Accelerated Variance Reduction Stochastic ADMM for Large-Scale Machine Learning

Yuanyuan Liu, *Member, IEEE*, Fanhua Shang, *Senior Member, IEEE*, Hongying Liu, *Member, IEEE*, Lin Kong, Licheng Jiao, *Fellow, IEEE*, and Zhouchen Lin, *Fellow, IEEE*

Abstract—Recently, many stochastic variance reduced alternating direction methods of multipliers (ADMMs) (e.g., SAG-ADMM and SVRG-ADMM) have made exciting progress such as linear convergence rate for strongly convex (SC) problems. However, their best-known convergence rate for non-strongly convex (non-SC) problems is $\mathcal{O}(1/T)$ as opposed to $\mathcal{O}(1/T^2)$ of accelerated deterministic algorithms, where T is the number of iterations. Thus, there remains a gap in the convergence rates of existing stochastic ADMM and deterministic algorithms. To bridge this gap, we introduce a new momentum acceleration trick into stochastic variance reduced ADMM, and propose a novel accelerated SVRG-ADMM method (called ASVRG-ADMM) for the machine learning problems with the constraint $Ax+By=c$. Then we design a linearized proximal update rule and a simple proximal one for the two classes of ADMM-style problems with $B=\tau I$ and $B\neq \tau I$, respectively, where I is an identity matrix and τ is an arbitrary bounded constant. Note that our linearized proximal update rule can avoid solving sub-problems iteratively. Moreover, we prove that ASVRG-ADMM converges linearly for SC problems. In particular, ASVRG-ADMM improves the convergence rate from $\mathcal{O}(1/T)$ to $\mathcal{O}(1/T^2)$ for non-SC problems. Finally, we apply ASVRG-ADMM to various machine learning problems, e.g., graph-guided fused Lasso, graph-guided logistic regression, graph-guided SVM, generalized graph-guided fused Lasso and multi-task learning, and show that ASVRG-ADMM consistently converges faster than the state-of-the-art methods.

Index Terms—Stochastic optimization, ADMM, variance reduction, momentum acceleration, strongly convex and non-strongly convex, smooth and non-smooth

1 Introduction

This paper mainly considers the following composite finite-sum equality-constrained optimization problem

$$\min_{x\in\mathbb{R}^{d_x},y\in\mathbb{R}^{d_y}} \Big\{f(x)+h(y),\ \text{s.t.},\ Ax+By=c\Big\}, \tag{1}$$

where $c\in\mathbb{R}^{d_c}$, $A\in\mathbb{R}^{d_c\times d_x}$, $B\in\mathbb{R}^{d_c\times d_y}$, $f(x):=\frac{1}{n}\sum_{i=1}^{n}f_i(x)$, each component function $f_i(\cdot)$ is convex, and $h(\cdot)$ is convex but possibly non-smooth. For instance, a popular choice of $f_i(\cdot)$ in binary classification problems is the logistic loss, i.e., $f_i(x)=\log(1+\exp(-b_i a_i^T x))$, where (a_i,b_i) is the feature-label pair, and $b_i\in\{\pm1\}$. With regard to $h(\cdot)$, we are interested in a sparsity-inducing regularizer, e.g., ℓ_1-norm [1], [2], group Lasso [3], [4] and nuclear norm [5], [6], [7].

Problem (1) arises in many places in machine learning, pattern recognition, computer vision, statistics, and operations research [8]. When the constraint in Eq. (1) is $Ax=y$, the formulation (1) becomes

$$\min_{x\in\mathbb{R}^{d_x},y\in\mathbb{R}^{d_y}} \Big\{f(x)+h(y),\ \text{s.t.},\ Ax=y\Big\}, \tag{2}$$

where $A\in\mathbb{R}^{d_y\times d_x}$. Recall that this class of problems include the graph-guided fused Lasso [3], generalized Lasso [4] and graph-guided support vector machine (SVM) [9] as notable examples. If the constraint degenerates to $x=y$, this class of problems include the regularized empirical risk minimization (ERM) problem, e.g., logistic regression, Lasso and linear SVM.

For solving the large-scale optimization problem involving a large sum of n component functions, stochastic gradient descent (SGD) [10] uses only one or a mini-batch of gradients in each iteration, and thus enjoys a significantly lower per-iteration complexity than deterministic methods including Nesterov's accelerated gradient descent (AGD) [11], [12] and accelerated proximal gradient (APG) [13], [14], i.e., $O(d_x)$ versus $O(nd_x)$. Therefore, SGD has been successfully applied to many large-scale machine learning problems [9], [15], [16], especially training deep network models [17]. However, the variance of the stochastic gradient estimator may be large, and thus we need to gradually reduce its step-size, which leads to slow convergence [18], especially for equality-constrained composite convex problems [19].

This paper mainly focuses on the large sample regime. In this regime, even first-order deterministic methods such as FISTA [14] become computationally burdensome due to their per-iteration complexity of $O(nd_x)$. As a result, SGD with low per-iteration complexity $O(d_x)$ has witnessed tremendous

- *Y. Liu, H. Liu, L. Kong, and L. Jiao are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi 710071, China. E-mail: {yyliu, hyliu}@xidian.edu.cn, xdkonglin0511@163.com, lchjiao@mail.xidian.edu.cn.*
- *F. Shang is with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi 710071, China, and also with the Peng Cheng Laboratory, Shenzhen 518066, China. E-mail: fhshang@xidian.edu.cn.*
- *Z. Lin is with the Key Laboratory of Machine Perception (MoE), School of EECS, Peking University, Beijing 100871, China. E-mail: zlin@pku.edu.cn.*

Manuscript received 8 Aug. 2019; revised 1 Mar. 2020; accepted 20 May 2020. Date of publication 8 June 2020; date of current version 3 Nov. 2021.
(Corresponding authors: Fanhua Shang and Hongying Liu.)
Recommended for acceptance by C. Zhang.
Digital Object Identifier no. 10.1109/TPAMI.2020.3000512

TABLE 1
Comparison of Convergence Rates and Memory Requirements of Various Stochastic ADMM Algorithms, Including Stochastic ADMM (STOC-ADMM) [9], Stochastic Average Gradient ADMM (SAG-ADMM) [19], Stochastic Dual Coordinate Ascent ADMM (SDCA-ADMM) [20], Scalable Stochastic ADMM (SCAS-ADMM) [21], Stochastic Variance Reduced Gradient ADMM (SVRG-ADMMM) [22], and Our ASVRG-ADMM

	Non-strongly convex	Strongly convex	Constraints	Space requirement
STOC-ADMM [9]	$\mathcal{O}(1/\sqrt{T})$	$\mathcal{O}(\log T/T)$	$Ax=y$	$O(d_x d_y + d_x^2)$
SAG-ADMM [19]	$\mathcal{O}(1/T)$	unknown	$Ax=y$	$O(d_x d_y + n d_x)$
SDCA-ADMM [20]	unknown	linear rate	$Ax+By=c$	$O(d_x d_y + n)$
SCAS-ADMM [21]	$\mathcal{O}(1/T)$	$\mathcal{O}(1/T)$	$Ax=y$	$O(d_x d_y)$
SVRG-ADMM [22]	$\mathcal{O}(1/T)$	linear rate	$Ax=y$	$O(d_x d_y)$
ASVRG-ADMM (ours)	$\mathcal{O}(1/T^2)$	linear rate	$Ax+By=c$	$O(d_x d_y)$

It should be noted although all the methods except SDCA-ADMM apply the same update rule in (4), their algorithms do not actually work for solving the problem (1) with the constraint $Ax+By=c$, where $B\neq \tau I$, τ is an arbitrary bounded constant, and I is an identity matrix.

progress in the recent years. Recently, a number of stochastic variance reduced methods such as SAG [23], SDCA [24], SVRG [18], Prox-SVRG [25] and VR-SGD [26] have been proposed to successfully address the problem of high variance of stochastic gradient estimators in ordinary SGD, resulting in linear convergence for strongly convex problems as opposed to sub-linear rates of SGD. More recently, the Nesterov's acceleration technique [27] was introduced in [28], [29], [30], [31] to further speed up the stochastic variance reduced algorithms, which results in the best-known convergence rates for both strongly convex (SC) and non-strongly convex (non-SC) problems, e.g., Katyusha [29]. This also motivates us to integrate the momentum acceleration trick into the stochastic alternating direction method of multipliers (ADMM) below.

1.1 Review of Stochastic ADMMs

It is well known that the ADMM is an effective optimization tool [32] to solve this class of composite optimization problems (1). The ADMM has shown attractive performance in a wide range of real-world problems, such as big data classification [33] and matrix and tensor recovery [5], [34], [35]. We refer the reader to [36], [37], [38], [39] for some review papers on the ADMM. Recently, several faster deterministic ADMM algorithms have been proposed to solve some special cases of Problem (1). For instance, [40] proposed an accelerated ADMM, and proved that their algorithm has an $\mathcal{O}(1/T^2)$ convergence rate for SC problems, similar to [37], [41].[1] [42], [43] proposed a faster ADMM algorithm with a convergence rate $\mathcal{O}(1/T^2)$ for solving the special case of Problem (1) with the constraint $Ax=y$. However, the per-iteration complexity of all the full-batch ADMMs is $O(nd_x)$, and thus they become very slow and are not suitable for large-scale machine learning problems.

To tackle the issue of high per-iteration complexity of deterministic ADMMs, [9], [44], [45] proposed some online or stochastic ADMM algorithms. However, all these variants only achieve the convergence rate of $\mathcal{O}(\log T/T)$ for SC problems and $\mathcal{O}(1/\sqrt{T})$ for non-SC problems, respectively, as compared with the linear convergence and $\mathcal{O}(1/T^2)$ rates of the accelerated deterministic ADMM algorithms mentioned above. Recently, several accelerated and faster converging versions of stochastic ADMMs such as SAG-ADMM [19], SDCA-ADMM [20] and SVRG-ADMM [22], which are all based on variance reduction techniques, have been proposed. With regard to strongly convex problems, [20], [22] proved that linear convergence can be obtained for the special ADMM form (i.e., Problem (2)) and the general ADMM form, respectively. [46] also proposed a fast stochastic variance reduced ADMM for stochastic composition optimization problems. More recently, [47], [48] proposed two accelerated stochastic ADMM algorithms for the problem (2) and four-composite optimization problems, respectively. For SAG-ADMM and SVRG-ADMM, an $\mathcal{O}(1/T)$ convergence rate can be guaranteed for non-strongly convex problems, which implies that there remains a gap in convergence rates between the stochastic ADMMs and accelerated deterministic algorithms, i.e., $\mathcal{O}(1/T)$ versus $\mathcal{O}(1/T^2)$.

1.2 Contributions

To fill in this gap, we design a new momentum acceleration trick similar to the ones in deterministic optimization and incorporate it into the stochastic variance reduction gradient (SVRG) based stochastic ADMM (SVRG-ADMM) [22]. Naturally, the proposed method has a low per-iteration cost as existing stochastic ADMM algorithms such as SVRG-ADMM, and does not require the storage of all gradients (or dual variables) as in SAG-ADMM [19] and SCAS-ADMM [21], as shown in Table 1.

The main differences between this paper and our previous conference paper [49] are listed as follows: 1) We briefly review recent work on stochastic ADMM for solving Problems (1) and (2). 2) When $B\neq\tau I$ in Eq. (1), where τ is an arbitrary bounded constant and I is an identity matrix, the sub-problem with respect to y (see Eq. (4) below) has no closed-form solution and has to be solved iteratively. To overcome this difficulty, we present a new linearized proximal update rule for both SC and non-SC problems (1) with the constraint $Ax+By=c$ when $B\neq\tau I$. In other words, the existing stochastic ADMM algorithms including the proposed ones in our previous work [49] do not work for this case. Although the theoretical guarantees of existing variance reduced stochastic ADMMs except SDCA-ADMM [20] are for Problem (1) with the general constraint $Ax+By=c$, they do not actually work for solving such problems. 3) For the case of $B=\tau I$, we use a simple proximal update rule as

1. Note that, for simplicity, we do not differentiate the $\mathcal{O}(1/T^2)$ and $o(1/T^2)$ because they are of the same order in the worst-case nature and their difference is insignificant in general, where T is the number of iterations.

in our previous work [49] instead of the linearized proximal one. Then we propose two novel accelerated SVRG-ADMM algorithms (called ASVRG-ADMM) for both SC and non-SC problems. 4) We also theoretically analyze the convergence properties of the proposed ASVRG-ADMM algorithms for both SC and non-SC problems and the two cases of $B \neq \tau I$ and $B = \tau I$, respectively. 5) We further improve the theoretical results in our previous work [49] by removing the boundedness assumption. 6) Finally, we report more experimental results especially for the ADMM problem (1) with the constraint $Ax + By = c$ to verify both the effectiveness and efficiency of ASVRG-ADMM.

The main contributions of this paper are summarized as follows.

- We propose an efficient accelerated variance reduced stochastic ADMM (ASVRG-ADMM) method, which integrates both our momentum acceleration trick and the variance reduction technique of SVRG-ADMM [22]. Moreover, ASVRG-ADMM has a linearized proximal rule and a simple proximal one for both cases of $B \neq \tau I$ and $B = \tau I$, respectively.
- We prove that ASVRG-ADMM achieves a linear convergence rate for SC problems, which is consistent with the best-known result in SDCA-ADMM [20] and SVRG-ADMM [22]. Besides, ASVRG-ADMM uses its linearized proximal rule and thus becomes more practical than existing algorithms, which have to solve the sub-problems iteratively.
- In particular, for the more general problem (1) with the constraint $Ax + By = c$ and $B \neq \tau I$, we also design a novel epoch initialization technique for the variable y at each epoch of our linearized proximal acceleration algorithm for SC problems.
- We also prove that ASVRG-ADMM has a convergence rate $\mathcal{O}(1/T^2)$ for non-SC problems, which means that ASVRG-ADMM is a factor T faster than SAG-ADMM and SVRG-ADMM, whose convergence rate is $\mathcal{O}(1/T)$. In particular, we design an adaptive increasing epoch length strategy and further improve the theoretical results by using this strategy and removing boundedness assumptions.
- Various experimental results on synthetic and real-world datasets further verify that our ASVRG-ADMM converges consistently much faster than the state-of-the-art stochastic ADMM methods.

The remainder of this paper is organized as follows. Section 2 discusses some recent advances in stochastic ADMM. Section 3 proposes a new accelerated stochastic variance reduction ADMM method (called ASVRG-ADMM) with the proposed momentum acceleration trick. Moreover, we analyze the convergence properties of ASVRG-ADMM in Section 4. Experimental results in Section 5 show the effectiveness of ASVRG-ADMM. In Section 6, we conclude this paper and discuss the future work.

2 Related Work

This section reveals recent progresses and efforts in stochastic optimization methods that are based on the stochastic alternating direction method of multipliers (ADMM).

2.1 Notation

Throughout this paper, the norm $\|\cdot\|$ denotes the standard euclidean norm, and $\|\cdot\|_1$ is the ℓ_1-norm, i.e., $\|x\|_1 = \sum_i |x_i|$. We denote by $\nabla f(x)$ the gradient of $f(x)$ if it is differentiable, or $f'(x)$ any of the subgradients of $f(\cdot)$ at x if $f(\cdot)$ is only Lipschitz continuous. To facilitate our discussion, we first make the following basic assumptions.

2.2 Basic Assumptions

Assumption 1 (Smoothness). *Each convex component function $f_i(\cdot)$ is L-smooth if its gradients are L-Lipschitz continuous, that is*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|, \text{ for all } x, y \in \mathbb{R}^d.$$

Assumption 2 (Strong Convexity). *A convex function $g(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex, if there exists a constant $\mu > 0$ such that*

$$g(y) \geq g(x) + \langle \nabla g(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2, \text{ for all } x, y \in \mathbb{R}^d.$$

If $g(\cdot)$ is non-smooth, we modify the above inequality by simply replacing $\nabla g(x)$ with an arbitrary sub-gradient $g'(x)$.

2.3 Stochastic ADMM

It is easy to see that Problem (2) is only a special case of the general ADMM form (1) when $B = -I$ and $c = \mathbf{0}$. Thus, the purpose of this paper is to propose an accelerated stochastic variance reduced ADMM method for solving the more general problem (1). Although the stochastic (or online) ADMM algorithms and theoretical results in [9], [19], [22], [44] are all for the problem (1), they do not actually work.

The augmented Lagrangian function of Problem (1) is

$$\begin{aligned}\mathcal{L}(x, y, \lambda) &= f(x) + h(y) + \langle \lambda, Ax + By - c \rangle \\ &\quad + \frac{\beta}{2}\|Ax + By - c\|^2\end{aligned} \tag{3}$$

where λ is the vector of Lagrangian multipliers (also called the dual variable), and $\beta > 0$ is a penalty parameter. To minimize Problem (1), together with the dual variable λ, the update steps of deterministic ADMM are

$$y_k = \arg\min\nolimits_y \Big\{ h(y) + \frac{\beta}{2}\|Ax_{k-1} + By - c + \lambda_{k-1}\|^2 \Big\}, \tag{4}$$

$$x_k = \arg\min\nolimits_x \Big\{ f(x) + \frac{\beta}{2}\|Ax + By_k - c + \lambda_{k-1}\|^2 \Big\}, \tag{5}$$

$$\lambda_k = \lambda_{k-1} + Ax_k + By_k - c. \tag{6}$$

To extend the deterministic ADMM to the online and stochastic settings, the update rules for y_k and λ_k remain unchanged, while in [9], [44], the update rule of x_k is approximated as follows:

$$\begin{aligned}x_k = \arg\min_x \Big\{ &\big\langle x, \nabla f_{i_k}(x_{k-1}) \big\rangle + \frac{1}{2\eta_k}\|x - x_{k-1}\|_G^2 \\ &+ \frac{\beta}{2}\|Ax + By_k - c + \lambda_{k-1}\|^2 \Big\},\end{aligned} \tag{7}$$

where we draw i_k uniformly at random from $[n] := \{1, \ldots, n\}$, $\eta_k \propto 1/\sqrt{k}$ is the learning rate or step-size, and

$\|z\|_G^2 = z^T G z$ with a given positive semi-definite matrix G, e.g., $G \succeq I_{d_1}$ as in [22]. Analogous to SGD, the stochastic ADMM variants also use an unbiased estimate of the gradient at each iteration, i.e., $\mathbb{E}[\nabla f_{i_k}(x_{k-1})] = \nabla f(x_{k-1})$. However, all those algorithms have much slower convergence rates than their deterministic counterparts mentioned above. This barrier is mainly due to the large variance introduced by the stochasticity of the gradients [18]. Essentially, to guarantee convergence of SGD and its ADMM variants, we need to employ a decaying sequence of step-sizes $\{\eta_k\}$, which in turn leads to slower convergence rates.

Recently, a number of variance reduced stochastic ADMM methods (e.g., SAG-ADMM and SVRG-ADMM) have been proposed and made exciting progress such as linear convergence rates. SVRG-ADMM [22] is particularly attractive here because of its low storage requirement compared with the algorithms in [19], [20]. Within each epoch of mini-batch SVRG-ADMM, the full gradient $\widetilde{p} = \nabla f(\widetilde{x})$ is first computed, where $\widetilde{x}$ is the average point of the previous epoch. Then $\nabla f_{i_k}(x_{k-1})$ and η_k in (7) are replaced by

$$\widetilde{\nabla} f_{I_k}(x_{k-1}) = \frac{1}{|I_k|} \sum_{i_k \in I_k} \left(\nabla f_{i_k}(x_{k-1}) - \nabla f_{i_k}(\widetilde{x})\right) + \widetilde{p}, \tag{8}$$

and a constant step-size η, respectively, where $I_k \subset [n]$ is a randomly chosen mini-batch of size b. Note that mini-batching is a useful technique to reduce the variance of the stochastic gradients [26], [50]. In fact, $\widetilde{\nabla} f_{I_k}(x_{k-1})$ is also an unbiased estimator of the gradient $\nabla f(x_{k-1})$, i.e., $\mathbb{E}[\widetilde{\nabla} f_{I_k}(x_{k-1})] = \nabla f(x_{k-1})$.

Algorithm 1. ASVRG-ADMM for Strongly-Convex Problems

Input: $m, \eta, \beta > 0, 1 \le b \le n$.
Initialize: $\widetilde{x}^0, \widetilde{y}^0, \theta, \nu = 1 + \frac{\eta\beta\|B^T B\|_2}{\theta}, \gamma = 1 + \frac{\eta\beta\|A^T A\|_2}{\theta}$;
for $s = 1, 2, \ldots, T$ **do**
  $\widetilde{p} = \nabla f(\widetilde{x}^{s-1})$, $\widetilde{\lambda}^{s-1} = -\frac{1}{\beta}(A^T)^\dagger \nabla f(\widetilde{x}^{s-1})$, $\lambda_0^s = \widetilde{\lambda}^{s-1}$;
  $x_0^s = z_0^s = \widetilde{x}^{s-1}$, $y_0^s = \widetilde{y}^{s-1}$ (Case of $B = \tau I$) or $y_0^s = -B^\dagger (Az_0^s - c)$ (Case of $B \neq \tau I$);
  for $k = 1, 2, \ldots, m$ **do**
    Choose $I_k \subseteq [n]$ of size b, uniformly at random;
    $\widetilde{\nabla} f_{I_k}(x_{k-1}^s) = \frac{1}{|I_k|}\sum_{i_k \in I_k}\left[\nabla f_{i_k}(x_{k-1}^s) - \nabla f_{i_k}(\widetilde{x}^{s-1})\right] + \widetilde{p}$;
    $y_k^s = \mathrm{Prox}_h^{\frac{1}{\beta\tau^2}}\left((-Az_{k-1}^s + c - \lambda_{k-1}^s)/\tau\right)$ for the case of $B = \tau I$,
    $y_k^s = \mathrm{Prox}_h^{\frac{\eta}{\theta\nu}}\left[y_{k-1}^s - \frac{\eta\beta}{\theta\nu} B^T(Az_{k-1}^s + By_{k-1}^s - c + \lambda_{k-1}^s)\right]$ for the case of $B \neq \tau I$;
    $z_k^s = z_{k-1}^s - \frac{\eta}{\gamma\theta}\left[\widetilde{\nabla} f_{I_k}(x_{k-1}^s) + \beta A^T(Az_{k-1}^s + By_k^s - c + \lambda_{k-1}^s)\right]$;
    $x_k^s = (1-\theta)\widetilde{x}^{s-1} + \theta z_k^s$;
  end for
  $\widetilde{x}^s = \frac{1}{m}\sum_{k=1}^m x_k^s$, $\widetilde{y}^s = (1-\theta)\widetilde{y}^{s-1} + \frac{\theta}{m}\sum_{k=1}^m y_k^s$;
end for
Output: $\widetilde{x}^T, \widetilde{y}^T$.

For the equality-constrained composite convex problem (2), Xu *et al.* [47] proposed a faster variant of SVRG-ADMM with an adaptive penalty parameter scheme. Fang *et al.* [48] proposed an accelerated stochastic ADMM with Nesterov's extrapolation and variance reduction techniques for solving four-composite optimization problems. Moreover, Huang *et al.* [51], and Huang and Chen [52] proposed several variants of SVRG-ADMM for solving non-smooth and non-convex optimization problems.

3 Momentum Accelerated Variance Reduction Stochastic ADMM

In this section, we propose an efficient accelerated variance reduced stochastic ADMM (ASVRG-ADMM) method for solving both SC and non-SC problems (1). In particular, we design two new linearized proximal accelerated algorithms for both SC and non-SC problems with the constraint $Ax + By = c$ and $B \neq \tau I$, respectively.

3.1 ASVRG-ADMM for Strongly Convex Problems

In this part, we first consider the case of Problem (1) when each $f_i(\cdot)$ is convex, L-smooth, and $f(\cdot)$ is μ-strongly convex. Recall that this class of problems include graph-guided logistic regression and support vector machines (SVM) as notable examples. To efficiently solve this class of problems, we incorporate both the momentum acceleration trick proposed in our previous work [49] and the variance reduced stochastic ADMM [22], as shown in Algorithm 1. All our algorithms including Algorithm 1 are divided into T epochs, and each epoch consists of m stochastic updates, where m is usually chosen to be $m = \Theta(n)$ as in [18], [49].

3.1.1 *Update Rule of y*

As in both SVRG-ADMM [22] and ASVRG-ADMM [49], the variable y is updated by solving the following problem for both strongly convex and non-strongly convex cases:

$$y_k^s = \arg\min_y \left\{h(y) + \frac{\beta}{2}\|Az_{k-1}^s + By - c + \lambda_{k-1}^s\|^2\right\}, \tag{9}$$

where the superscript s indicates the sth epoch, the subscript k denotes the kth inner-iteration, z_{k-1}^s is an auxiliary variable and its update rule is given in Section 3.1.2.

When $B = \tau I$ (e.g., B is an identity matrix), the solution to the problem in Eq. (9) can be relatively easily obtained. In other words, we still apply the simple proximal rule proposed in our previous work [49] to solve such problems. For this case, we give the following proximal update rule

$$y_k^s = \mathrm{Prox}_h^{\frac{1}{\beta\tau^2}}\left((-Az_{k-1}^s + c - \lambda_{k-1}^s)/\tau\right),$$

where the proximal operator $\mathrm{Prox}_h^\delta(\cdot)$ is defined as

$$\mathrm{Prox}_h^\delta(w) = \arg\min_x \left\{\frac{1}{2\delta}\|x - w\|^2 + h(x)\right\}.$$

However, when $B \neq \tau I$ (e.g., B is not a diagonal matrix), it is often hard to solve the problem (9) in practice [32]. To address this issue, we use the inexact Uzawa method [53] and design the following linearized proximal rule

$$y_k^s = \arg\min_y \left\{h(y) + \frac{\beta}{2}\left\|Az_{k-1}^s + By - c + \lambda_{k-1}^s\right\|^2 + \frac{\theta_{s-1}}{2\eta}\|y - y_{k-1}^s\|_{Q_s}^2\right\},$$

where $Q_s = \nu I_{d_2} - \frac{\eta\beta}{\theta_{s-1}} B^T B$ with $\nu \ge 1 + \frac{\eta\beta\|B^T B\|_2}{\theta_{s-1}}$ to ensure that $Q_s \succeq I$, where $\|\cdot\|_2$ is the spectral norm, i.e., the largest singular value of the matrix. The above problem is equivalent to the following problem:

$$y_k^s = \arg\min_y \left\{ h(y) + \frac{\nu\theta_{s-1}}{2\eta} \left\| y - y_{k-1}^s + \frac{\eta\beta}{\theta_{s-1}\nu} p_k^s \right\|^2 \right\}, \tag{10}$$

where $p_k^s = B^T(Az_{k-1}^s + By_{k-1}^s - c + \lambda_{k-1}^s)$. We can easily obtain the following proximal update rule for Problem (10):

$$y_k^s = \mathrm{Prox}_h^{\frac{\eta}{\theta_{s-1}\nu}} \Big[y_{k-1}^s - \frac{\eta\beta}{\theta_{s-1}\nu} B^T(Az_{k-1}^s + By_{k-1}^s - c + \lambda_{k-1}^s) \Big].$$

From the above analysis, it is clear that we introduce the linearized proximal operation into the proposed algorithms (including Algorithms 1 and 2 below) and make our algorithms much more practical than existing stochastic ADMM algorithms including SVRG-ADMM [22] and the algorithms proposed in [49]. Then the new algorithms proposed in this paper as well as their convergence analysis are different from those in our previous work [49]. In particular, the convergence guarantees of the proposed algorithms become more challenging.

To ensure linear convergence of the proposed linearized proximal algorithm for strongly convex problems as SVRG-ADMM, we also design the following new epoch initialization strategy for y_0^s at each epoch of Algorithm 1 for the general case of $B \neq \tau I$

$$y_0^s = -B^{\dagger}(Az_0^s - c), \tag{11}$$

where B is required to be a matrix of full column rank, and $(\cdot)^{\dagger}$ denotes the pseudo-inverse of a matrix. Note that for the case of $B = \tau I$, we use the epoch initialization settings (e.g., $y_0^s = \widetilde{y}^{s-1}$ and $\widetilde{\lambda}^{s-1} = -\frac{1}{\beta}(A^T)^{\dagger}\nabla f(\widetilde{x}^{s-1})$) in our previous work [49] to guarantee linear convergence of Algorithm 1, where the snapshot points $\widetilde{y}^{s-1}$ and $\widetilde{x}^{s-1}$ are defined in Algorithm 1, while only these settings cannot guarantee the convergence of our algorithm for the general case of $B \neq \tau I$. Therefore, we present the new initialization setting of y_0^s in Eq. (11) instead of $y_0^s = \widetilde{y}^{s-1}$. That is, the only difference in the initialization settings at the s-epoch for the two cases is the setting of y_0^s. Clearly, the epoch initialization techniques involve the pseudo-inverses of A^T and B. In fact, they can be efficiently pre-computed by the algorithms in [54], [55], especially randomized algorithms with only $O(n)$ complexity.

3.1.2 Update Rule of z

z is an auxiliary variable, and its update rule is given as follows. Similar to [19], [22], we also use the inexact Uzawa method [53] to approximate (5), which can avoid computing the inverse of the matrix $(\frac{1}{\eta}I_{d_1} + \beta A^T A)$. Moreover, the momentum parameter θ_s ($0 \leq \theta_s \leq 1$ and its update rule is provided in Section 3.1.4) is introduced into the proximal term $\frac{1}{2\eta}\|z - z_{k-1}^s\|_{G_s}^2$ similar to that of (7), and then the problem with respect to z is formulated as follows:

$$\min_z \left\{ \left\langle z - z_{k-1}^s,\, \widetilde{\nabla} f_{I_k}(x_{k-1}^s) \right\rangle + \frac{\theta_{s-1}}{2\eta}\|z - z_{k-1}^s\|_{G_s}^2 + \frac{\beta}{2}\|Az + By_k^s - c + \lambda_{k-1}^s\|^2 \right\}, \tag{12}$$

where $\widetilde{\nabla} f_{I_k}(x_{k-1}^s)$ is the stochastic variance reduced gradient estimator independently introduced in [18], [56], and $G_s = \gamma I_{d_1} - \frac{\eta\beta}{\theta_{s-1}} A^T A$ with $\gamma \geq 1 + \frac{\eta\beta\|A^T A\|_2}{\theta_{s-1}}$ to ensure that $G_s \succeq I$ similar to [22]. In fact, there is also an alternative to set G_s as an identity matrix, and then the problem (12) can be solved through matrix inversion [9], [19].

3.1.3 Our Momentum Accelerated Update Rule for x

In particular, our momentum accelerated update rule for x is defined as follows:

$$x_k^s = \widetilde{x}^{s-1} + \theta_{s-1}(z_k^s - \widetilde{x}^{s-1}) = (1 - \theta_{s-1})\widetilde{x}^{s-1} + \theta_{s-1} z_k^s, \tag{13}$$

where $\theta_{s-1}(z_k^s - \widetilde{x}^{s-1})$ is a new momentum term similar to those as in accelerated deterministic methods [27], which helps accelerate the convergence speed of our algorithms by using the iterate of the previous epoch, i.e., $\widetilde{x}^{s-1}$. Note that θ_{s-1} is a momentum parameter, and its update rule is given below. The momentum term, $\theta_{s-1}(z_k^s - \widetilde{x}^{s-1})$, plays a key role as the Katyusha momentum in [29]. Different from Katyusha [29], which uses both the Nesterov's momentum and Katyusha momentum, our ASVRG-ADMM algorithms (including Algorithms 1 and 2 below) have only one momentum term.

3.1.4 Momentum Parameter θ_s

In all epochs of Algorithm 1, the momentum parameter θ_s can be set to a constant θ, which must satisfy the condition $0 \leq \theta \leq 1 - \delta(b)/(\alpha - 1)$, where $\alpha = \frac{1}{L\eta}$ and $\delta(b) = \frac{n-b}{b(n-1)}$. In particular, we also provide the selecting schemes for the momentum parameter θ and corresponding theoretical analysis for the two cases of $B = \tau I$ and $B \neq \tau I$, which all are presented in the Supplementary Material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.3000512.

The detailed procedure for solving the strongly convex problem (1) is shown in Algorithm 1, where we use the same epoch initialization technique for $\widetilde{\lambda}^s$ as in [22]. Similar to x_k^s, $\widetilde{y}^s = (1 - \theta_{s-1})\widetilde{y}^{s-1} + \frac{\theta_{s-1}}{m}\sum_{k=1}^m y_k^s$. When $\theta = 1$, ASVRG-ADMM degenerates to the linearized proximal variant of SVRG-ADMM in [22], as shown in the Supplementary Material, available online.

3.2 ASVRG-ADMM for Non-Strongly Convex Problems

In this part, we consider the non-strongly convex (non-SC) problems of the form (1) when each $f_i(\cdot)$ is convex, L-smooth, and $h(\cdot)$ is not necessarily strongly convex (possibly non-smooth), e.g., graph-guided fused Lasso. The detailed procedure for solving the non-SC problem (1) is shown in Algorithm 2, which has slight differences in the initialization and output of each epoch from Algorithm 1. In addition, the key difference between them is the update rule for the momentum parameter θ_s. Different from the strongly convex case, the momentum parameter θ_s for the non-SC case is required to satisfy the following inequalities:

$$\frac{1 - \theta_s}{\theta_s^2} = \frac{1}{\theta_{s-1}^2} \ \text{ and } \ 0 \leq \theta_s \leq 1 - \frac{\delta(b)}{\alpha - 1}, \tag{14}$$

where $\delta(b) := \frac{n-b}{b(n-1)}$ is a decreasing function with respect to the mini-batch size b. The condition (14) allows the momentum parameter to decrease, but not too fast, similar to the requirement on the step-size η_k in classical SGD and stochastic ADMM [57]. Unlike deterministic acceleration methods, θ_s must satisfy both inequalities in (14).

Motivated by the momentum acceleration techniques in [27], [58] for deterministic optimization, we give the update rule of the momentum parameter θ_s for the mini-batch case

$$\theta_s = \frac{\sqrt{\theta_{s-1}^4 + 4\theta_{s-1}^2} - \theta_{s-1}^2}{2} \quad \text{and} \quad \theta_0 = 1 - \frac{\delta(b)}{\alpha - 1}. \tag{15}$$

For the special case of $b=1$, we have $\delta(1)=1$ and $\theta_0 = 1 - \frac{1}{\alpha-1}$, while $b=n$ (i.e., the deterministic version), $\delta(n)=0$ and $\theta_0 = 1$. Since the sequence $\{\theta_s\}$ is decreasing, $\theta_s \leq 1 - \frac{\delta(b)}{\alpha-1}$ is satisfied. That is, θ_s in Algorithm 2 is adaptively adjusted as in (15).

4 Convergence Analysis

In this section, we theoretically analyze the convergence properties of our ASVRG-ADMM algorithms (i.e., Algorithms 1 and 2) for SC and non-SC problems with the cases of $B \neq \tau I$ and $B = \tau I$, respectively. We first make the following assumption for the case of SC problems.

Assumption 3. *The matrices A and B^T both have full row rank.*

The first two assumptions (i.e., Assumptions 1 and 2) are common in the convergence analysis of first-order optimization methods, while the last one (i.e., Assumption 3) has been used in the convergence analysis of deterministic ADMM [7], [59], [60] and stochastic ADMM [22] for only the strongly convex case. Following [22], we first introduce the following function as a convergence criterion, where $h'(y)$[2] is the (sub)gradient of $h(\cdot)$ at y

$$\begin{aligned} P(x,y) := \quad & f(x) - f(x^*) - \langle \nabla f(x^*), x - x^* \rangle \\ & + h(y) - h(y^*) - \langle h'(y^*), y - y^* \rangle, \end{aligned}$$

where (x^*, y^*) denotes an optimal solution of Problem (1). By the convexity of $f(\cdot)$ and $h(\cdot)$, $P(x,y) \geq 0$ for all $x, y \in \mathbb{R}^d$.

Note that we present a new linearized proximal technique in (10) to update y_k^s, and thus we need to provide new convergence guarantees for our algorithms (i.e., Algorithms 1 and 2), which are different from those in our previous work [49]. Next, we present five main theoretical results for the convergence properties of Algorithms 1 and 2. And the detailed proofs of all the theoretical results are provided in this paper or the Supplementary Material, available online.

We first sketch the proofs of our main theoretical results as follows: The proofs of our main results rely on the one-epoch inequalities in Lemma 4 ($B \neq \tau I$) below and Lemma 7 ($B = \tau I$) in the Supplementary Material, available online. That is, the proofs of Theorems 1-5 below rely on the one-epoch inequalities in Lemmas 4 and 7, but require telescoping such inequalities in different manners. Furthermore, $P(x,y)$ in Lemma 4 consists of two terms, and thus we give the upper bounds of the two terms in Lemmas 2 and 3 to obtain Lemma 4, as well as applying Lemmas 2 and 6 to get Lemma 7 in the Supplementary Material, available online. In addition, to remove the strong assumption used in Theorems 3 and 4, we also design an adaptive strategy of increasing epoch length for Algorithm 2, and the corresponding theoretical result is given in Theorem 5, which shows that Algorithm 2 with an adaptive increasing epoch length attains the same convergence rate without the boundedness assumption.

2. Note that $\nabla f(x)$ is the gradient of a smooth function $f(\cdot)$ at x, while $h'(y)$ denotes a subgradient (or the gradient) of a non-smooth (or smooth) function $h(\cdot)$ at y.

Algorithm 2. ASVRG-ADMM for Non-SC Problems

Input: $m, \eta, \beta > 0, 1 \leq b \leq n$.
Initialize: $\widetilde{x}^0 = z_0^1, y_0^1 = \widetilde{y}^0, \widetilde{\lambda}^0, \theta_0 = 1 - \frac{L\eta\delta(b)}{1-L\eta}$.
for $s = 1, 2, \ldots, T$ **do**
  $x_0^s = (1-\theta_{s-1})\widetilde{x}^{s-1} + \theta_{s-1} z_0^s$, $\lambda_0^s = \widetilde{\lambda}^{s-1}$;
  $\widetilde{p} = \nabla f(\widetilde{x}^{s-1})$, $\nu = 1 + \frac{\eta\beta\|B^T B\|_2}{\theta_{s-1}}$, $\gamma = 1 + \frac{\eta\beta\|A^T A\|_2}{\theta_{s-1}}$;
  for $k = 1, 2, \ldots, m$ **do**
    Choose $I_k \subseteq [n]$ of size b, uniformly at random;
    $\widetilde{\nabla} f_{I_k}(x_{k-1}^s) = \frac{1}{|I_k|}\sum_{i_k \in I_k}\left[\nabla f_{i_k}(x_{k-1}^s) - \nabla f_{i_k}(\widetilde{x}^{s-1})\right] + \widetilde{p}$;
    $y_k^s = \mathrm{Prox}_h^{\frac{1}{\beta\tau^2}}\left((-Az_{k-1}^s + c - \lambda_{k-1}^s)/\tau\right)$ for the case of $B = \tau I$,
    $y_k^s = \mathrm{Prox}_h^{\frac{\eta}{\nu\theta_{s-1}}}\left[y_{k-1}^s - \frac{\eta\beta B^T(Az_{k-1}^s + By_{k-1}^s - c + \lambda_{k-1}^s)}{\nu\theta_{s-1}}\right]$ for the case of $B \neq \tau I$;
    $z_k^s = z_{k-1}^s - \frac{\eta}{\gamma\theta_{s-1}}\left[\widetilde{\nabla} f_{I_k}(x_{k-1}^s) + \beta A^T(Az_{k-1}^s + By_k^s - c + \lambda_{k-1}^s)\right]$;
    $x_k^s = (1-\theta_{s-1})\widetilde{x}^{s-1} + \theta_{s-1} z_k^s$;
    $\lambda_k^s = \lambda_{k-1}^s + Az_k^s + By_k^s - c$;
  end for
  $\widetilde{x}^s = \frac{1}{m}\sum_{k=1}^m x_k^s$, $\widetilde{y}^s = (1-\theta_{s-1})\widetilde{y}^{s-1} + \frac{\theta_{s-1}}{m}\sum_{k=1}^m y_k^s$;
  $\widetilde{\lambda}^s = \lambda_m^s$, $y_0^{s+1} = y_m^s$, $z_0^{s+1} = z_m^s$, $\theta_s = \frac{\sqrt{\theta_{s-1}^4 + 4\theta_{s-1}^2} - \theta_{s-1}^2}{2}$;
end for
Output: $\widetilde{x}^T, \widetilde{y}^T$.

4.1 Key Lemmas

In this part, we give and prove some intermediate key results for our convergence analysis.

Lemma 1.

$$\begin{aligned} &\mathbb{E}[\|\widetilde{\nabla} f_{I_k}(x_{k-1}^s) - \nabla f(x_{k-1}^s)\|^2] \\ \leq & 2L\delta(b)\Big[f(\widetilde{x}^{s-1}) - f(x_{k-1}^s) + \langle \nabla f(x_{k-1}^s, x_{k-1}^s - \widetilde{x}^{s-1}\rangle\Big], \end{aligned}$$

where $\delta(b) = \frac{n-b}{b(n-1)} \leq 1$ and $1 \leq b \leq n$.

The proofs of Lemmas 1, 2 and all the theorems below are provided in the Supplementary Material, available online. Lemma 1 provides an upper bound on the expected variance of the mini-batch SVRG estimator $\widetilde{\nabla} f_{I_k}(x_{k-1}^s)$.

Lemma 2. *Let (x^*, y^*) be an optimal solution of Problem (1), and λ^* be the corresponding Lagrange multiplier that maximizes the dual. Let $\varphi_k^s = \beta(\lambda_k^s - \lambda^*)$, and suppose that each $f_i(\cdot)$ is L-smooth. If the inequality $1 - \theta_{s-1} \geq \frac{\delta(b)}{\alpha-1}$ is satisfied, then*

$$\begin{aligned} &\mathbb{E}[f(\widetilde{x}^s) - f(x^*) - \langle \nabla f(x^*),\ \widetilde{x}^s - x^* \rangle] \\ &- \mathbb{E}\left[\frac{\theta_{s-1}}{m}\sum_{k=1}^m \left\langle A^T\varphi_k^s,\ x^* - z_k^s \right\rangle\right] \\ \leq\ & (1-\theta_{s-1})\mathbb{E}\left[f(\widetilde{x}^{s-1}) - f(x^*) - \langle \nabla f(x^*),\ \widetilde{x}^{s-1} - x^* \rangle\right] \\ &+ \frac{\theta_{s-1}^2}{2m\eta}\mathbb{E}\left[\|x^* - z_0^s\|_{G_s}^2 - \|x^* - z_m^s\|_{G_s}^2\right]. \end{aligned}$$

For the case of $B \neq \tau I$, we have the following result, which corresponds to Lemma 7 in the Supplementary Material, available online, for the case of $B = \tau I$.

Lemma 3. *Let $\{(\widetilde{y}^s, y_k^s)\}$ be the sequence generated by Algorithm 1 (or Algorithm 2), we have*

$$\begin{aligned}
&\mathbb{E}[h(\widetilde{y}^s) - h(y^*) - \langle h'(y^*),\, \widetilde{y}^s - y^*\rangle] \\
&\quad - \frac{\theta_{s-1}}{m}\sum_{k=1}^{m}\mathbb{E}\left[\langle B^T\varphi_k^s,\; y^* - y_k^s\rangle\right] \\
&\leq (1-\theta_{s-1})\mathbb{E}\left[h(\widetilde{y}^{s-1}) - h(y^*) - \langle h'(y^*),\, \widetilde{y}^{s-1} - y^*\rangle\right] \\
&\quad + \frac{\beta\theta_{s-1}}{2m}\mathbb{E}\left[\|Az_0^s - Ax^*\|^2 - \|Az_m^s - Ax^*\|^2 + \sum_{k=1}^{m}\|\lambda_k^s - \lambda_{k-1}^s\|^2\right] \\
&\quad + \frac{\theta_{s-1}^2}{2m\eta}\mathbb{E}\left[\|y^* - y_0^s\|_{Q_s}^2 - \|y^* - y_m^s\|_{Q_s}^2\right].
\end{aligned}$$

Since a new linearized proximal rule is proposed to update the variable y for Algorithms 1 and 2 in the case of $B \neq \tau I$, we need to give the following proof for Lemma 3, which is different from Lemma 6 in the Supplementary Material, available online, for the case of $B = \tau I$.

Proof. Since $\lambda_k^s = \lambda_{k-1}^s + Az_k^s + By_k^s - c$, and using the optimality condition of Problem (10) (i.e., $h'(y_k^s) + \beta B^T(Az_{k-1}^s + By_k - c + \lambda_{k-1}) + \frac{\theta_{s-1}}{\eta}Q_s(y_k^s - y_{k-1}^s) = 0$), we have

$$\begin{aligned}
&h(y_k^s) - h(y^*) \\
&\leq \langle h'(y_k^s),\; y_k^s - y^*\rangle \\
&= \left\langle \beta B^T(Az_{k-1}^s + By_k - c + \lambda_{k-1}) + \frac{\theta_{s-1}Q_s(y_k^s - y_{k-1}^s)}{\eta},\, y^* - y_k^s\right\rangle \\
&= \left\langle \beta B^T\lambda_k^s + \frac{\theta_{s-1}Q_s}{\eta}(y_k^s - y_{k-1}^s),\; y^* - y_k^s\right\rangle \\
&\quad + \left\langle \beta B^T(Az_{k-1}^s - Az_k^s),\; y^* - y_k^s\right\rangle \\
&= \frac{\beta}{2}\langle B^T\lambda_k^s,\; y^* - y_k^s\rangle + \frac{\theta_{s-1}}{2\eta}\left(\|y^* - y_{k-1}^s\|_{Q_s}^2 - \|y^* - y_k^s\|_{Q_s}^2\right) \\
&\quad + \frac{\beta}{2}\left(\|Az_{k-1}^s - Ax^*\|^2 - \|Az_k^s - Ax^*\|^2 + \|\lambda_k^s - \lambda_{k-1}^s\|^2\right),
\end{aligned}$$

where the last equality follows from $Ax^* + By^* = c$ and Property 1 in the Supplementary Material, available online. Taking expectation over the random choice of i_k, we have

$$\begin{aligned}
&\mathbb{E}\left[h(y_k^s) - h(y^*) - \langle h'(y^*), y_k^s - y^*\rangle - \langle B^T\varphi_k^s,\; y^* - y_k^s\rangle\right] \\
&\leq \frac{\beta}{2}\mathbb{E}\left[\|Az_{k-1}^s - Ax^*\|^2 - \|Az_k^s - Ax^*\|^2\right] \\
&\quad + \frac{1}{2}\mathbb{E}\left[\beta\|\lambda_k^s - \lambda_{k-1}^s\|^2 + \frac{\theta_{s-1}}{\eta}\left(\|y^* - y_{k-1}^s\|_{Q_s}^2 - \|y^* - y_k^s\|_{Q_s}^2\right)\right].
\end{aligned}$$

Using the update rule of $\widetilde{y}^s = (1-\theta_{s-1})\widetilde{y}^{s-1} + \frac{\theta_{s-1}}{m}\sum_{k=1}^m y_k^s$, $h(\widetilde{y}^s) \leq (1-\theta_{s-1})h(\widetilde{y}^{s-1}) + \frac{\theta_{s-1}}{m}\sum_{k=1}^m h(y_k^s)$, and taking expectation over whole history and summing up the above inequality for all $k = 1, \ldots, m$, we have

$$\begin{aligned}
&\mathbb{E}\left[h(\widetilde{y}^s) - h(y^*) - \langle h'(y^*), \widetilde{y}^s - y^*\rangle - \frac{\theta_{s-1}}{m}\sum_{k=1}^{m}\langle B^T\varphi_k^s, y^* - y_k^s\rangle\right] \\
&\leq \frac{\theta_{s-1}}{m}\mathbb{E}\left[\sum_{k=1}^{m}\left(h(y_k^s) - h(y^*) + \langle h'(y^*) - \theta_{s-1}B^T\varphi_k^s, y^* - y_k^s\rangle\right)\right] \\
&\quad + \mathbb{E}\left[\frac{\theta_{s-1}^2}{2\eta}\left(\|y^* - y_{k-1}^s\|_{Q_s}^2 - \|y^* - y_k^s\|_{Q_s}^2\right)\right] \\
&\quad + (1-\theta_{s-1})\mathbb{E}\left[h(\widetilde{y}^{s-1}) - h(y^*) - \langle h'(y^*), \widetilde{y}^{s-1} - y^*\rangle\right] \\
&\leq \frac{\beta\theta_{s-1}}{2m}\mathbb{E}\left[\|Az_0^s - Ax^*\|^2 - \|Az_m^s - Ax^*\|^2 + \sum_{k=1}^{m}\|\lambda_k^s - \lambda_{k-1}^s\|^2\right] \\
&\quad + (1-\theta_{s-1})\mathbb{E}\left[h(\widetilde{y}^{s-1}) - h(y^*) - \langle h'(y^*), \widetilde{y}^{s-1} - y^*\rangle\right] \\
&\quad + \frac{\theta_{s-1}^2}{2\eta}\left[\|y^* - y_{k-1}^s\|_{Q_s}^2 - \|y^* - y_k^s\|_{Q_s}^2\right].
\end{aligned}$$

This completes the proof. □

For the case of $B \neq \tau I$, we also have the following one-epoch inequality, which is a key lemma to prove Theorems 2, 4 and 5 below and is corresponding to Lemma 7 in the Supplementary Material, available online, for the case of $B = \tau I$, and Lemma 7 is also a main result to prove Theorems 1, 3 and 6 below.

Lemma 4 (One-epoch Upper Bound). *Using the same notation as in Lemma 2, let $\{(z_k^s, x_k^s, y_k^s, \lambda_k^s, \widetilde{x}^s, \widetilde{y}^s)\}$ be the sequence generated by Algorithm 1 (or Algorithm 2) with $\theta_s \leq 1 - \frac{\delta(b)}{\alpha-1}$. Then the following inequality holds for all k*

$$\begin{aligned}
&\mathbb{E}\left[P(\widetilde{x}^s, \widetilde{y}^s) - \frac{\theta_{s-1}}{m}\sum_{k=1}^{m}\left((x^* - z_k^s)^T A^T\varphi_k^s + (y^* - y_k^s)^T B^T\varphi_k^s\right)\right] \\
&\leq \mathbb{E}\left[\frac{P(\widetilde{x}^{s-1}, \widetilde{y}^{s-1})}{1/(1-\theta_{s-1})} + \frac{\theta_{s-1}^2}{2m\eta}\left(\|x^* - z_0^s\|_{G_s}^2 - \|x^* - z_m^s\|_{G_s}^2\right)\right] \\
&\quad + \frac{\beta\theta_{s-1}}{2m}\mathbb{E}\left[\|Az_0^s - Ax^*\|^2 - \|Az_m^s - Ax^*\|^2 + \sum_{k=1}^{m}\|\lambda_k^s - \lambda_{k-1}^s\|^2\right] \\
&\quad + \frac{\theta_{s-1}^2}{2m\eta}\mathbb{E}\left[\mathcal{R}^s - \|y^* - y_m^s\|_{Q_s}^2\right],
\end{aligned}$$

where $\mathcal{R}^s$ is defined as follows:

$$\mathcal{R}^s = \begin{cases} \sigma\|Ax^* - Az_0^s\|^2, & \text{if } f(x) \text{ is SC}, \\ \|y^* - y_0^s\|_{Q_s}^2, & \text{if } f(x) \text{ is non-SC} \end{cases}, \tag{16}$$

and $\sigma = \|B^\dagger\|_2^2(\frac{2\eta\beta\|B^TB\|_2}{\theta_{s-1}} + 1)$.

Proof. Using Lemmas 2 and 3 and the definition of $P(x, y)$, we have

$$\begin{aligned}
&\mathbb{E}\left[P(\widetilde{x}^s, \widetilde{y}^s) - \frac{\theta_{s-1}}{m}\sum_{k=1}^{m}\left((x^* - z_k^s)^T A^T\varphi_k^s + (y^* - y_k^s)^T B^T\varphi_k^s\right)\right] \\
&\leq \mathbb{E}\left[\frac{P(\widetilde{x}^{s-1}, \widetilde{y}^{s-1})}{1/(1-\theta_{s-1})} + \frac{\theta_{s-1}^2\left(\|x^* - z_0^s\|_{G_s}^2 - \|x^* - z_m^s\|_{G_s}^2\right)}{2m\eta}\right] \\
&\quad + \frac{\beta\theta_{s-1}}{2m}\mathbb{E}\left[\|Az_0^s - Ax^*\|^2 - \|Az_m^s - Ax^*\|^2 + \sum_{k=1}^{m}\|\lambda_k^s - \lambda_{k-1}^s\|^2\right] \\
&\quad + \frac{\theta_{s-1}^2}{2m\eta}\mathbb{E}\left[\|y^* - y_0^s\|_{Q_s}^2 - \|y^* - y_m^s\|_{Q_s}^2\right].
\end{aligned}$$

When $f(\cdot)$ is μ-strongly convex and $Ax^* + By^* = c$, we have $y^* = B^\dagger(c - Ax^*)$. Using the update rule of $y_0^s =$

$B^{\dagger}(c - Az_0^s)$ and $\nu = 1 + \frac{\eta\beta\|B^T B\|_2}{\theta_{s-1}}$, we have

$$\begin{aligned}\|y^* - y_0^s\|_{Q_s}^2 &= \|B^{\dagger}(Az_0^s - Ax^*)\|_{Q_s}^2 \\ &\leq \|B^{\dagger}\|_2^2 \|Az_0^s - Ax^*\|_{Q_s}^2 \\ &\leq \|B^{\dagger}\|_2^2 \left\|\nu I - \frac{\eta}{\theta_{s-1}} B^T B\right\|_2 \|Az_0^s - Ax^*\|^2 \\ &\leq \|B^{\dagger}\|_2^2 \Big(\frac{2\eta\beta\|B^T B\|_2}{\theta_{s-1}} + 1\Big)\|Az_0^s - Ax^*\|^2.\end{aligned}$$

Therefore, the result of Lemma 4 holds. □

4.2 Linear Convergence

For Algorithm 1, we first give the following results for the two cases of $B = \tau I$ and $B \neq \tau I$, respectively.

Theorem 1 (Case of $B = \tau I$). *Using the same notation as in Lemma 2 with $\theta \leq 1 - \frac{\delta(b)}{\alpha - 1}$, suppose that $f(\cdot)$ is μ-strongly convex, each $f_i(\cdot)$ is L-smooth and Assumption 3 holds, and m is sufficiently large so that*

$$\rho_1 = \underbrace{\frac{\theta\|\theta G + \eta\beta A^T A\|_2}{\eta m \mu}}_{1} + \underbrace{(1-\theta)}_{2} + \underbrace{\frac{L\theta}{\beta m \sigma_{\min}(AA^T)}}_{3} < 1, \tag{17}$$

where $\sigma_{\min}(AA^T)$ is the smallest eigenvalue of the positive semi-definite matrix AA^T, and $G_s \equiv G$ as in Eq. (12). Then

$$\mathbb{E}\big[P(\widetilde{x}^T, \widetilde{y}^T)\big] \leq \rho_1^T P(\widetilde{x}^0, \widetilde{y}^0).$$

The theoretical result in our previous work [49] can be viewed as the special case of Theorem 1 when $B = I$. From Theorem 1, we can see that ASVRG-ADMM achieves linear convergence, which is consistent with that of SVRG-ADMM, while SCAS-ADMM has only an $\mathcal{O}(1/T)$ convergence rate.

Remark 1. Theorem 1 shows that our result improves slightly upon the rate ρ_1 in SVRG-ADMM [22] with the same η and β. Specifically, ρ_1 in Eq. (17) consists of three components, which are corresponding to those of Theorem 1 in [22]. In Algorithm 1, recall that $\theta \leq 1$ and G is defined in Eq. (12). Thus, the upper bound of Eq. (17) is slightly smaller than that of Theorem 1 in [22]. In particular, we can set $\eta = 1/8L$ (i.e., $\alpha = 8$) and $\theta = 1 - \delta(b)/(\alpha - 1) = 1 - \delta(b)/7$. Therefore, the second term in Eq. (17) equals to $\delta(b)/7$, while that of SVRG-ADMM is approximately equal to $4L\eta\delta(b)/(1 - 4L\eta\delta(b)) \geq \delta(b)/2$. In summary, the convergence speed of SVRG-ADMM can be slightly improved by ASVRG-ADMM.

Theorem 2 (Case of $B \neq \tau I$). *Using the same notation as in Lemma 2 with $\theta \leq 1 - \frac{\delta(b)}{\alpha - 1}$, suppose that $f(\cdot)$ is μ-strongly convex, each $f_i(\cdot)$ is L-smooth and Assumption 3 holds, and m is sufficiently large so that*

$$\rho_2 = \frac{\theta\varrho}{\eta m \mu} + (1-\theta) + \frac{L\theta}{\beta m \sigma_{\min}(AA^T)} < 1, \tag{18}$$

where $\varrho = \|\theta G + \eta(\beta + \theta\sigma)A^T A)\|_2$. Then

$$\mathbb{E}\big[P(\widetilde{x}^T, \widetilde{y}^T)\big] \leq \rho_2^T P(\widetilde{x}^0, \widetilde{y}^0).$$

From Theorem 2, ASVRG-ADMM can also achieve linear convergence for the more complex ADMM-style problem (1) with $B \neq \tau I$. It is not hard to see that the convergence rate ρ_2 in Theorem 2 is slightly larger than that (i.e., ρ_1) of Theorem 1, meaning slow convergence for more complex optimization problems, as verified by our experiments.

4.3 Convergence Rate of $\mathcal{O}(1/T^2)$

We first assume that $y \in \mathcal{Y}$ and $z \in \mathcal{Z}$, where $\mathcal{Y}$ and $\mathcal{Z}$ are the convex compact sets with diameters $D_{\mathcal{Y}} = \sup_{y_1, y_2 \in \mathcal{Y}} \|y_1 - y_2\|$ and $D_{\mathcal{Z}} = \sup_{z_1, z_2 \in \mathcal{Z}} \|z_1 - z_2\|$, respectively, and $D_{\Lambda} = \sup_{\lambda_1, \lambda_2 \in \Lambda} \|\lambda_1 - \lambda_2\|$. The above assumption is called the boundedness assumption. We also denote $D_{x^*} = \|\widetilde{x}^0 - x^*\|$, $D_{y^*} = \|\widetilde{y}^0 - y^*\|$ and $D_{\lambda^*} = \|\widetilde{\lambda}^0 - \lambda^*\|$, where $(\widetilde{x}^0, \widetilde{y}^0, \widetilde{\lambda}^0)$ are initial points, (x^*, y^*) is an optimal solution of Problem (1) and λ^* is the corresponding dual variable. The boundedness of D_{x^*}, D_{y^*} and D_{λ^*} are easily satisfied, which is called the basic conditions in this paper.

For Algorithm 2, we give the following results for the cases of $B = \tau I$ and $B \neq \tau I$, respectively, whose proofs are provided in the Supplementary Material, available online.

Theorem 3 (Case of $B = \tau I$). *Let ς be a positive constant, suppose that each $f_i(\cdot)$ is L-smooth, $\mathcal{Z}$ and Λ are the convex compact sets with diameters $D_{\mathcal{Z}}$ and D_{Λ}, then*

$$\begin{aligned}&\mathbb{E}\big[P(\widetilde{x}^T, \widetilde{y}^T) + \varsigma\|A\widetilde{x}^T + \tau\widetilde{y}^T - c\|\big] \\ &\leq \frac{4(\alpha-1)\delta(b)(P(\widetilde{x}^0, \widetilde{y}^0) + \varsigma\|A\widetilde{x}^0 + \tau\widetilde{y}^0 - c\|)}{(\alpha - 1 - \delta(b))^2 (T+1)^2} \\ &\quad + \frac{2L\alpha D_{x^*}^2}{m(T+1)^2} + \frac{4\alpha\beta(\|A^T A\|_2 D_{\mathcal{Z}}^2 + D_{\Lambda}^2)}{m(\alpha-1)(T+1)}.\end{aligned} \tag{19}$$

Remark 2. With $m = \Theta(n)$, Theorem 3 shows that the convergence bound consists of the three components, which converge as $\mathcal{O}(1/T^2)$, $\mathcal{O}(1/nT^2)$ and $\mathcal{O}(1/nT)$, respectively, while the three components of SVRG-ADMM converge as $\mathcal{O}(1/T)$, $\mathcal{O}(1/nT)$ and $\mathcal{O}(1/nT)$. Clearly, ASVRG-ADMM achieves the convergence rate of $\mathcal{O}(1/T^2)$ as opposed to $\mathcal{O}(1/T)$ of SVRG-ADMM and SAG-ADMM ($m \gg T$ in general). All the components in the bound of SCAS-ADMM converge as $\mathcal{O}(1/T)$. Thus, it is clear that ASVRG-ADMM is at least a factor T faster than existing stochastic ADMM algorithms including SAG-ADMM, SVRG-ADMM and SCAS-ADMM. Theorem 3 shows that the convergence result in our previous work [49] can be viewed as the special case of Theorem 3. In addition, Theorems 3 and 4 below require the boundedness assumption and the basic conditions (i.e., D_{x^*}, D_{y^*} and D_{λ^*} are bounded by some constants).

Theorem 4 (Case of $B \neq \tau I$). *Using the same notation as in Lemma 2, and suppose that each $f_i(\cdot)$ is L-smooth, and $\mathcal{Y}$, $\mathcal{Z}$ and Λ are the convex compact sets with diameters $D_{\mathcal{Y}}$, $D_{\mathcal{Z}}$ and D_{Λ}, then we have*

$$\begin{aligned}&\mathbb{E}\big[P(\widetilde{x}^T, \widetilde{y}^T) + \varsigma\|A\widetilde{x}^T + B\widetilde{y}^T - c\|\big] \\ &\leq \frac{4(\alpha-1)\delta(b)(P(\widetilde{x}^0, \widetilde{y}^0) + \varsigma\|A\widetilde{x}^0 + B\widetilde{y}^0 - c\|)}{(\alpha - 1 - \delta(b))^2 (T+1)^2} \\ &\quad + \frac{2\alpha\beta\big(2\|A^T A\|_2 D_{\mathcal{Z}}^2 + \|B^T B\|_2 D_{\mathcal{Y}}^2 + 2D_{\Lambda}^2\big)}{m(\alpha-1)(T+1)} \\ &\quad + \frac{2L\alpha(D_{x^*}^2 + D_{y^*}^2)}{m(T+1)^2}.\end{aligned} \tag{20}$$

4.4 $\mathcal{O}(1/T^2)$ Without Boundedness Assumption

The result in Theorem 4 shows that ASVRG-ADMM attains the optimal convergence rate $O(1/T^2)$ for the non-SC problem (1) with $B \neq \tau I$. Compared with SVRG-ADMM and SAG-ADMM, ASVRG-ADMM attains a better convergence rate for non-SC problems, but with the price on the boundedness of the feasible primal sets $\mathcal{Z}$, $\mathcal{Y}$, and the feasible dual set Λ. Note that many previous works such as [61], [62] also require such assumptions of boundedness when proving the convergence of ADMMs. In order to remove the strong assumption and further improve our theoretical results, we design an adaptive strategy of increasing epoch length, i.e., $m_{s+1} = \lceil \frac{\theta_{s-1}}{\theta_s} m_s \rceil$, while a constant epoch length m is used in original Algorithm 2. The increasing epoch length strategy is similar to that in [63], that is, $m_{s+1} = \lceil \frac{\theta_{s-1}}{\theta_s} m_s \rceil$ instead of $m_{s+1} = 2m_s$ in [63]. By replacing the epoch length m in Algorithm 2 with m_s, we can obtain the following improved theoretical result. It should be noted that the increasing factor $\frac{\theta_{s-1}}{\theta_s}$ approaches 1 as the number of epochs increases, which means that the epoch length increases very slowly. Below we only present the convergence result for the general case of $B \neq \tau I$, the theoretical result for the case of $B = \tau I$ and the detailed proofs for all the results are provided in the Supplementary Material, available online.

Theorem 5 (Without boundedness assumption). *Using the same notation as in Lemma 2, suppose that each $f_i(\cdot)$ is L-smooth. Let $\{(\widetilde{x}^s, \widetilde{y}^s, \widetilde{\lambda}^s)\}$ be the sequence generated by Algorithm 2 with our adaptive increasing epoch length strategy for the case of $B \neq \tau I$, then*

$$\begin{aligned}
&\mathbb{E}\left[P(\widetilde{x}^T, \widetilde{y}^T) + \varsigma\|A\widetilde{x}^T + B\widetilde{y}^T - c\|\right] \\
&\leq \frac{4(\alpha-1)\delta(b)(P(\widetilde{x}^0, \widetilde{y}^0) + \varsigma\|A\widetilde{x}^0 + B\widetilde{y}^0 - c\|)}{(\alpha - 1 - \delta(b))^2 (T+1)^2} \\
&\quad + \frac{2(\alpha-1)\beta\left(2\|A^T A\|_2 D_{x^*}^2 + \|B^T B\|_2 D_{y^*}^2 + 2D_{\lambda^*}^2\right)}{(\alpha - 1 - \delta(b)) m_1 (T+1)^2} \\
&\quad + \frac{2L\alpha(D_{x^*}^2 + D_{y^*}^2)}{m_1 (T+1)^2}.
\end{aligned} \tag{21}$$

Remark 3. With the setting $m_1 = \Theta(n/T)$ to guarantee the same overall complexity with the original algorithm, Theorem 5 shows that ASVRG-ADMM with our adaptive epoch length strategy obtains the rate of $\mathcal{O}(1/T^2)$. The upper bound only relies on the constants D_{x^*}, D_{y^*} and D_{λ^*}, while the theoretical result in Theorem 4 requires that $\mathcal{Y}$, $\mathcal{Z}$ and Λ are all bounded with the diameters $D_{\mathcal{Y}}$, $D_{\mathcal{Z}}$ and D_Λ. That is, ASVRG-ADMM with our adaptive epoch length strategy achieves the convergence rate $\mathcal{O}(1/T^2)$ without the boundedness assumption.

4.5 Discussion

All our algorithms and convergence results can be extended to the following settings: When the mini-batch size $b = n$ and $m = 1$, then $\delta(n) = 0$, that is, the first term of both (19) and (20) vanishes, and ASVRG-ADMM degenerates to the deterministic two-block[3] ADMM version [64]. The convergence rate of (20) becomes $\mathcal{O}\Big(\frac{D_{x^*}^2 + D_{y^*}^2}{(T+1)^2} + \frac{D_{\mathcal{Z}}^2 + D_{\mathcal{Y}}^2 + D_\Lambda^2}{T+1}\Big)$, which is consistent with the result for accelerated deterministic ADMM [34], [37]. Many empirical risk minimization problems can be viewed as the special case of Problem (2) when $A = I$. Thus, our method can be extended to solve them, and has an $\mathcal{O}(1/T^2 + 1/(nT^2))$ rate, which is consistent with the best-known result as in [29], [30].

3. Note that the formulation (1) is called two-block because of the two sets of variables (x, y), which are updated alternately.

TABLE 2
Summary of All the Real-World Data Sets Used in Our Experiments

Data sets	♯ training	♯ test	♯ mini-batch
a9a	16,281	16,280	20
epsilon	400,000	100,000	30
w8a	32,350	32,350	20
20newsgroups	13,000	3,242	15
SUSY	3,500,000	1,500,000	100
HIGGS	7,700,000	3,300,000	150

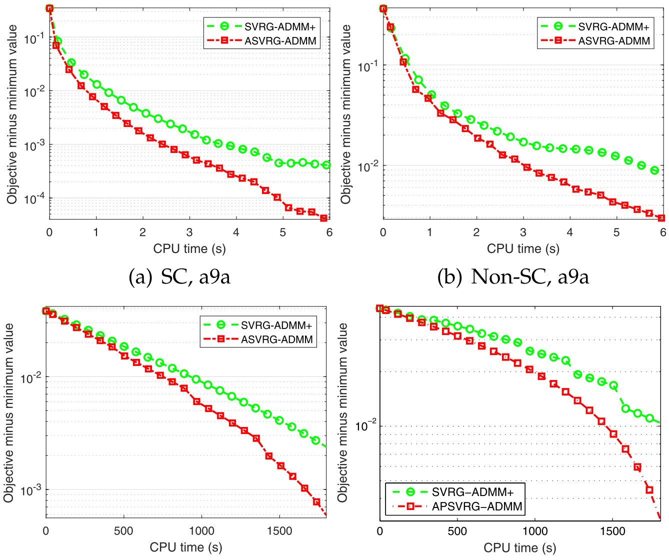
5 Experimental Results

In this section, we apply ASVRG-ADMM to solve various machine learning problems, e.g., non-SC graph-guided fused Lasso, SC and non-SC graph-guided logistic regression, and SC graph-guided SVM problems. We compare ASVRG-ADMM with the state-of-the-art methods: STOC-ADMM [9], OPG-ADMM [45], SAG-ADMM [19], SCAS-ADMM [21] and SVRG-ADMM [22]. All methods were implemented in MATLAB, and the experiments were performed on a PC with an Intel i5-2400 CPU and 16GB RAM.

5.1 Synthetic Data

In this subsection, we evaluate the empirical performance of the proposed algorithms for solving both SC and non-SC problems (1) on some synthetic data. Here, each $f_i(x)$ is the logistic loss function on the feature-label pair (a_i, b_i), i.e., $f_i(x) = \log(1 + \exp(-b_i a_i^T x))$ $(i = 1, 2, \ldots, n)$ for the non-SC case and $f_i(x) = \log(1 + \exp(-b_i a_i^T x)) + \frac{\lambda_2}{2}\|x\|_2^2$ for the SC case, where $\lambda_2 \geq 0$ is a regularization parameter. We used a relatively small data set, a9a (about 733K), and a relatively large data set, epsilon (about 11G), as listed in Table 2. Since the original SVRG-ADMM [22] cannot be used to solve the minimization problem (i.e., $\min_{x,y}\{\frac{1}{2}\sum_{i=1}^n f_i(x) + \lambda_1\|y\|_1\}$) with the constraint $Ax + By = c$, where $B \neq \tau I$ and $\lambda_1 \geq 0$ is a regularization parameter, we also present its linearized proximal variant (called SVRG-ADMM+), as shown in the Supplementary Material, available online. Note that the constraint matrix A is set to $A = [G; I]$ as in [9], [19], [22], [62], where G is the sparsity pattern of the graph obtained by sparse inverse covariance selection [65], while both B and c are randomly generated. In particular, the generated matrix B has full column rank, but is not an identity matrix.

Fig. 1 shows the training loss (i.e., the training objective value minus the minimum value) of SVRG-ADMM+ and ASVRG-ADMM for solving both the SC and non-SC problems, where the regularization parameters $\lambda_1 = \lambda_2 = 10^{-4}$. All the experimental results show that our ASVRG-ADMM method (i.e., Algorithms 1 and 2) converges consistently much faster than SVRG-ADMM+, which empirically verifies our theoretical results of ASVRG-ADMM.



(a) SC, a9a (b) Non-SC, a9a

(c) SC, epsilon (d) Non-SC, epsilon

Fig. 1. Comparison of the linearized proximal SVRG-ADMM and our ASVRG-ADMM algorithms for both SC and non-SC problems on the two data sets: a9a (top) and epsilon (bottom).

5.2 Real-World Applications

In this subsection, we apply our ASVRG-ADMM method to solve a number of real-world machine learning problems such as graph-guided fused Lasso, graph-guided logistic regression, graph-guided SVM, generalized graph-guided logistic regression and multi-task learning.

5.2.1 Graph-Guided Fused Lasso

We evaluate the empirical performance of ASVRG-ADMM for solving the non-SC graph-guided fused Lasso problem

$$\min_{x,y}\left\{\frac{1}{n}\sum_{i=1}^{n} f_i(x)+\lambda_1\|y\|_1,\ \text{s.t.}, Ax=y\right\}, \tag{22}$$

where $f_i(\cdot)$ is the logistic loss function on the feature-label pair (a_i, b_i), i.e., $f_i(x)=\log(1+\exp(-b_i a_i^T x))$, and $\lambda_1 \geq 0$ is the regularization parameter. As in [9], [22], [62], we set $A=[G; I]$, where G is the sparsity pattern of the graph obtained by sparse inverse covariance selection [65]. We used four publicly available data sets[4] in our experiments, as listed in Table 2. The parameter m, as well as b and β, of ASVRG-ADMM is set to $m=\lceil 2n/b\rceil$ as in [19], [22]. All the other algorithms except STOC-ADMM adopted the linearization of the penalty term $\frac{\beta}{2}\|Ax-y_k+\lambda_{k-1}\|^2$ to avoid the inversion of $\frac{1}{\eta_k}I_{d_1}+\beta A^T A$ at each iteration, which can be computationally expensive for large matrices.

Fig. 2 shows the training loss (i.e., the training objective value minus the minimum value) and test error of all the algorithms for the non-SC problem (22) with the regularization parameter $\lambda_1=10^{-5}$ on the four data sets. SAG-ADMM could not generate experimental results on the HIGGS data set because it ran out of memory. These figures clearly indicate that the variance reduced stochastic ADMM algorithms (i.e., SAG-ADMM, SCAS-ADMM, SVRG-ADMM and ASVRG-ADMM) converge much faster than those without variance reduction techniques, e.g., STOC-ADMM and OPG-ADMM. In particular, ASVRG-ADMM consistently outperforms the other algorithms in terms of convergence speed in all the settings, which empirically verifies our theoretical result that ASVRG-ADMM has a faster convergence rate $\mathcal{O}(1/T^2)$, as opposed to the best-known rate of $\mathcal{O}(1/T)$. Moreover, the test error of ASVRG-ADMM is consistently better than those of the other methods.

5.2.2 Graph-Guided Logistic Regression

We also discuss the performance of ASVRG-ADMM for the SC graph-guided logistic regression problem

$$\min_{x,y}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(f_i(x)+\frac{\lambda_2}{2}\|x\|^2\right)+\lambda_1\|y\|_1,\ \text{s.t.}, Ax=y\right\}. \tag{23}$$

Due to limited space and similar experimental phenomena on the four data sets, we only report the experimental results on the a9a and w8a data sets in Fig. 3, where $\lambda_1=10^{-5}$ and $\lambda_2=10^{-2}$. We can see that SVRG-ADMM and ASVRG-ADMM achieve comparable performance, and they significantly outperform the other methods in terms of convergence speed, which is consistent with their linear (geometric) convergence guarantees. Moreover, ASVRG-ADMM converges slightly faster than SVRG-ADMM, which shows the effectiveness of the proposed momentum trick to accelerate variance reduced stochastic ADMM, as we expected.

5.2.3 Graph-Guided SVM

We also evaluate the performance of ASVRG-ADMM for solving the SC graph-guided SVM problem

$$\min_{x,y}\left\{\frac{1}{n}\sum_{i=1}^{n}\left([1-b_i a_i^T x]_+ +\frac{\lambda_2}{2}\|x\|_2^2\right)+\lambda_1\|y\|_1\right\},\quad \text{s.t.}, Ax=y, \tag{24}$$

where $[x]_+=\max(0,x)$ is the non-smooth hinge loss. To effectively solve (24), we use the smooth Huberized hinge loss in [66] to approximate the hinge loss. For the 20newsgroups data set,[5] we randomly divide it into 80 percent training set and 20 percent test set. Following [9], we set $\lambda_1=\lambda_2=10^{-5}$, and use the one-versus-rest scheme for the multi-class classification.

Fig. 4 shows the average prediction accuracies and standard deviations of testing accuracies over 10 different runs. Since STOC-ADMM, OPG-ADMM, SAG-ADMM and SCAS-ADMM consistently perform worse than SVRG-ADMM and ASVRG-ADMM in all settings, we only report the results of STOC-ADMM. We can see that SVRG-ADMM and ASVRG-ADMM consistently outperform the classical SVM and STOC-ADMM. Moreover, ASVRG-ADMM performs much better than the other methods in all settings, which further verifies the effectiveness of ASVRG-ADMM.

4. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

5. http://www.cs.nyu.edu/~roweis/data.html

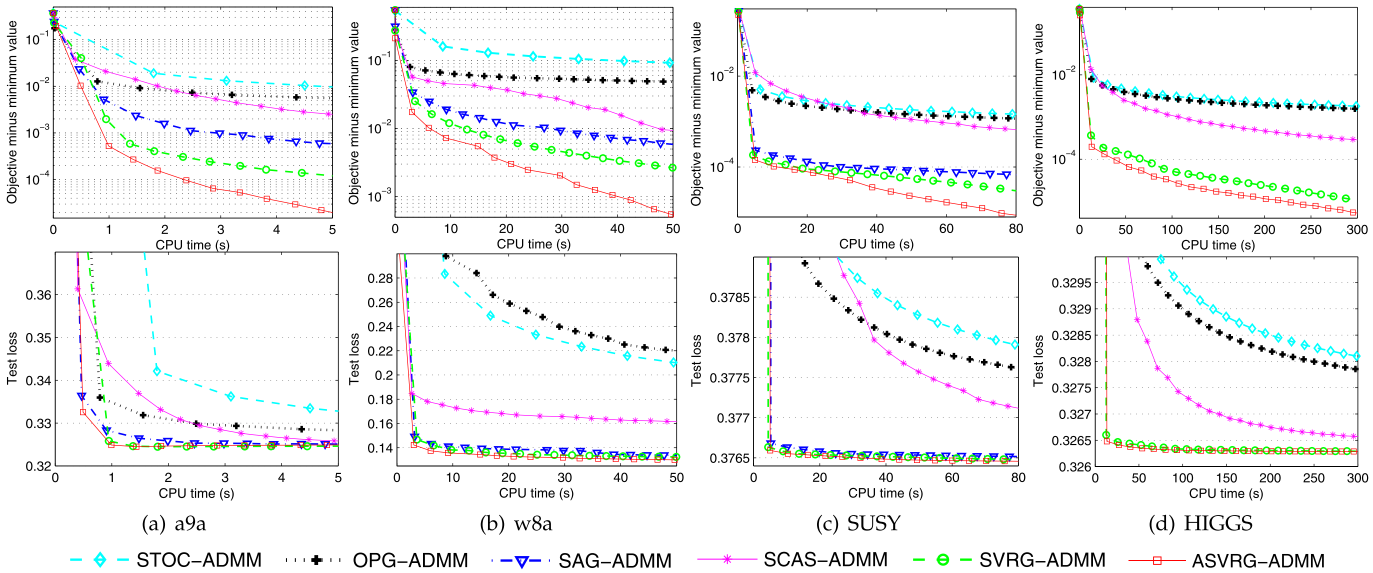


Fig. 2. Comparison of different stochastic ADMM methods for non-SC graph-guided fused Lasso problems on the four data sets. The y-axis represents the objective value minus the minimum value (top) or test loss (bottom), and the x-axis corresponds to the running time (seconds).

5.2.4 Generalized Graph-Guided Logistic Regression

Moreover, we apply ASVRG-ADMM to solve the non-SC graph-guided logistic regression problem as in [67]

$$\min_{x,y}\left\{\frac{1}{n}\sum_{i=1}^{n} f_i(x)+\lambda_1\|x\|_1+\lambda_2\|y\|_1,\ \text{s.t.}, Ax=y\right\}. \tag{25}$$

All the problems in (22), (23) and (24) can be cast as the form (2), while Problem (25) can be cast as the form (1), i.e., $\min_{x,v}\{f(x)+\|v\|_1, \text{s.t.}\, Cx+Bv=0\}$, where $v=[\lambda_1 z^T, \lambda_2 y^T]^T$ and z are slack variables, $C=[I_{d_x}, A^T]^T, B=-\begin{bmatrix}\frac{1}{\lambda_1}I_{d_x} & 0\\ 0 & \frac{1}{\lambda_2}I_{d_y}\end{bmatrix}$.

The experimental results on the a9a data set are shown in Fig. 5, from which we can see that SVRG-ADMM+ and ASVRG-ADMM converge significantly faster than STOC-ADMM+. Note that SVRG-ADMM+ and STOC-ADMM+ are the linearized proximal variants of SVRG-ADMM and STOC-ADMM. Moreover, ASVRG-ADMM outperforms them in terms of both convergence speed and test error, which shows the effectiveness of our momentum trick to accelerate variance reduced stochastic ADMM.

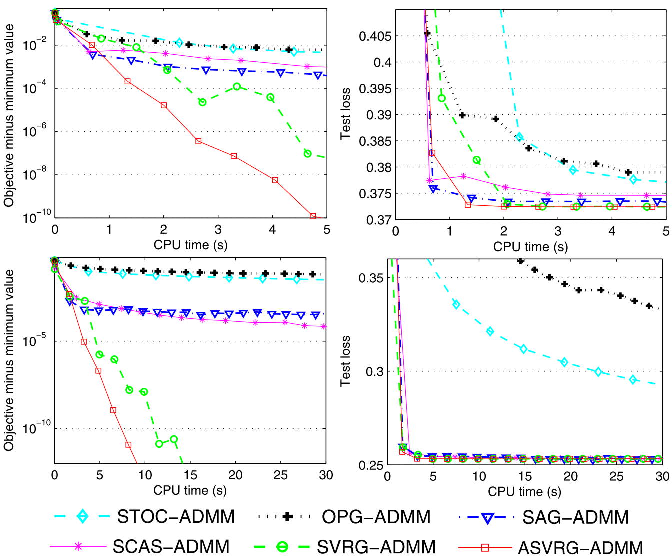


Fig. 3. Comparison of the stochastic ADMM methods for SC graph-guided logistic regression problems on a9a (top) and w8a (bottom).

5.2.5 Multi-Task Learning

Finally, we consider the multi-task learning problem and can cast it as the non-SC constrained problem: $\min_{X,Y}\{\sum_{i=1}^{N} f_i(X)+\lambda_1\|Y\|_*, \text{s.t.}, X=Y\}$, where $X,Y\in\mathbb{R}^{d\times N}$, N is the number of tasks, $f_i(X)$ is the multinomial logistic loss on the ith task, and $\|Y\|_*$ is the nuclear norm. The experimental results in Fig. 6 show that ASVRG-ADMM outperforms the

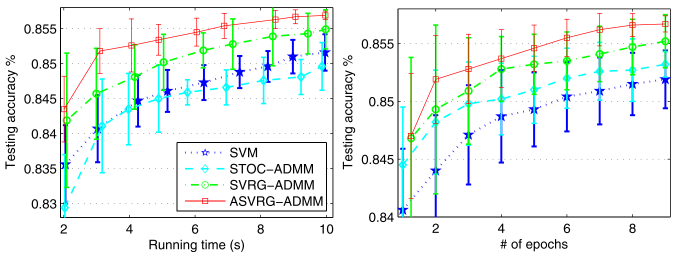


Fig. 4. Accuracy comparison of multi-class classification on 20newsgroups: accuracy versus running time (left) or number of epochs (right).

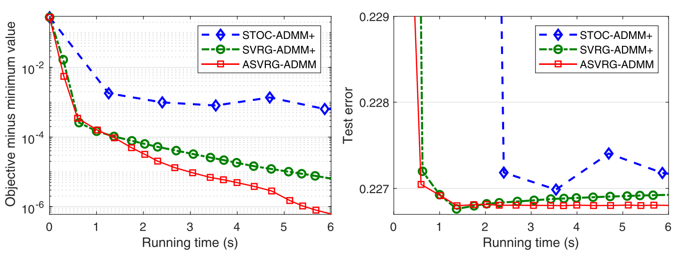


Fig. 5. Comparison of all the methods for generalized graph-guided fused Lasso on a9a, where regularization parameters $\lambda_1=\lambda_2=10^{-5}$.

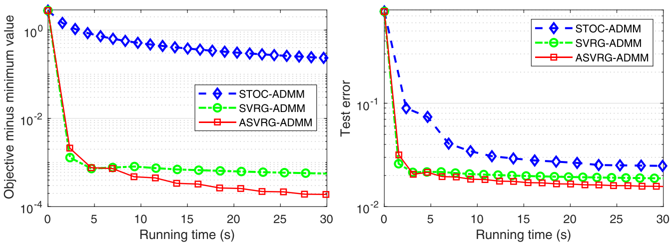


Fig. 6. Comparison of all the methods for multi-task learning problems on 20newsgroups, where the regularization parameter $\lambda_1 = 10^{-4}$.

other methods including SVRG-ADMM in terms of both convergence speed and test error.

6 Conclusions and Further Work

In this paper, we proposed an efficient accelerated stochastic variance reduced ADMM (ASVRG-ADMM) method, in which we combined both our proposed momentum acceleration trick and the variance reduction stochastic ADMM [22]. We also designed two different update rules for the general ADMM (i.e., $B \neq \tau I$) and special ADMM (i.e., $B = \tau I$) problems, respectively. That is, we presented a new linearized proximal scheme for the case of $B \neq \tau I$, and adopted a simple proximal scheme in our previous work [49] for the case of $B = \tau I$. Moreover, we theoretically analyzed the convergence properties of the proposed linearized proximal accelerated SVRG-ADMM algorithms, which show that ASVRG-ADMM achieves linear convergence and $\mathcal{O}(1/T^2)$ rates for strongly convex and non-strongly convex cases, respectively. In particular, ASVRG-ADMM is at least a factor T faster than existing stochastic ADMM methods for non-strongly convex problems.

Our empirical study showed that the convergence speed of ASVRG-ADMM is much faster than those of the state-of-the-art stochastic ADMM methods such as SVRG-ADMM. We can apply our proposed momentum acceleration trick to accelerate existing incremental gradient descent algorithms such as [68], [69] for solving regularized empirical risk minimization problems. An interesting direction of future work is the research of our proposed momentum acceleration trick for accelerating incremental gradient descent ADMM algorithms such as SAG-ADMM [19] and SAGA-ADMM [52]. In addition, it is also interesting to extend our algorithms and theoretical results from the two-block version to the multi-block ADMM case [70].

Acknowledgments

The author would like to thank all the reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China (Nos. 61876221, 61876220, 61976164, 61836009, U1701267, 61871310, and 61801353), the Project supported the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61621005), the Major Research Plan of the National Natural Science Foundation of China (Nos. 91438201, and 91438103), the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT_15R53), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048), the Science Foundation of Xidian University (Nos. 10251180018 and 10251180019), and the National Science Basic Research Plan in Shaanxi Province of China (Nos. 2019JQ-657 and 2020JM-194). Z. Lin was supported by NSF China (Grant nos. 61625301 and 61731018), Major Scientific Research Project of Zhejiang Lab (Grant nos. 2019KB0AC01 and 2019KB0AB02), Beijing Academy of Artificial Intelligence, and Qualcomm.

References

[1] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[2] W. Zhang *et al.*, "Sparse learning with stochastic composite optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1223–1236, Jun. 2017.

[3] S. Kim, K. A. Sohn, and E. P. Xing, "A multivariate regression approach to association analysis of a quantitative trait network," *Bioinformatics*, vol. 25, pp. i204–i212, 2009.

[4] R. J. Tibshirani and J. Taylor, "The solution path of the generalized lasso," *Ann. Statist.*, vol. 39, no. 3, pp. 1335–1371, 2011.

[5] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[6] C. Lu, J. Feng, S. Yan, and Z. Lin, "A unified alternating direction method of multipliers by majorization minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 527–541, Mar. 2018.

[7] F. Shang, J. Cheng, Y. Liu, Z.-Q. Luo, and Z. Lin, "Bilinear factor matrix norm minimization for robust PCA: Algorithms and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2066–2080, Sep. 2018.

[8] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, pp. 231–358, 2015.

[9] H. Ouyang, N. He, L. Q. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 80–88.

[10] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.

[11] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Math. Doklady*, vol. 27, pp. 372–376, 1983.

[12] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, pp. 125–161, 2013.

[13] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," University of Washington, 2008.

[14] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[15] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 919–926.

[16] C. Hu, J. T. Kwok, and W. Pan, "Accelerated gradient methods for stochastic optimization and online learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 781–789.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[18] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 315–323.

[19] L. W. Zhong and J. T. Kwok, "Fast stochastic alternating direction method of multipliers," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 46–54.

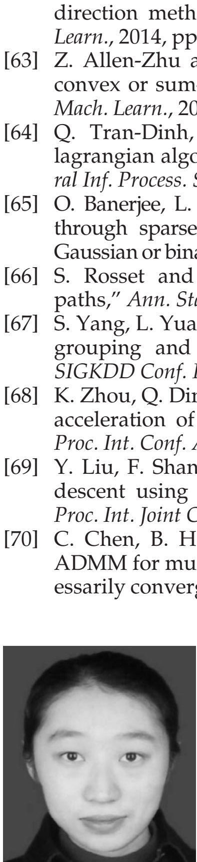
[20] T. Suzuki, "Stochastic dual coordinate ascent with alternating direction method of multipliers," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 736–744.

[21] S.-Y. Zhao, W.-J. Li, and Z.-H. Zhou, "Scalable stochastic alternating direction method of multipliers," 2015, *arXiv:1502.03529v3*.

[22] S. Zheng and J. T. Kwok, "Fast-and-light stochastic ADMM," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2407–2613.

[23] N. L. Roux, M. Schmidt, and F. Bach, "A stochastic gradient method with an exponential convergence rate for finite training sets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 2672–2680.

[24] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," *J. Mach. Learn. Res.*, vol. 14, pp. 567–599, 2013.
[25] L. Xiao and T. Zhang, "A proximal stochastic gradient method with progressive variance reduction," *SIAM J. Optim.*, vol. 24, no. 4, pp. 2057–2075, 2014.
[26] F. Shang *et al.*, "VR-SGD: A simple stochastic variance reduction method for machine learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 1, pp. 188–202, Jan. 2020.
[27] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Boston, MA, USA: Kluwer, 2004.
[28] A. Nitanda, "Stochastic proximal gradient descent with acceleration techniques," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1574–1582.
[29] Z. Allen-Zhu, "Katyusha: The first direct acceleration of stochastic gradient methods," *J. Mach. Learn. Res.*, vol. 18, no. 221, pp. 1–51, 2018.
[30] L. T. K. Hien, C. Lu, H. Xu, and J. Feng, "Accelerated stochastic mirror descent algorithms for composite non-strongly convex optimization," *J. Optim. Theory Appl.*, vol. 181, no. 2, pp. 541–566, 2019.
[31] F. Shang, Y. Liu, J. Cheng, K. W. Ng, and Y. Yoshida, "Guaranteed sufficient decrease for stochastic variance reduced gradient optimization," in *Proc. 21st Int. Conf. Artif. Intell. Statist.*, 2018, pp. 1027–1036.
[32] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 612–620.
[33] F. Nie, Y. Huang, X. Wang, and H. Huang, "Linear time solver for primal SVM," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 505–513.
[34] C. Lu, H. Li, Z. Lin, and S. Yan, "Fast proximal linearized alternating direction method of multiplier with parallel splitting," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 739–745.
[35] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Math. Program.*, vol. 162, pp. 165–199, 2017.
[36] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[37] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM J. Imag. Sci.*, vol. 7, no. 3, pp. 1588–1623, 2014.
[38] J. Eckstein and W. Yao, "Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives," *Pacific J. Optim.*, vol. 11, pp. 619–644, 2015.
[39] H. Li and Z. Lin, "Accelerated alternating direction method of multipliers: An optimal $O(1/K)$ nonergodic analysis," *J. Sci. Comput.*, vol. 79, no. 2, pp. 671–699, 2019.
[40] M. Kadkhodaie, K. Christakopoulou, M. Sanjabi, and A. Banerjee, "Accelerated alternating direction method of multipliers," in *Proc. SIGKDD Conf. Knowl. Discov. Data Mining*, 2015, pp. 497–506.
[41] D. Davis and W. Yin, "Faster convergence rates of relaxed Peaceman-Rachford and ADMM under regularity assumptions," *Math. Oper. Res.*, vol. 42, no. 3, pp. 783–805, 2017.
[42] W. Tian and X. Yuan, "An alternating direction method of multipliers with a worst-case $O(1/n^2)$ convergence rate," *Math. Comput.*, vol. 88, pp. 1685–1713, 2019.
[43] G. Franca, D. P. Robinson, and R. Vidal, "ADMM and accelerated ADMM as continuous dynamical systems," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1554–1562.
[44] H. Wang and A. Banerjee, "Online alternating direction method," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1119–1126.
[45] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 392–400.
[46] Y. Yu and L. Huang, "Fast stochastic variance reduced ADMM for stochastic composition optimization," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 3364–3370.
[47] Y. Xu, M. Liu, Q. Lin, and T. Yang, "ADMM without a fixed penalty parameter: Faster convergence with new adaptive penalization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1267–1277.
[48] C. Fang, F. Cheng, and Z. Lin, "Faster and non-ergodic $O(1/K)$ stochastic alternating direction method of multipliers," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4479–4488.
[49] Y. Liu, F. Shang, and J. Cheng, "Accelerated variance reduced stochastic ADMM," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2287–2293.
[50] J. Koneeny, J. Liu, P. Richtarik, and M. Takae, "Mini-batch semi-stochastic gradient descent in the proximal setting," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 242–255, Mar. 2016.
[51] F. Huang, S. Chen, and H. Huang, "Faster stochastic alternating direction method of multipliers for nonconvex optimization," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2839–2848.
[52] F. Huang and S. Chen, "Mini-batch stochastic ADMMs for nonconvex nonsmooth optimization," 2019, *arXiv: 1802.03284v3*.
[53] X. Zhang, M. Burger, and S. Osher, "A unified primal-dual algorithm framework based on Bregman iteration," *J. Sci. Comput.*, vol. 46, no. 1, pp. 20–46, 2011.
[54] G. H. Golub and C. F. V. Loan, *Matrix Computions*. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2013.
[55] M. W. Mahoney, "Randomized algorithms for matrices and data," *Found. Trends Mach. Learn.*, vol. 3, no. 2, pp. 123–224, 2011.
[56] L. Zhang, M. Mahdavi, and R. Jin, "Linear convergence with condition number independent access of full gradients," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 980–988.
[57] P. Tseng, "An incremental gradient(-projection) method with momentum term and adaptive step size rule," *SIAM J. Optim.*, vol. 8, no. 2, pp. 506–531, 1998.
[58] P. Tseng, "Approximation accuracy, gradient methods, and error bound for structured convex optimization," *Math. Program.*, vol. 125, pp. 263–295, 2010.
[59] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A general analysis of the convergence of ADMM," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 343–352.
[60] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *J. Sci. Comput.*, vol. 66, pp. 889–916, 2016.
[61] B. He and X. Yuan, "Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective," *SIAM J. Imag. Sci.*, vol. 5, no. 1, pp. 119–149, 2012.
[62] S. Azadi and S. Sra, "Towards an optimal stochastic alternating direction method of multipliers," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 620–628.
[63] Z. Allen-Zhu and Y. Yuan, "Improved SVRG for non-strongly-convex or sum-of-non-convex objectives," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1080–1089.
[64] Q. Tran-Dinh, "Non-ergodic alternating proximal augmented lagrangian algorithms with optimal rates," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4816–4824.
[65] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *J. Mach. Learn. Res.*, vol. 9, pp. 485–516, 2008.
[66] S. Rosset and J. Zhu, "Piecewise linear regularized solution paths," *Ann. Statist.*, vol. 35, no. 3, pp. 1012–1030, 2007.
[67] S. Yang, L. Yuan, Y.-C. Lai, X. Shen, P. Wonka, and J. Ye, "Feature grouping and selection over an undirected graph," in *Proc. SIGKDD Conf. Knowl. Discov. Data Mining*, 2012, pp. 922–930.
[68] K. Zhou, Q. Ding, F. Shang, J. Cheng, D. Li, and Z. Q. Luo, "Direct acceleration of SAGA using sampled negative momentum," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1602–1610.
[69] Y. Liu, F. Shang, and L. Jiao, "Accelerated incremental gradient descent using momentum acceleration with scaling factor," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 3045–3051.
[70] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Comput.*, vol. 155, pp. 57–79, 2016.



Yuanyuan Liu (Member, IEEE) received the PhD degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2013. She is currently a professor with the School of Artificial Intelligence, Xidian University, China. Prior to joining Xidian University, she was a postdoctoral research fellow with the Department of Computer Science and Engineering, Chinese University of Hong Kong. From 2013 to 2014, she was a postdoctoral research fellow with the Department of Systems Engineering and Engineering Management, CUHK. Her current research interests include machine learning, pattern recognition, and image processing.

Fanhua Shang (Senior Member, IEEE) received the PhD degree in circuits and systems from Xidian University, Xi'an, China, in 2012. He is currently a professor with the School of Artificial Intelligence, Xidian University, China. Prior to joining Xidian University, he was a research associate with the Chinese University of Hong Kong. From 2013 to 2015, he was a postdoctoral research fellow with the Chinese University of Hong Kong. From 2012 to 2013, he was a postdoctoral research associate with the Department of Electrical and Computer Engineering, Duke University. His current research interests include machine learning, data mining, pattern recognition, and computer vision.

Hongying Liu (Member, IEEE) received the BE and MS degrees in computer science and technology from the Xi'an University of Technology, Xi'an, China, in 2006 and 2009, respectively, and the PhD degree in engineering from Waseda University, Tokyo, Japan, in 2012. Currently, she is a faculty member with the School of Artificial Intelligence, and also with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, China. Her major research interests include image processing, intelligent signal processing, machine learning, etc.

Lin Kong received the BS degree in statistics from Xidian University, Xi'an, China, in 2019. She is currently working toward the master's degree with the School of Artificial Intelligence, Xidian University, Xi'an, China. Her current research interests include stochastic optimization for machine learning, large-scale machine learning, etc.

Licheng Jiao (Fellow, IEEE) received the BS degree from Shanghai Jiaotong University, Shanghai, China, in 1982, and the MS and PhD degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. He was a postdoctoral fellow with the National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an, from 1990 to 1991, where he has been a professor with the School of Electronic Engineering, since 1992, and currently the director of the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education of China. He has charged of about 40 important scientific research projects, and published more than 20 monographs and a hundred papers in international journals and conferences. His current research interests include image processing, natural computation, machine learning, and intelligent information processing. He is the chairman of Awards and Recognition Committee, the vice board chairperson of the Chinese Association of Artificial Intelligence, a councilor of the Chinese Institute of Electronics, a committee member of the Chinese Committee of Neural Networks, and an expert of the Academic Degrees Committee of the State Council.

Zhouchen Lin (Fellow, IEEE) received the PhD degree in applied mathematics from Peking University, Beijing, China, in 2000. He is currently a professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an area chair of CVPR 2014/2016/2019/2020/2021, ICCV 2015, NIPS 2015/2018/2019/2020, AAAI 2019/2020, IJCAI 2020, and ICML 2020. He is a fellow of the IAPR.