
\begin{aligned}
&+\sqrt{\frac{\mu\delta}{Cm}}(\xi, w_{k+1}-v_k)+\frac{L}{2}\|w_{k+1}-v_k\|_2^2-\frac{\mu}{2}\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\sqrt{\frac{\mu\delta}{Cm}}\\
&\cdot\|v_k-z_k\|_2^2+\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\sqrt{\frac{\mu\delta}{Cm}}(v_k-z_k, G_k+\xi_k)+\frac{\delta}{2Cm}\|G_k+\xi_k\|_2^2\Bigg]\\
\overset{(19)}{=}\quad&\mathbb{E}_{I_k}\left[\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)(-\Phi_k^*+F(v_k)+R(w_{k+1})+(w_k-v_k, G_k+\xi_k))\right.\\
&-\frac{\delta}{Cm}(\nabla F(v_k), G_k+\xi_k)-\sqrt{\frac{\mu\delta}{Cm}}\frac{\delta}{Cm}(\xi_k, G_k+\xi_k)+\frac{\delta}{2Cm}\left(\frac{L\delta}{Cm}+1\right)\\
&\left.\cdot\|G_k+\xi_k\|_2^2-\frac{\mu}{2}\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)\cdot\sqrt{\frac{Cm}{\mu\delta}}\|v_k-w_k\|_2^2\right],
\end{aligned}
\tag{A.3}
$$

where in the first inequality we employed $\mathbb{E}[F_{I_k}(v_k)] = F(v_k)$. In addition, we give the following

$$
\begin{aligned}
\mathbb{E}_{I_k}[F(v_k)+R(w_{k+1})+(w_k-v_k, G_k+\xi_k)] &= \mathbb{E}_{I_k}[F(w_k)+(G_k, w_k-v_k)+R(w_{k+1})+(\xi_k, w_k-w_{k+1})\\
&\quad+(\xi_k, w_{k+1}-v_k)]\\
&\le \mathbb{E}_{I_k}\left[F(w_k)-\tfrac{\mu}{2}\|w_k-v_k\|_2^2+R(w_k)-\tfrac{\delta}{Cm}(\xi_k, v_k+\xi_k)\right],
\end{aligned}
\tag{A.4}
$$

where in the first inequality we used $\mathbb{E}_{I_k}[G_k] = \nabla F(v_k)$ and convexity of F and R. Thus we have

$$
\begin{aligned}
\mathbb{E}_{I_k}[P(w_{k+1})-\Phi_{k+1}^*] &\overset{(A.3),(A.4)}{\le} \mathbb{E}_{I_k}\left[\left(1-\sqrt{\tfrac{\mu\delta}{Cm}}\right)(-\Phi_k^*+P(w_k))-\tfrac{\mu}{2}\right.\\
&\left.\cdot\left(1-\tfrac{\mu\delta}{Cm}\right)\sqrt{\tfrac{Cm}{\mu\delta}}\|w_k-v_k\|_2^2-\tfrac{\delta}{Cm}(\nabla F(v_k)+\xi_k, v_k+\xi_k)+\tfrac{\delta}{2Cm}\left(\tfrac{L\delta}{Cm}+1\right)\|G_k+\xi_k\|_2^2\right]\\
&\overset{(20)}{\le} \mathbb{E}_{I_k}\left[\left(1-\sqrt{\tfrac{\mu\delta}{Cm}}\right)(-\Phi_k^*+P(w_k))-\tfrac{\mu}{2}\left(1-\tfrac{\mu\delta}{Cm}\right)\sqrt{\tfrac{Cm}{\mu\delta}}\|w_k-v_k\|_2^2-\tfrac{\delta}{2Cm}\|\nabla F(v_k)+\xi_k\|_2^2\right.\\
&\quad\left.+\tfrac{\delta}{2Cm}\tfrac{L\delta}{Cm}\|G_k+\xi_k\|_2^2+\tfrac{\delta}{2Cm}\|\nabla F(v_k)-G_k\|_2^2\right]\\
&\overset{(21),m>\frac{2L\delta}{C}}{\le} \mathbb{E}_{I_k}\left[\left(1-\sqrt{\tfrac{\mu\delta}{Cm}}\right)(-\Phi_k^*+P(w_k))-\tfrac{\mu}{2}\left(1-\tfrac{\mu\delta}{Cm}\right)\sqrt{\tfrac{Cm}{\mu\delta}}\|w_k-v_k\|_2^2+\tfrac{\delta}{Cm}\|\nabla F(v_k)-G_k\|_2^2\right].
\end{aligned}
$$

By taking expectation with respect to the history of random variables $I_1, \dots, I_{k-1}$, the induction hypothesis finishes the proof of (17). □

A2. Proof of Theorem 1

Now, we give the proof of Theorem 1.

From (17), (23) and (16) with $w = w_*$, we have

$$
\begin{aligned}
&\mathbb{E}[P(w_k)-P(w_*)] \le \left(1-\sqrt{\tfrac{\mu\delta}{Cm}}\right)^{k-1}(\Phi_1-P)(w_*)+\mathbb{E}\left[\sum_{l=1}^{k-1}\left(1-\sqrt{\tfrac{\mu\delta}{Cm}}\right)^{k-1-l}\right.\\
&\left.\left\{\left(-\tfrac{\mu}{2}\tfrac{1-\mu\delta/Cm}{\sqrt{\mu\delta/Cm}}+\alpha(b)\cdot\tfrac{2\delta L^2}{Cm}\right)\|v_l-w_l\|_2^2+\alpha(b)\cdot\tfrac{8\delta L}{Cm}(P(w_k)-P(w_*)+P(\widetilde{w})-P(w_*))\right\}\right]
\end{aligned}
$$

When $m \ge \max\left\{\frac{2\delta L}{C}\sqrt[3]{\frac{2L\alpha(b)^2}{\mu}}, \frac{2\delta L}{C}\right\}$ is satisfied, we have

$$
\begin{aligned}
\mathbb{E}[P(w_k)-P(w_*)] &\le \left(1-\sqrt{\tfrac{\mu\delta}{Cm}}\right)^{k-1}(\Phi_1-P)(w_*)+\mathbb{E}\left[\sum_{l=1}^{k-1}\left(1-\sqrt{\tfrac{\mu\delta}{Cm}}\right)^{k-1-l}\right.\\
&\quad\left.\alpha(b)\cdot\tfrac{8\delta L}{Cm}(P(w_l)-P(w_*)+P(\widetilde{w})-P(w_*))\right]\\
\le\ &\left(1-\sqrt{\tfrac{\mu\delta}{Cm}}\right)^{k-1}(\Phi_1-P)(w_*)+8L\sqrt{\tfrac{\delta}{\mu Cm}}(P(\widetilde{w})-P(w_*))+\mathbb{E}\left[\sum_{l=1}^{k-1}\left(1-\sqrt{\tfrac{\mu\delta}{Cm}}\right)^{k-1-l}\right.\\
&\quad\left.\alpha(b)\cdot\tfrac{8\delta L}{Cm}(P(w_l)-P(w_*))\right],
\end{aligned}
\tag{A.5}
$$

where in the last inequality we employed $\sum_{l=1}^{k-1}\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)^{k-1-l} \le \sum_{t=0}^{\infty}\left(1-\sqrt{\frac{\mu\delta}{Cm}}\right)^{t} = \sqrt{\frac{Cm}{\mu\delta}}$.

We use V_k to denote $\mathbb{E}[P(w_k) - P(w_*)]$, and we take W_k to denote the last expression in (A.5). Hence, for $k \geq 1$, $V_k \leq W_k$. For $k \geq 2$, we have

$$\begin{aligned}
W_k = & \left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)\left\{\left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)^{k-2} (\Phi_1 - P)(w_*) + 8L\sqrt{\frac{\delta}{\mu Cm}}V_1\right. \\
& \left. + \sum_{l=1}^{k-2}\left(1 - \sqrt{\frac{\mu\delta}{Cm}}\right)^{k-2-l} \alpha(b) \cdot \frac{8\delta L}{Cm} V_l\right\} + \alpha(b) \cdot \frac{8\delta L}{Cm} V_{k-1} \\
& + 8L\sqrt{\frac{\mu\delta}{Cm}}\sqrt{\frac{\delta}{\mu Cm}}V_1 \\
\leq & \left(1 - \sqrt{\frac{\mu\delta}{Cm}} + \alpha(b) \cdot \frac{8\delta L}{Cm}\right) W_{k-1} + 8L\sqrt{\frac{\mu\delta}{Cm}}\sqrt{\frac{\delta}{\mu Cm}} W_1
\end{aligned}$$

Since $m > \frac{2L\delta}{C}$ is satisfied

$$W_k \leq \left[\left(1 - \sqrt{\frac{\mu\delta}{Cm}} + \alpha(b) \cdot \frac{8\delta L}{Cm}\right)^{k-1} + \frac{1}{\frac{1}{8L}\sqrt{\frac{\mu Cm}{\delta}} - \alpha(b)}\right] W_1$$

From the strong convexity of $P(w)$, we have

$$\begin{aligned}
W_1 = & \left(1 + 8L\frac{\delta}{\mu Cm}\right)(P(\tilde{w}) - P(w_*)) + \frac{\mu}{2}\|\tilde{w} - w_*\|_2^2 \\
\leq & \left(2 + 8L\frac{\delta}{\mu Cm}\right)(P(\tilde{w}) - P(w_*)).
\end{aligned}$$

Thus, for $k \geq 2$, we have

$$\begin{aligned}
V_k \leq W_k \leq & \left(\left(1 - \sqrt{\frac{\mu\delta}{Cm}} + \alpha(b) \cdot \frac{8\delta L}{Cm}\right)^{k-1} + \frac{1}{\frac{1}{8L}\sqrt{\frac{\mu Cm}{\delta}} - \alpha(b)}\right) \\
& \times \left(2 + \frac{8L\delta}{\mu Cm}\right)(P(\tilde{w}) - P(w_*))
\end{aligned}$$

References

[1] C. Wang, X. Wang, C. Zhang, Z. Xia, Geometric correction based color image watermarking using fuzzy least squares support vector machine and bessel k form distribution, Signal Process. 134 (2017) 197–208.

[2] J. Zhu, S. Rosset, R. Tibshirani, T.J. Hastie, 1-norm support vector machines, in: Advances in Neural Information Processing Systems, 2004, pp. 49–56.

[3] S. Tong, E. Chang, Support vector machine active learning for image retrieval, in: ACM International Conference on Multimedia, 2001, pp. 107–118.

[4] T. Kronvall, S.I. AdalbjÃ¶rnsson, S. Nadig, A. Jakobsson, Group-sparse regression using the covariance fitting criterion, Signal Process. 139 (C) (2017) 116–130.

[5] G. Mateos, J.A. Bazerque, G.B. Giannakis, Distributed sparse linear regression, IEEE Trans. Signal Process. 58 (10) (2010) 5262–5276.

[6] Y. Plan, R. Vershynin, Robust 1-bit compressed sensing and sparse logistic regression: a convex programming approach, IEEE Trans. Inf. Theory 59 (1) (2013) 482–494.

[7] L. Meier, S.V. De Geer, P. Buhlmann, The group lasso for logistic regression, J. R. Statist. Soc. Ser. B 70 (1) (2008) 53–71.

[8] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imag. Sci. 2 (1) (2009) 183–202.

[9] A. Beck, M. Teboulle, Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems, IEEE Trans. Image Process. 18 (11) (2009) 2419–2434.

[10] Y. Nesterov, Gradient methods for minimizing composite objective function, Core Discussion Papers 140 (1) (2007) 125–161.

[11] Y. Nesterov, Smooth minimization of non-smooth functions, Math. Program. 103 (1) (2005) 127–152.

[12] L. Bottou, Stochastic gradient descent tricks, in: Neural networks: Tricks of the trade, Springer, 2012, pp. 421–436.

[13] S. Klein, J.P.W. Pluim, M. Staring, M.A. Viergever, Adaptive stochastic gradient descent optimisation for image registration, Int. J. Comput. Vis. 81 (3) (2009) 227–239.

[14] Y. Tsuruoka, J. Tsujii, S. Ananiadou, Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty, in: Joint Conference of the Meeting of the ACL and the International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1, 2009, pp. 477–485.

[15] T. Zhang, Solving large scale linear prediction problems using stochastic gradient descent algorithms, in: International Conference on Machine learning, ACM, 2004, p. 116.

[16] Y. Nesterov, Introductory Lectures on Convex Optimization : a Basic Course, Springer Science + Business Media, New York, 2004.

[17] S. Ghadimi, G. Lan, Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: a generic algorithmic framework, SIAM J. Optim. 22 (4) (2012) 1469–1492.

[18] S. Shalev-Shwartz, T. Zhang, Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization, in: International Conference on Machine Learning, 2014, pp. 64–72.

[19] A. Nitanda, Stochastic proximal gradient descent with acceleration techniques, in: Advances in Neural Information Processing Systems, 2014, pp. 1574–1582.

[20] L. Xiao, T. Zhang, A proximal stochastic gradient method with progressive variance reduction, SIAM J. Optim. 24 (4) (2014) 2057–2075.

[21] A. Nitanda, Accelerated stochastic gradient descent for minimizing finite sums, in: International Conference on Artificial Intelligence and Statistics, 2016.

[22] Z. Allen-Zhu, L. Orecchia, Linear coupling of gradient and mirror descent: a novel simple interpretation of nesterovs accelerated method, 2014 arXiv:1407.1537.

[23] C. Hu, W. Pan, J.T. Kwok, Accelerated gradient methods for stochastic optimization and online learning, in: Advances in Neural Information Processing Systems, 2009, pp. 781–789.

[24] H. Li, Z. Lin, Accelerated proximal gradient methods for nonconvex programming, in: Advances in Neural Information Processing Systems, 2015, pp. 379–387.

[25] M. Wang, J. Liu, E.X. Fang, Accelerating stochastic composition optimization, J. Mach. Learn. Res. 18 (105) (2017) 1–23.

[26] O. Shamir, T. Zhang, Stochastic gradient descent for non-smooth optimization: convergence results and optimal averaging schemes, in: International Conference on Machine Learning, 2012, pp. 71–79.

[27] V.S. Borkar, S.P. Meyn, The o.d. e. method for convergence of stochastic approximation and reinforcement learning, SIAM J. Control Optim. 38 (2) (2000) 447–469.

[28] D.P. Bertsekas, J.N. Tsitsiklis, Gradient convergence in gradient methods with errors, SIAM J. Optim. 10 (3) (1999) 627–642.

[29] C. Darken, J. Moody, Note on learning rate schedules for stochastic optimization, in: Advances in Neural Information Processing Systems, 1990, pp. 832–838.

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

[31] N.L. Roux, M. Schmidt, F.R. Bach, A stochastic gradient method with an exponential convergence rate for finite training sets, in: Advances in Neural Information Processing Systems, 2012, pp. 2663–2671.

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

[33] S. Shalev-Shwartz, T. Zhang, Stochastic dual coordinate ascent methods for regularized loss minimization, J. Mach. Learn. Res. 14 (Feb) (2013) 567–599.

[34] R. Johnson, T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in: Advances in Neural Information Processing Systems, 2013, pp. 315–323.

[35] J. Konečnỳ, P. Richtárik, Semi-stochastic gradient descent methods, Front. Appl. Math. Statist. 3 (2017) 9.

[36] J. Konečnỳ, J. Liu, P. Richtárik, M. Takáč, Mini-batch semi-stochastic gradient descent in the proximal setting, IEEE J. Sel. Top. Signal Process. 10 (2) (2016) 242–255.

[37] F. Yousefian, A. Nedi, U.V. Shanbhag, On stochastic gradient and subgradient methods with adaptive steplength sequences, Automatica 48 (1) (2012) 56–67.

[38] M. Mahsereci, P. Hennig, Probabilistic line searches for stochastic optimization, J. Mach. Learn. Res. 18 (119) (2017) 1–59.

[39] C. Tan, S. Ma, Y.H. Dai, Y. Qian, Barzilai-Borwein step size for stochastic gradient descent, in: Advances in Neural Information Processing Systems, 2016, pp. 685–693.

[40] S. De, A. Yadav, D. Jacobs, T. Goldstein, Automated inference with adaptive batches, in: International Conference on Artificial Intelligence and Statistics, 2017.

[41] Z. Yang, C. Wang, Z. Zhang, J. Li, Random Barzilai-Borwein step size for mini-batch algorithms, Eng. Appl. Artif. Intell. 72 (2018) 124–135.

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

[43] W. Yin, Analysis and generalizations of the linearized bregman method, SIAM J. Imag. Sci. 3 (4) (2010) 856–877.

[44] M.A.T. Figueiredo, R.D. Nowak, S.J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, IEEE J. Sel. Top. Signal Process. 1 (4) (2007) 586–597.

[45] S.J. Wright, R.D. Nowak, M.A.T. Figueiredo, Sparse reconstruction by separable approximation, IEEE Trans. Signal Process. 57 (7) (2009) 2479–2493.

[46] M. Schmidt, R. Babanezhad, M.O. Ahmed, A. Defazio, A. Clifton, A. Sarkar, Non-uniform stochastic average gradient method for training conditional random fields, in: International Conference on Artificial Intelligence and Statistics, 2015.