Loopless Variance Reduced Stochastic ADMM for Equality Constrained Problems in IoT Applications

Yuanyuan Liu, *Member, IEEE*, Jiacheng Geng, Fanhua Shang, *Senior Member, IEEE*, Weixin An, Hongying Liu, *Member, IEEE*, and Qi Zhu

***Abstract*—The alternating direction method of multipliers (ADMMs) is an efficient optimization method for solving equality constrained problems in Internet of Things (IoT) applications. Recently, several stochastic variance reduced ADMM algorithms (e.g., SVRG-ADMM) have made exciting progress, such as linear convergence for strongly convex (SC) problems. However, SVRG-ADMM and its variants have an outer loop where the full gradient at the snapshot is computed, and their outer loop contains an inner loop, in which a large number of variance reduced gradients are estimated from random samples. This loopy design makes these methods more complex to analyze and determine the inner loop length, which must be proportional to the condition number to achieve best convergence, and is often set to $\mathcal{O}(n)$ as a suboptimal choice, where n is the number of samples. To tackle these issues, we propose an efficient loopless variance reduced stochastic ADMM algorithm, called LVR-SADMM. In our LVR-SADMM, we remove the outer loop and replace it with a biased coin-flip, in which we update the snapshot with a small probability to trigger the full gradient computation. Moreover, we also theoretically analyze the convergence property of LVR-SADMM, which shows that it enjoys a fast linear convergence rate for SC problems. In particular, we also present an accelerated loopless SVRG-ADMM (LAVR-SADMM) method for both SC and non-SC problems. Various experimental results on many real-world data sets verify that the proposed methods can achieve an average speedup of 2× in the SC case and 5× in the non-SC case over their loopy counterparts, respectively.**

***Index Terms*—Alternating direction method of multipliers (ADMMs), loopless algorithm, momentum acceleration, stochastic variance reduced gradient (SVRG).**

I. Introduction

STOCHASTIC optimization has received significant interest in both academic and industrial communities in recent years [1], [2]. With the growth of the Internet of Things (IoT), it is increasingly important to be able to solve problems with a very large number of training examples or features. In this article, we mainly consider a class of composite convex minimization problems that are to minimize a finite-sum loss function [i.e., $(1/n)\sum_{i=1}^{n} f_i(x)$] and a regularizer $h(y)$ subject to an equality constraint, $Ax + By = c$. This class of problems covers a wide range of IoT applications in large-scale machine learning, data science, statistics, and operations research. For instance, one can obtain the regularized empirical risk minimization (ERM) problem as: $\min_x (1/n)\sum_{i=1}^{n} f_i(x) + h(x)$ by setting $A = I$, $B = -I$, and $c = 0$. By setting of the constraint $Ax = y$, one can obtain the ERM problem with a structured sparsity regularizer: $\min_{x,y}(1/n)\sum_{i=1}^{n} f_i(x) + h(y)$, s.t. $Ax = y$. Let $h(\cdot)$ be the ℓ_1-norm regularizer [i.e., $h(y) = \tau_1\|y\|_1$ with a regularized parameter τ_1], and if $f_i(\cdot)$ 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] with the ℓ_2-norm regularizer [i.e., $(\tau_2/2)\|x\|^2$], it becomes a graph-guided logistic regression (GGLR) problem [3], where $\tau_1, \tau_2 \geq 0$ are two regularization parameters. If $f_i(\cdot)$ is the *hinge loss* [i.e., $\max(0, 1 - b_i a_i^T x)$] with the ℓ_2-norm regularizer, one can obtain a *graph-guided SVM* problem [4]. Many other applications include sparse learning [5]–[7], matrix completion [8]–[11], and deep neural networks [12], [13].

To solve the above equality constrained composite optimization problems, the alternating direction method of multipliers (ADMMs) is an efficient optimization tool [14]. It was derived from the augmented Lagrangian method, whose core idea is to transform equality constrained problems into unconstrained ones. However, ADMM and its variants are deterministic methods and suffer a high computational cost per iteration for large-scale problems. Stochastic gradient descent (SGD) [15] uses only one or a small batch of random example(s) in each iteration to form an estimator of the full gradient, and thus, enjoys a significantly lower per-iteration cost than deterministic methods. In recent years, some efficient stochastic ADMM algorithms [4], [16], [17] have also been proposed. Analogous to SGD, they have a very low per-iteration computational complexity and are applicable for large-scale problems. However, the variance of the stochastic gradient estimator in both SGD and stochastic ADMMs

Manuscript received October 22, 2020; revised March 1, 2021 and June 13, 2021; accepted June 24, 2021. Date of publication July 8, 2021; date of current version January 24, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61876221, Grant 61876220, Grant 61976164, Grant 61836009, and Grant U1701267; in part by the Project supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61621005; in part by the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT_15R53; in part by the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) under Grant B07048; and in part by the National Science Basic Research Plan in Shaanxi Province of China under Grant 2020JM-194. *(Corresponding author: Fanhua Shang.)*

Yuanyuan Liu, Jiacheng Geng, Weixin An, Hongying Liu, and Qi Zhu are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: yyliu@xidian.edu.cn; hyliu@xidian.edu.cn).

Fanhua Shang is with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China, and also with Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: fhshang@xidian.edu.cn).

Digital Object Identifier 10.1109/JIOT.2021.3095561

may be large [12], which leads to poor performance and slow convergence.

In recent years, many variance reduction techniques, such as stochastic average gradient (SAG) [18], stochastic dual coordinate ascent (SDCA) [19], stochastic variance reduced gradient (SVRG) [12], and variance reduced SGD (VR-SGD) [20], have been introduced into stochastic ADMMs, and result in some variance reduced stochastic ADMM algorithms, such as SAG-ADMM [21], SDCA-ADMM [22], and SVRG-ADMM [23]. Among these methods, SVRG-ADMM is much more attractive than SAG-ADMM and SDCA-ADMM due to its significantly lower storage requirement compared with SAG-ADMM and SDCA-ADMM, which require storage of all the gradients of component functions or dual variables. Inspired by this, this article will propose an efficient loopless SVRG-ADMM algorithm for solving both strongly convex (SC) and nonstrongly convex (non-SC) problems.

More recently, a new momentum acceleration technique has been proposed in our previous work [2], [24], and has been incorporated into SVRG-ADMM to result in an accelerated SVRG-ADMM (ASVRG-ADMM). ASVRG-ADMM [24] inherits the linear convergence of SVRG-ADMM for SC problems and improves the convergence rate from $\mathcal{O}(1/K)$ to $\mathcal{O}(1/K^2)$ for non-SC problems, where K is the number of epochs or outer-loop iterations. Such an improved convergence result consequently fills the gap in the convergence rates between stochastic ADMMs and deterministic ADMMs. In this article, we will also propose an efficient loopless ASVRG-ADMM method for solving various real-world applications.

A. Motivations

A common structural feature of existing stochastic variance reduction methods (e.g., SVRG [12] and SVRG-ADMM [23]) and their accelerated versions such as ASVRG-ADMM [24] is the inclusion of an outer loop, where the exact full gradient at a snapshot point (also called a reference point) is first computed using a full pass through the training data, and each outer loop contains an inner loop, in which a variance reduced gradient estimator is constructed using the full gradient and new stochastic gradient information. That is, they are all two-stage optimization methods. However, this loopy design incurs the following issues. First, the loopy design makes the convergence analysis of the methods much more complex. As shown in the theoretical analysis and proofs in [23] and [24], we need to perform elaborate aggregations across the inner loop by summing up specific inequalities to construct the convergence criterions at two adjacent epochs to prove convergence. Second, we require to decide the termination of the inner loop. For the two-stage optimization methods, such as SVRG [12], SVRG-ADMM [23], and ASVRG-ADMM [24], to get best convergence, the optimal inner loop length m must be proportional to the condition number of the loss function [25], [26]. For SC problems, the condition number is defined as: L/μ, where L is the Lipschitz constant of the objective function and μ is its strong convexity parameter. However, μ is often unknown or hard to estimate. Even when we can estimate it in such regularized problems with an explicit SC regularizer, the estimate is often quite loose. Due to this fact, m is often set to $\mathcal{O}(n)$ as a suboptimal choice. As suggested in [12], $m = 2n$ for SVRG, and $m = \lfloor 2n/b \rfloor$ for SVRG-ADMM and ASVRG-ADMM, where b is the minibatch size.

Kovalev *et al.* [25] has proposed the loopless variants of SVRG [12] and Katyusha [27] (called L-SVRG and L-Katyusha, respectively), which remove the outer loop and replace it by a biased coin flip. The snapshot point is updated to the latest iterate with a small probability p and remains unchanged otherwise. This loopless structure makes the variants more intelligible and easier to analyze. In addition, the optimal probability p for L-SVRG can be made independent of the condition number according to its theoretical analysis. Moreover, [26] has further put forward the L-SVRG and L-Katyusha methods with arbitrary sampling and also extends L-SVRG to non-SC and nonconvex settings. All the exciting results motivate us to incorporate this loopless technique into existing well-known two-stage stochastic ADMM algorithms, such as SVRG-ADMM and ASVRG-ADMM, to address the issues mentioned above.

B. Our Main Contributions

This article proposes two new efficient loopless stochastic variance reduced ADMM algorithms to push toward faster convergence speed for equality constrained problems.

We summarize our main contributions as follows.

1) *Loopless Algorithms and Their Extensions:* To address the issues of SVRG-ADMM mentioned above, we propose a novel loopless variant of SVRG-ADMM, called loopless variance reduction stochastic ADMM (LVR-SADMM), for solving equality constrained minimization problems. We also extend our LVR-SADMM algorithm from the SC setting to the non-SC one, and also present its momentum accelerated variant (namely, LAVR-SADMM) for the two classes of problems.
2) *Slightly Smaller Variance Upper Bound:* As shown in our theoretical results in Section III-B, the upper bound on the expected variance of the stochastic gradient estimator for our LVR-SADMM is slightly smaller than that of the original SVRG-ADMM [23].
3) *Simpler Theoretical Analysis and New Convergence Criterion:* Compared with SVRG-ADMM, our theoretical analysis is notably simpler and more intuitive, as we simply use a single iteration analysis to establish convergence. In spite of its simplicity, our convergence criterion is novel, and our proofs are also new, rather than a trivial simplification of the loopy analysis in [23].
4) *Linear Convergence:* Theorem 1 in Section III-B indicates that our LVR-SADMM enjoys a fast linear convergence rate for SC problems, which is the same as its loopy counterpart, SVRG-ADMM. However, LVR-SADMM converges much faster in practice.
5) *Superior Practical Behavior:* We conduct extensive experiments on publicly available data sets to demonstrate that our loopless methods (LVR-SADMM and

TABLE I
NOTATIONS

Notation	Definition
(x^*, y^*, λ^*)	An optimal solution of the target problem
$h'(y)$	The (sub)gradient of function h at y
$\nabla f(x)$	The full gradient of f at x when f is differentiable
μ	The strong convexity parameter
L	The smoothness parameter
β	The penalty parameter
η	The step-size or learning rate
γ	A constant in the matrix G
p	The probability to update the snapshot point
b	The size of mini-batch
$\Vert a\Vert$	The Euclidean norm of a vector a
$\Vert A\Vert_2$	The spectral norm of a matrix A, i.e., the largest singular value of A
$(\cdot)^\dagger$	The pseudo-inverse
$\sigma_{\min}(A)$	The smallest eigenvalue of A
$Q\succeq I$	$Q-I$ is a positive semi-definite matrix, where Q is a matrix and I is an identity matrix
$\Vert x\Vert_Q^2$	x^TQx
Φ^k	The Lyapunov function
θ	The momentum weight

LAVR-SADMM) enjoy superior practical behavior than their loopy counterparts for both SC and non-SC problems. For the loopy methods, SVRG and SVRG-ADMM, to achieve best convergence, the optimal inner loop size depends on the condition number, which is usually unknown or hard to estimate correctly. Therefore, one often sets the inner loop size to be $\mathcal{O}(n)$ as a suboptimal choice. However, for our loopless methods, LVR-SADMM and LAVR-SADMM, with a flexible probability p, it is much easier to achieve better practical behavior than loopy methods as long as we choose p from the optimal interval to be independent of the condition number. This is the main cause that our loopless methods converge much faster than their loopy counterparts.

C. Roadmap

The remainder of this article is organized as follows. In Section II, we discuss some related work of stochastic ADMMs. Section III presents our LVR-SADMM algorithm and provides its convergence analysis. In Section IV, we develop an accelerated LVR-SADMM algorithm for both SC and non-SC cases. In Section V, we exhibit practical performance of LVR-SADMM and LAVR-SADMM for many SC and non-SC problems. In Section VI, we conclude this article and discuss future work.

II. Preliminaries

In this section, we review recent progresses and efforts in stochastic ADMM algorithms.

A. Notations and Common Assumptions

We summarize the notations used in this article in Table I. For a convex function $f : \mathbb{R}^d \to \mathbb{R}, f$ is μ-SC if there exists $\mu \geq 0$ such that $f(y) \geq f(x) + \langle\nabla f(x), y-x\rangle + (\mu/2)\|y-x\|^2$ for all $x, y \in \mathbb{R}^d$. f is L-smooth if there exists $L \geq 0$ such that $\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|$ for all $x, y \in \mathbb{R}^d$.

Before presenting our algorithms and convergence analysis, we give two common assumptions for the target problem.

Assumption 1: Each component function f_i is convex, continuously differentiable, and L_i-smooth.

Assumption 2: h is convex but not necessarily smooth.

B. Related Work

In this article, we mainly focus on the following minimization problem:

$$\min_{x\in\mathbb{R}^{d_1}, y\in\mathbb{R}^{d_2}} \quad \{f(x) + h(y), \text{ s.t. } Ax + By = c\} \tag{1}$$

where $A \in \mathbb{R}^{d_3\times d_1}$ and $B \in \mathbb{R}^{d_3\times d_2}$ are given matrices, $c \in \mathbb{R}^{d_3}$ is a constant vector, and $f(x) := (1/n)\sum_{i=1}^n f_i(x)$. Here, n is the number of training samples, $f_i(\cdot)$ is the loss function on sample i, and $h(\cdot)$ is a regularizer. In fact, problem (1) is the general form of the ADMM. When $B = -I$ and $c = 0$, the general constraint becomes $Ax = y$ as in GGLR and graph-guided SVM problems.

To solve problem (1), the main update rules of the ADMM [14] are formulated as follows:

$$y^k = \arg\min_y \quad \left\{h(y) + \frac{\beta}{2}\left\|Ax^{k-1} + By - c + \lambda^{k-1}\right\|^2\right\} \tag{2}$$

$$x^k = \arg\min_x \quad \left\{f(x) + \frac{\beta}{2}\left\|Ax + By^k - c + \lambda^{k-1}\right\|^2\right\} \tag{3}$$

$$\lambda^k = \lambda^{k-1} + Ax^k + By^k - c \tag{4}$$

where λ is the scaled dual variable as in [14], and $\beta > 0$ is a penalty parameter.

When the number of training samples (i.e., n) is very large, the per-iteration cost of standard ADMM is very high due to the computation of $\nabla f(x)$. Therefore, some efficient stochastic ADMM algorithms, such as [4] and [16], have been proposed in recent years. Their update rules for y^k and λ^k remain unchanged, while they approximate x^k as follows:

$$x^k = \arg\min_x \quad \left\{\left\langle x, \nabla f_{i_k}\left(x^{k-1}\right)\right\rangle + \frac{1}{2\eta_k}\left\|x - x^{k-1}\right\|_G^2 + \frac{\beta}{2}\left\|Ax + By^k - c + \lambda^{k-1}\right\|^2\right\} \tag{5}$$

where i_k is drawn uniformly at random from $\{1, 2, \ldots, n\}$, $\eta_k \propto 1/\sqrt{k}$ is a step-size or learning rate. Indeed, although the stochastic gradient $\nabla f_{i_k}(x^{k-1})$ is an unbiased estimator of $\nabla f(x^{k-1})$, the algorithms require the step size to asymptotically decrease to guarantee convergence, as pointed out by [12], [28], due to the variance of random sampling.

Recently, several variance reduced stochastic ADMM algorithms have been proposed, such as SAG-ADMM [21], SDCA-ADMM [22], and SVRG-ADMM [23]. SVRG-ADMM is superior to the other two methods since it does not require storing any intermediate gradients or dual variables. In particular, the SVRG estimator is formulated as follows:

$$\widetilde{\nabla} f_{I_k}\left(x^{k-1}\right) = \frac{1}{b}\sum_{i_k\in I_k}\left(\nabla f_{i_k}\left(x^{k-1}\right) - \nabla f_{i_k}(\widetilde{x})\right) + \nabla f(\widetilde{x}) \tag{6}$$

where I_k is a random minibatch set of size b uniformly drawn from $\{1, 2, \ldots, n\}$, and $\widetilde{x}$ is a snapshot point. Notably, this

SVRG estimator is unbiased [i.e., $\mathbb{E}[\widetilde{\nabla} f_{I_k}(x^{k-1})] = \nabla f(x^{k-1})$]. As a result, we can use a constant step size η to achieve faster convergence due to variance reduced stochastic gradients.

More recently, [24] proposed an efficient momentum ASVRG-ADMM method, which attains linear convergence in the SC case and achieves a convergence rate of $\mathcal{O}(1/K^2)$ in the non-SC case. Moreover, the loopless variants of SVRG and Katyusha (L-SVRG and L-Katyusha) have been developed in [25] and simplify the convergence analysis for both SVRG and Katyusha. Besides, L-SVRG is said to be the first result that a variant of SVRG achieves linear convergence with no requirement to know the condition number, which is often unknown or hard to estimate correctly in practice. The loopless methods remove the outer loop of the original methods and instead use a probabilistic update for the snapshot point to trigger the computation of the full gradient.

C. Comparison With Related Work

In this article, we introduce the loopless technique into the recently proposed SVRG-ADMM [23] and ASVRG-ADMM [24], and propose two new loopless LVR-SADMM and LAVR-SADMM methods for solving problem (1). Note that our theoretical analysis is different from those of both L-SVRG and SVRG-ADMM. Because of the equality constraint and the coupling terms in problem (1), our theoretical analysis becomes much more complex than that of L-SVRG [25]. On the other hand, our analysis is significantly simpler than that of SVRG-ADMM, since we only use a single iteration analysis to establish convergence. Moreover, we adopt a closed-form expression for x^k by substituting the specific form of the matrix G (this specific form is also adopted in [23], [24], and [29]) into the subproblem of x, on which our algorithm and convergence analysis are based. Besides the loopless design of our methods, this processing is another main factor that differs our analysis from that of SVRG-ADMM, since the latter uses a general G satisfying $G \succeq I$ throughout the proofs. Despite the simplicity of our proofs, our convergence criterion is novel and quite different from that of SVRG-ADMM.

III. Loopless Variance Reduction Stochastic ADMM

In this section, we propose a new LVR-SADMM algorithm for solving the SC and non-SC problem (1). Moreover, we also provide its convergence analysis.

A. Our LVR-SADMM Algorithm

Using the SVRG estimator in (6) and a constant step size η, the subproblem with respect to x is formulated as follows:

$$x^k = \arg\min_{x} \left\{ x^T \widetilde{\nabla} f_{I_k}\left(x^{k-1}\right) + \frac{1}{2\eta}\left\| x - x^{k-1} \right\|_G^2 + \frac{\beta}{2}\left\| Ax + By^k - c + \lambda^{k-1} \right\|^2 \right\}. \tag{7}$$

When setting $G = I$ as in [4], the closed-form solution of (7) is formulated as follows:

$$x^k = \left(\frac{1}{\eta} I + \beta A^T A\right)^{-1} \times \left(\frac{1}{\eta} x^{k-1} - \widetilde{\nabla} f_{I_k}\left(x^{k-1}\right) - \beta A^T\left(By^k - c + \lambda^{k-1}\right)\right). \tag{8}$$

The update rule of x^k in (8) involves computing the inverse of the matrix $((1/\eta)I + \beta A^T A)$, which is time consuming when $A^T A$ is large. To avoid computing the inverse of the matrix, one can set $G = \gamma I - \beta\eta A^T A$ as in the inexact Uzawa method [23], [29], where $\gamma \geq \beta\eta\|A^T A\|_2 + 1$ is a constant. Then, a new update rule for (7) is formulated as follows:

$$x^k = x^{k-1} - \frac{\eta}{\gamma}\left[\widetilde{\nabla} f_{I_k}\left(x^{k-1}\right) + \beta A^T\left(Ax^{k-1} + By^k - c + \lambda^{k-1}\right)\right]. \tag{9}$$

The details of our LVR-SADMM algorithm for solving the SC problem (1) are formally presented in Algorithm 1. Below, we make several remarks about our LVR-SADMM.

1) Our LVR-SADMM method is inspired by L-SVRG and L-Katyusha [25], [26], and our key idea is to remove the outer loop in SVRG-ADMM and instead use a small triggering probability to update the snapshot and calculate the full gradient. In essence, LVR-SADMM is a new LVR-SADMM algorithm, which uses both the loopless and variance reduction techniques. Despite this simple and intuitive thought, our analysis is significantly simpler and superior experimental results can be obtained, compared with its loopy counterpart, as shown in Section V.
2) In each iteration of LVR-SADMM, the snapshot $\widetilde{x}^k$ is updated to the previous iterate x^{k-1} with probability p.[1] Meanwhile, with the same probability, the dual variable λ^k is updated to $-(1/\beta)(A^T)^{\dagger}\nabla f(x^{k-1})$ as in SVRG-ADMM [23]. Otherwise, $\widetilde{x}^k$ remains unchanged (i.e., $\widetilde{x}^k = \widetilde{x}^{k-1}$), and $\lambda^k = \lambda^{k-1} + Ax^{k-1} + By^k - c$.
3) Note that the parameter γ in step 5 of Algorithm 1 is often set to a constant so that $\gamma \geq \gamma_{\min} \equiv \beta\eta\|A^T A\|_2 + 1$, and thus, it can guarantee that $G \succeq I$ as in [23] and [29]. In our theoretical analysis below, we can simply set $\gamma > [(2\beta\eta\|AA^T\|_2)/p]$ for our convergence criterion.

B. Convergence Analysis

In this section, we analyze the convergence property of our LVR-SADMM algorithm for SC problems. We first give our new convergence criterion, which plays an important role in our theoretical analysis. Then, our intermediate results and main theoretical results are presented.

Below, we give two additional assumptions for our convergence analysis.

[1] Note that we use x^{k-1} instead of x^k to provide simpler analysis, and this small modification barely affects the empirical behavior but is necessary for our convergence analysis. During practical implementation, it would be a better choice that x^k instead of x^{k-1} is used to update $\widetilde{x}^k$ and λ^k [i.e., $\lambda^k = -(1/\beta)(A^T)^{\dagger}\nabla f(x^k)$ instead of $\lambda^k = -(1/\beta)(A^T)^{\dagger}\nabla f(x^{k-1})$].

Algorithm 1 LVR-SADMM for the SC Problem (1)

```
Input: Penalty parameter $\beta$, learning rate $\eta$, a constant $\gamma > 0$, mini-batch size $1 \le b \le n$, and probability $p \in (0, 1]$.
Initialize: $x^0 = \widetilde{x}^0$, $y^0$, and $\lambda^0$.
for $k = 1, 2, \ldots$ do
    Choose $I_k \subseteq \{1, 2, \ldots, n\}$ of size b uniformly at random;
    $\widetilde{\nabla} f_{I_k}(x^{k-1}) = \frac{1}{|I_k|}\sum_{i_k \in I_k}\big[\nabla f_{i_k}(x^{k-1}) - \nabla f_{i_k}(\widetilde{x}^{k-1})\big] + \nabla f(\widetilde{x}^{k-1})$;
    $y^k = \arg\min_y \Big\{h(y) + \frac{\beta}{2}\|Ax^{k-1} + By - c + \lambda^{k-1}\|^2\Big\}$;
    $x^k = x^{k-1} - \frac{\eta}{\gamma}\Big[\widetilde{\nabla} f_{I_k}(x^{k-1}) + \beta A^T(Ax^{k-1} + By^k - c + \lambda^{k-1})\Big]$;
    $(\widetilde{x}^k,\ \lambda^k) = \begin{cases} (x^{k-1},\ -\frac{1}{\beta}(A^T)^{\dagger}\nabla f(x^{k-1})), & \text{with probability } p, \\ (\widetilde{x}^{k-1},\ \lambda^{k-1} + Ax^{k-1} + By^k - c), & \text{with probability } 1-p; \end{cases}$
end for
```

Assumption 3: The function f is μ-SC, if there exists a constant $\mu > 0$ such that

$$f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2 \tag{10}$$

for all $x, y \in \mathbb{R}^d$.

Assumption 4: The matrix A has full row rank.

Assumption 4 was also used in some related work, such as [23], [24], and [30]–[32]. Note that all the assumptions are the same as in SVRG-ADMM for the SC case, which indicates that there is no extra assumption for our analysis.

1) Convergence Criterion: We first introduce a new convergence criterion, and the Lyapunov function for our convergence analysis is defined as follows:

$$\Phi^k := W^k + D^k + V^k \tag{11}$$

where

$$\begin{aligned} W^k &:= \left\|x^k - x^*\right\|^2 + \frac{2\eta}{\gamma}\left[h\left(y^k\right) - h\left(y^*\right) - h'\left(y^*\right)^T\left(y^k - y^*\right)\right] \\ D^k &:= \frac{4\eta^2\delta(b)}{p\gamma^2 n}\sum_{i=1}^{n}\left\|\nabla f_i\left(\widetilde{x}^k\right) - \nabla f_i\left(x^*\right)\right\|^2 \\ V^k &:= \frac{\beta\eta\left(\gamma - 2\beta\eta\left\|AA^T\right\|_2\right)}{\gamma^2(1-p)}\left\|\lambda^k - \lambda^*\right\|^2 \\ \gamma &> \frac{2\beta\eta\left\|AA^T\right\|_2}{p}, \quad \text{and } \delta(b) = \frac{n-b}{b(n-1)}. \end{aligned}$$

Note that this convergence criterion is quite different from that used in SVRG-ADMM [23], whose criterion is

$$\begin{aligned} R(x, y) :=& f(x) - f(x^*) - \nabla f(x^*)^T(x - x^*) \\ &+ h(y) - h(y^*) - h'(y^*)^T(y - y^*). \end{aligned} \tag{12}$$

The difference is due to our loopless design and the *linearization* procedure as in (9). It is clear that Φ^k, W^k, D^k, and V^k are all nonnegative.

2) One-Iteration Analysis: Our theoretical results mainly contain two key lemmas (i.e., Lemmas 1 and 2 below) and one theorem (i.e., Theorem 1 below). All the results indicate that our LVR-SADMM converges linearly as its loopy counterpart. All the detailed proofs of the lemmas and theorem are provided in the Supplementary Material.

Lemma 1 (Variance Upper Bound):

$$\begin{aligned} &\mathbb{E}\left[\left\|\widetilde{\nabla} f_{I_k}\left(x^{k-1}\right) - \nabla f\left(x^{k-1}\right)\right\|^2\right] \\ &\le 4L\delta(b)\left[f\left(x^{k-1}\right) - f(x^*) - \left\langle\nabla f\left(x^*\right), x^{k-1} - x^*\right\rangle\right] \\ &\quad + \frac{2\delta(b)}{n}\sum_{i=1}^{n}\left\|\nabla f_i\left(\widetilde{x}^{k-1}\right) - \nabla f_i\left(x^*\right)\right\|^2 \end{aligned} \tag{13}$$

where $\delta(b) = (n - b/b(n-1)) \le 1$, $L := \max_i L_i$, and $1 \le b \le n$.

Remark 1: Since $(1/n)\sum_{i=1}^{n}\|\nabla f_i(\widetilde{x}^{k-1}) - \nabla f_i(x^*)\|^2 \le 2L[f(\widetilde{x}^{k-1}) - f(x^*) - \langle\nabla f(x^*), \widetilde{x}^{k-1} - x^*\rangle]$, it is clear that Lemma 1 presents a slightly smaller upper bound on the expected variance of the gradient estimator $\widetilde{\nabla} f_{I_k}(x^{k-1})$ than that of SVRG-ADMM [23] (see [23, Proposition 1]).

Lemma 2:

$$\begin{aligned} &\mathbb{E}\left[\left\|x^k - x^*\right\|^2 + \frac{2\eta}{\gamma}\left(\zeta^k\right)^T A\left(x^{k-1} - x^*\right)\right] \\ &\le \left(1 - \frac{\mu\eta}{\gamma}\right)\left\|x^{k-1} - x^*\right\|^2 + \frac{2\eta\gamma - 4L\eta^2(\delta(b)+1)}{\gamma^2} \\ &\quad \times \left[f\left(x^*\right) - f\left(x^{k-1}\right) + \nabla f\left(x^*\right)^T\left(x^{k-1} - x^*\right)\right] \\ &\quad + \frac{p}{2}D^{k-1} + \frac{2\eta^2\left\|AA^T\right\|_2}{\gamma^2}\left\|\zeta^k\right\|^2 \end{aligned} \tag{14}$$

where $\zeta^k := \beta(\lambda^{k-1} - \lambda^* + Ax^{k-1} + By^k - c)$, and $D^k := [(4\eta^2\delta(b))/p\gamma^2 n]\sum_{i=1}^{n}\|\nabla f_i(\widetilde{x}^k) - \nabla f_i(x^*)\|^2$.

Our main result is the following theorem, which gives the convergence rate of Algorithm 1.

Theorem 1: Suppose that Assumptions 1–4 hold. If the probability p, learning rate η, and parameter γ satisfy

$$\begin{aligned} &0 < p < \frac{(3\delta(b)+1)\sigma_{\min}\left(AA^T\right)}{(3\delta(b)+1)\sigma_{\min}\left(AA^T\right) + \left\|AA^T\right\|_2} \\ &0 < \eta < \min\left\{\frac{1}{L}, \frac{\gamma}{\mu}, \frac{\left(\beta(1-p)\sigma_{\min}\left(AA^T\right) - pL\right)\gamma}{2\beta L\left((1-p)\sigma_{\min}\left(AA^T\right)(3\delta(b)+1) - p\left\|AA^T\right\|_2\right)}\right\} \end{aligned}$$

and $\gamma > [(2\beta\eta\|AA^T\|_2)/p]$, then the following inequality holds for all $k \ge 1$:

$$\begin{aligned} &\mathbb{E}\left[W^k + D^k + V^k\right] \\ &\le \left(1 - \frac{\mu\eta}{\gamma}\right)W^{k-1} + \left(1 - \frac{p}{2}\right)D^{k-1} + \frac{1-p}{1 - \frac{2\beta\eta\left\|AA^T\right\|_2}{\gamma}}V^{k-1}. \end{aligned} \tag{15}$$

Let $\rho = \max\{1-(\mu\eta/\gamma), 1-(p/2),(1-p/1-[(2\beta\eta\|AA^T\|_2)/\gamma])\}$, and it is clear that $\rho < 1$. Then, $\mathbb{E}[\Phi^k] \leq \rho^k\Phi^0$.

Theorem 1 indicates that LVR-SADMM enjoys a fast linear convergence rate in expectation, which is identical to that of SVRG-ADMM. Although they have the same convergence rate, LVR-SADMM converges much faster in practice.

Discussion 1: According to Theorem 1, we know that the contraction of the Lyapunov function is $\max\{1-(\mu\eta/\gamma), 1-(p/2), (1-p/1-[(2\beta\eta\|AA^T\|_2)/\gamma])\}$. Since $\eta \leq 1/L$, the first term is greater than $1-(\mu/\gamma L)$, and thus, the oracle complexity is not better than $\mathcal{O}(L/\mu\log(1/\varepsilon))$. LVR-SADMM calls the stochastic gradient oracle in expectation $\mathcal{O}(b+pn)$ times in one iteration. Using these two results, one can obtain the total oracle gradient complexity $\mathcal{O}(([b/p]+n+[bL/\mu]+[Lpn/\mu])\log(1/\varepsilon))$. Therefore, if we choose $p \in [\min\{j(b/n), j(\mu/L)\}, \max\{j(b/n), j(\mu/L)\}]$, where $j=\mathcal{O}(1)$, the oracle gradient complexity of our LVR-SADMM becomes $\mathcal{O}((n+(L/\mu))\log(1/\varepsilon))$. In other words, our LVR-SADMM has the same oracle gradient complexity as SVRG-ADMM. However, LVR-SADMM converges significantly faster than SVRG-ADMM in practice, as shown in our experimental results.

Moreover, we list some remarks about our convergence results.

1) Although we simply apply the loopless idea to SVRG-ADMM, our convergence analysis is quite different from those of L-SVRG and SVRG-ADMM. Owing to the existence of the equality constraint and the coupling term in problem (1), our analysis becomes much more complicated than L-SVRG. On the other hand, our theoretical analysis is significantly simpler than SVRG-ADMM, which is consistent with the advantage of the loopless approach that a single iteration analysis is sufficient to establish convergence as in [25].
2) To avoid finding the inverse of the matrix, we substitute $G=\gamma I-\beta\eta A^TA$ into the subproblem in (7) and obtain a closed-form solution of x^k in (9), whereas SVRG-ADMM [23] uses a general matrix G as long as $G \succeq I$ throughout the whole proofs. In fact, we only require to find a constant γ satisfying $\gamma > [(2\beta\eta\|AA^T\|_2)/p]$ for our LVR-SADMM algorithm.

C. Extension to Nonstrongly Convex Settings

In this section, we extend our LVR-SADMM algorithm to the non-SC setting. The details of LVR-SADMM for solving non-SC problems are described in Algorithm 2. The main difference between Algorithms 1 and 2 is the update rule of the dual variable λ^k. That is, $\lambda^k = \lambda^{k-1}+Ax^k+By^k-c$ for Algorithm 2, while $\lambda^k = -(1/\beta)(A^T)^\dagger\nabla f(x^{k-1})$ with probability p for Algorithm 1. The mechanism behind this difference is that Algorithm 1 proposed in this article is inspired by [23, Algorithm 1], where the dual variable uses a special update rule in the outer loop, i.e., $\widetilde{\lambda}^s = -(1/\beta)(A^T)^\dagger\nabla f(\widetilde{x}^s)$, and s is the outer loop index. In our convergence analysis of Algorithm 1, we find that this update rule is also necessary to obtain the desired convergence result. Therefore, we adopt this update rule with probability p for the technical reason to show linear convergence.

Algorithm 2 LVR-SADMM for the Non-SC Problem (1)

```
Input: Penalty parameter $\beta$, learning rate $\eta$, a constant $\gamma > 0$,
    mini-batch size $1 \leq b \leq n$, and probability $p \in (0, 1]$.
Initialize: $x^0 = \widetilde{x}^0, y^0, \lambda^0$.
for $k = 1, 2, \ldots$ do
    Choose $I_k \subseteq \{1, 2, \ldots, n\}$ of size b, uniformly at random;
    $\widetilde{\nabla}f_{I_k}(x^{k-1}) = \frac{1}{|I_k|}\sum_{i_k\in I_k}\left[\nabla f_{i_k}(x^{k-1}) - \nabla f_{i_k}(\widetilde{x}^{k-1})\right] + \nabla f(\widetilde{x}^{k-1})$;
    $y^k = \arg\min_y\left\{h(y) + \frac{\beta}{2}\|Ax^{k-1}+By-c+\lambda^{k-1}\|^2\right\}$;
    $x^k = x^{k-1}-\frac{\eta}{\gamma}\left[\widetilde{\nabla}f_{I_k}(x^{k-1})+\beta A^T(Ax^{k-1}+By^k-c+\lambda^{k-1})\right]$;
    $\lambda^k = \lambda^{k-1} + Ax^k + By^k - c$;
    $\widetilde{x}^k = \begin{cases} x^{k-1}, & \text{with probability } p, \\ \widetilde{x}^{k-1}, & \text{with probability } 1-p; \end{cases}$
end for
```

IV. Accelerated Loopless Variance Reduction Stochastic ADMM

In this section, we also develop a new accelerated variant of LVR-SADMM (called LAVR-SADMM) for solving both SC and non-SC problems. Moreover, we also discuss our algorithms' applications in IoT in Section IV-C.

A. Our LAVR-SADMM in Strongly Convex Setting

In this section, we further introduce the loopless idea into ASVRG-ADMM proposed in our previous work [24], and propose a new accelerated LVR-SADMM algorithm for solving SC problems, as shown in Algorithm 3. Inspired by [24, Algorithm 1], we introduce an auxiliary variable z to construct the momentum acceleration form, i.e., $x^k = (1-\theta)\widetilde{x}^{k-1}+\theta z^k$. Similar to the subproblem in (7), after introducing the momentum weight θ $(0 \leq \theta \leq 1)$ into the proximal term $(1/2\eta)\|x-x^{k-1}\|_G^2$, a new update rule of z^k is formulated as follows:

$$z^k = \arg\min_z \left\{z^T\widetilde{\nabla}f_{I_k}\left(x^{k-1}\right) + \frac{\theta}{2\eta}\left\|z-z^{k-1}\right\|_G^2 + \frac{\beta}{2}\left\|Az+By^k-c+\lambda^{k-1}\right\|^2\right\}. \quad (16)$$

Here, we still adopt the inexact Uzawa method as in [24], i.e., $G=\gamma I_{d_1}-(\beta\eta/\theta)A^TA$ with $\gamma \geq \gamma_{\min} \equiv [(\beta\eta\|A^TA\|_2)/\theta]+1$ to guarantee that $G \succeq I$. Consequently, one can arrive at a closed-form expression of z^k as follows:

$$z^k = z^{k-1} - \frac{\eta}{\gamma\theta}\left[\widetilde{\nabla}f_{I_k}\left(x^{k-1}\right) + \beta A^T\left(Az^{k-1}+By^k-c+\lambda^{k-1}\right)\right]. \quad (17)$$

In the SC case, θ can be set to a constant as in [24], and we also take a probabilistic update for λ^k as in Algorithm 1.

Algorithm 3 LAVR-SADMM for the SC Problem (1)

```
Input: Penalty parameter $\beta$, learning rate $\eta$, a constant $\gamma > 0$, mini-batch size $1 \leq b \leq n$, and probability $p \in (0, 1]$.
Initialize: $x^0 = z^0 = \widetilde{x}^0, y^0, \theta$, and $\lambda^0$.
for $k = 1, 2, \ldots$ do
    Choose $I_k \subseteq \{1, 2, \ldots, n\}$ of size b, uniformly at random;
    $\widetilde{\nabla} f_{I_k}(x^{k-1}) = \frac{1}{|I_k|}\sum_{i_k \in I_k}\big[\nabla f_{i_k}(x^{k-1}) - \nabla f_{i_k}(\widetilde{x}^{k-1})\big] + \nabla f(\widetilde{x}^{k-1})$;
    $y^k = \arg\min_y \Big\{h(y) + \frac{\beta}{2}\|Az^{k-1} + By - c + \lambda^{k-1}\|^2\Big\}$;
    $z^k = z^{k-1} - \frac{\eta}{\gamma\theta}\Big[\widetilde{\nabla} f_{I_k}(x^{k-1}) + \beta A^T(Az^{k-1} + By^k - c + \lambda^{k-1})\Big]$;
    $x^k = (1-\theta)\widetilde{x}^{k-1} + \theta z^k$;
    $(\widetilde{x}^k,\ \lambda^k) = \begin{cases} (x^{k-1},\ -\frac{1}{\beta}(A^T)^{\dagger}\nabla f(x^{k-1})), & \text{with probability } p, \\ (\widetilde{x}^{k-1},\ \lambda^{k-1} + Az^{k-1} + By^k - c), & \text{with probability } 1-p; \end{cases}$
end for
```

Algorithm 4 LAVR-SADMM for the Non-SC Problem (1)

```
Input: Penalty parameter $\beta$, learning rate $\eta$, a constant $\gamma > 0$, mini-batch size $1 \leq b \leq n$, and probability $p \in (0, 1]$.
Initialize: $x^0 = z^0 = \widetilde{x}^0, y^0, \lambda^0$, and $\theta_0 = 1 - \frac{L\eta\delta(b)}{1-L\eta}$.
for $k = 1, 2, \ldots$ do
    Choose $I_k \subseteq \{1, 2, \ldots, n\}$ of size b, uniformly at random;
    $\widetilde{\nabla} f_{I_k}(x^{k-1}) = \frac{1}{|I_k|}\sum_{i_k \in I_k}\big[\nabla f_{i_k}(x^{k-1}) - \nabla f_{i_k}(\widetilde{x}^{k-1})\big] + \nabla f(\widetilde{x}^{k-1})$;
    $y^k = \arg\min_y \Big\{h(y) + \frac{\beta}{2}\|Az^{k-1} + By - c + \lambda^{k-1}\|^2\Big\}$;
    $z^k = z^{k-1} - \frac{\eta}{\gamma\theta_{k-1}}\Big[\widetilde{\nabla} f_{I_k}(x^{k-1}) + \beta A^T(Az^{k-1} + By^k - c + \lambda^{k-1})\Big]$;
    $x^k = (1-\theta_{k-1})\widetilde{x}^{k-1} + \theta_{k-1} z^k$;
    $\lambda^k = \lambda^{k-1} + Az^k + By^k - c$;
    $(\widetilde{x}^k,\ \theta_k) = \begin{cases} (x^{k-1},\ \frac{\sqrt{\theta_{k-1}^4 + 4\theta_{k-1}^2} - \theta_{k-1}^2}{2}), & \text{with probability } p, \\ (\widetilde{x}^{k-1},\ \theta_{k-1}), & \text{with probability } 1-p; \end{cases}$
end for
```

B. LAVR-SADMM in Nonstrongly Convex Setting

In this section, we generalize our LAVR-SADMM to the non-SC case, as shown in Algorithm 4. The momentum weight θ_k is updated as: $\theta_k = [(\sqrt{\theta_{k-1}^4 + 4\theta_{k-1}^2} - \theta_{k-1}^2)/2]$ with probability p, and remains unchanged otherwise. θ_0 is initialized to $1 - (L\eta\delta(b)/1 - L\eta)$. Note that this update rule of the momentum weight is inspired by [24]. The second main difference between Algorithms 3 and 4 is that in Algorithm 4, we do not use a probabilistic update for λ^k as in Algorithm 2. The reason for this difference is the same as that we have explained in Section III-C.

C. Our Algorithms' Applications in IoT

IoT devices with emerging products are widely used, e.g., mobile sensors, IoT-enabled drones, and vehicles. Besides, there are lots of IoT-enabled massively data-intensive applications, such as somatosensory games, hyperscale machine-type communications, holographic rendering, and multiway teleconferencing. We would say that our algorithms, including LVR-SADMM and LAVR-SADMM, can be used in many ADMM-based IoT applications. For instance, [33] used stochastic ADMM to tackle the issues with communication bottleneck and straggler node in distributed learning systems. More specifically, they formulated the decentralized optimization problem as: $\min_{x,y}\sum_{i=1}^{n} f_i(x_i; \mathcal{D}_i)$, s.t. $\mathbb{1} \otimes y - x = 0$, where $\mathcal{D}_i$ is the private data set at agent i, $\mathbb{1} = (1, \ldots, 1)^T \in \mathbb{R}^n$, and $\otimes$ is the Kronecker product. The augmented Lagrangian function of this problem is $\mathcal{L}_\beta(x, y, \lambda) = \sum_{i=1}^{n} f_i(x_i; \mathcal{D}_i) + \langle\lambda, \mathbb{1} \otimes y - x\rangle + (\beta/2)\|\mathbb{1} \otimes y - x\|^2$, where λ is the dual variable and $\beta > 0$ is a penalty parameter. Moreover, our algorithms can be applied to other ADMM-based IoT applications, such as [34] and [35]. Wu *et al.* [34] designed an ADMM algorithm and its decentralized format to implement the optimal transmission frequency management system for IoT edge intelligence. Yan *et al.* [35] proposed an ADMM-based algorithm to solve the robust beamforming design problem for downlink cloud radio access networks.

V. Experimental Results

In this section, we conduct extensive numerical experiments to demonstrate the efficiency of our LVR-SADMM and LAVR-SADMM methods for solving SC GGLR, *graph-guided SVM*, and non-SC graph-guided fused Lasso (GGFL) problems on some publicly available data sets (e.g., *bio-train*, *covtype*, *HIGGS*, and *epsilon*), as shown in Table II. All the experiments were performed in MATLAB on a PC with Intel i7-7700 3.6 GHz CPU and 32-GB RAM. As in [23] and [24], it has been verified that SVRG-ADMM and its accelerated variant ASVRG-ADMM outperform several state-of-the-art algorithms, such as STOC-ADMM [4], OPG-ADMM [17], RDA-ADMM [17], SAG-ADMM [21], SDCA-ADMM [22],

TABLE II
SUMMARY OF THE DATA SETS USED IN OUR EXPERIMENTS AND THE MINIBATCH SIZE b FOR ALL THE ALGORITHMS

Data sets	# training	# test	dimensionality	b
bio-train	72,876	72,875	74	20
covtype	290,506	290,506	54	20
HIGGS	7,700,000	3,300,000	28	150
epsilon	200,000	200,000	2,000	150

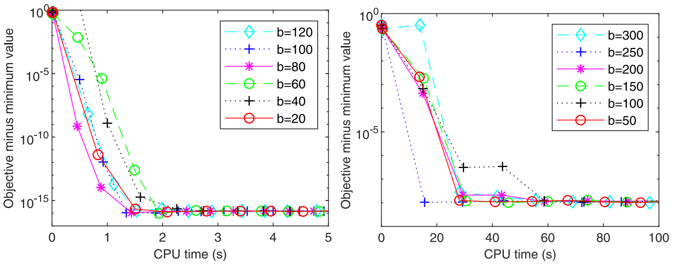


Fig. 1. Objective value minus minimum value versus CPU time (seconds) of LVR-SADMM with different minibatch sizes for GGLR problems on *covtype* (left) and *HIGGS* (right).

and SCAS-ADMM [36]. Thus, we compare our methods only with their counterparts, SVRG-ADMM and ASVRG-ADMM.

A. *Robustness Analysis of Minibatch Size*

We first test the robustness of the proposed methods on the size of minibatch b, and try different minibatch sizes in our LVR-SADMM for solving GGLR problems. Fig. 1 plots the objective value minus minimum value versus CPU time of our LVR-SADMM with different sizes of minibatch for solving the GGLR problem on *covtype* and *HIGGS*. As we can see, the practical performance of our loopless stochastic ADMM methods is quite robust with different minibatch sizes.

B. *Graph-Guided Logistic Regression*

In this section, we evaluate the performance of our LVR-SADMM and LAVR-SADMM methods for solving the following SC GGLR model:

$$\min_{x,y} \left\{ \frac{1}{n} \sum_{i=1}^{n} \Big(\log\big(1 + \exp\big(-b_i a_i^T x\big)\big) + \frac{\tau_2}{2} \|x\|^2 \Big) + \tau_1 \|y\|_1 \right\}$$
$$\text{s.t.,} \quad Ax = y$$

where $a_i \in \mathbb{R}^{d_1}$, $b_i \in \{-1, 1\}$, and $\tau_1, \tau_2 > 0$ are two regularization parameters. As in [24], we set $\tau_1 = 10^{-5}$ and $\tau_2 = 10^{-2}$ in our experiments. Similarly, we set $A = [M; I]$ as in [4], [21], [23], [24], and [37], where M is the sparsity pattern of the graph obtained by sparse inverse covariance selection [38]. For our LVR-SADMM and LAVR-SADMM, their parameters are set as follows: p is the same order of magnitude as b/n [i.e., $p = j(b/n)$, $j = \Theta(1)$] and $\gamma = 1$ as in [23] and [24]. For SVRG-ADMM and ASVRG-ADMM, we also utilize the linearized update for x^k and set the inner loop length $m = \lfloor j(n/b) \rfloor$, $j = \Theta(1)$, $\gamma = 1$ and choose the snapshot $\widetilde{x}$ and starting points in the outer loop to be the uniform average of x^k rather than the last iterate x^m as in [23] and [24]. We set the momentum weight $\theta = 0.9$ for ASVRG-ADMM and LAVR-SADMM. η and β are set according to the theoretical analysis in the methods. In addition, we set the minibatch size $b = 20$ for *bio-train* and *covtype*, $b = 150$ for *HIGGS* and *epsilon*.

Fig. 2 shows the training objective value minus the minimum value and test loss versus CPU time for the SC GGLR problem on the four data sets. We observe that both LVR-SADMM and LAVR-SADMM achieve an average speedup of 2× over SVRG-ADMM and ASVRG-ADMM, which verifies the effectiveness of our loopless technique and is consistent with our previous analysis about the mechanism behind the superior empirical behavior of our loopless methods versus their loopy counterparts. In particular, our accelerated algorithm, LAVR-SADMM, performs slightly better than other methods, including ASVRG-ADMM and LVR-SADMM. Besides, our nonaccelerated algorithm, LVR-SADMM, converges even faster than ASVRG-ADMM. Theoretically, all the four methods enjoy linear convergence in the SC case, while SVRG-ADMM and LVR-SADMM can be slightly improved by ASVRG-ADMM in terms of convergence bounds. As we mentioned in Section I, the inner loop size m for loopy algorithms is set to $\mathcal{O}(n/b)$ as a suboptimal choice rather than be proportional to the condition number of the problem (which is often unknown in reality), while loopless methods can achieve better practical performance with the probability p chosen properly from the optimal interval to be independent of the condition number. Consequently, LAVR-SADMM and LVR-SADMM converge significantly faster than ASVRG-ADMM and SVRG-ADMM in practice, respectively. We can find that the loopless technique has a greater impact on convergence speed than the momentum acceleration trick in the SC case, and thus, our LVR-SADMM converges even faster than ASVRG-ADMM.

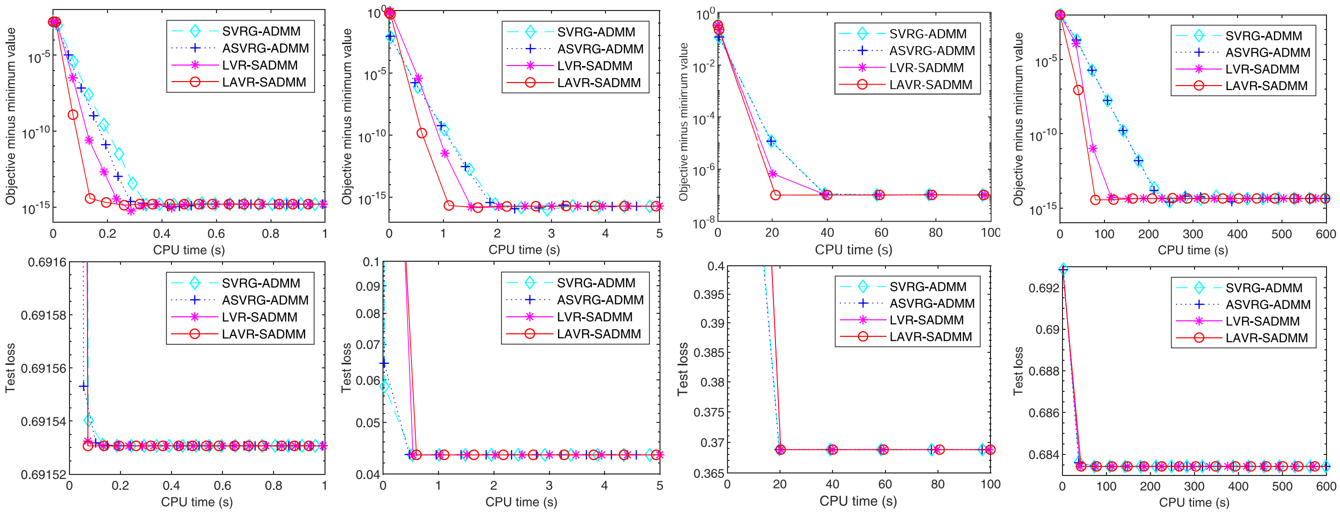
C. *Graph-Guided SVM*

In this section, we also analyze the convergence behavior of our loopless methods for solving the following SC *graph-guided SVM* model:

$$\min_{x,y} \left\{ \frac{1}{n} \sum_{i=1}^{n} \Big(\max(0, 1 - b_i a_i^T x) + \frac{\tau_2}{2} \|x\|^2 \Big) + \tau_1 \|y\|_1 \right\}$$
$$\text{s.t.,} \quad Ax = y.$$

Here, $\max(0, 1 - b_i a_i^T x)$ is the nonsmooth *hinge loss*. As in [4] and [24], we set $\tau_1 = \tau_2 = 10^{-5}$, and A is obtained in the same way as in the GGLR problem. Besides, we set $b = 80$, $\theta = 0.85$, and other parameters (i.e., m, γ, p, η, and β) are set in the same way as in GGLR. We use the publicly available data set, *20newsgroups*[2] (16 242 samples and 100 dimensions). 80% samples of the data set are used for training and 20% for test. We also adopt the one-versus-rest strategy for this multiclass classification task. Since the previous work [24] has experimentally verified that SVRG-ADMM outperforms STOC-ADMM and classical SVM, we only report the results of our algorithms and their loopy counterparts. In addition,

[2] http://www.cs.nyu.edu/ roweis/data.html



(a) (b) (c) (d)

Fig. 2. Comparison of SVRG-ADMM [23], ASVRG-ADMM [24], LVR-SADMM, and LAVR-SADMM for solving GGLR problems on the four data sets. Top: Objective value minus minimum value versus CPU time (seconds); Bottom: Test loss versus CPU time (seconds). (a) Bio-train. (b) Covtype. (c) HIGGS. (d) Epsilon.

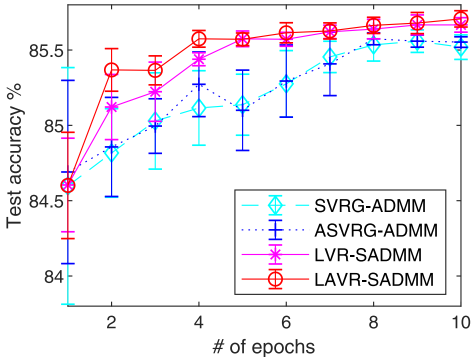


Fig. 3. Accuracies of the four stochastic variance reduction ADMM methods for multiclass classification tasks on *20newsgroups*.

we report the average accuracies and corresponding standard deviations on ten runs.

Fig. 3 shows the test accuracy versus the number of epochs of the four methods on *20newsgroups*, where the x-axis is the number of epochs, and the y-axis denotes the average prediction accuracies and standard deviations of all the methods on test data. The result indicates that LVR-SADMM and LAVR-SADMM outperform SVRG-ADMM and ASVRG-ADMM. We notice that ASVRG-ADMM is only slightly better than SVRG-ADMM, which is reasonable in the SC setting as explained in the last section.

D. Graph-Guided Fused Lasso

In this section, we further evaluate the performance of our LVR-SADMM and LAVR-SADMM methods in the non-SC case. The non-SC GGFL problem [39] is formulated as follows:

$$\min_{x,y}\left\{\frac{1}{n}\sum_{i=1}^{n}\log\left(1+\exp\left(-b_i a_i^T x\right)\right)+\tau_1\|y\|_1,\ \text{s.t.},\ \ Ax=y\right\}.$$

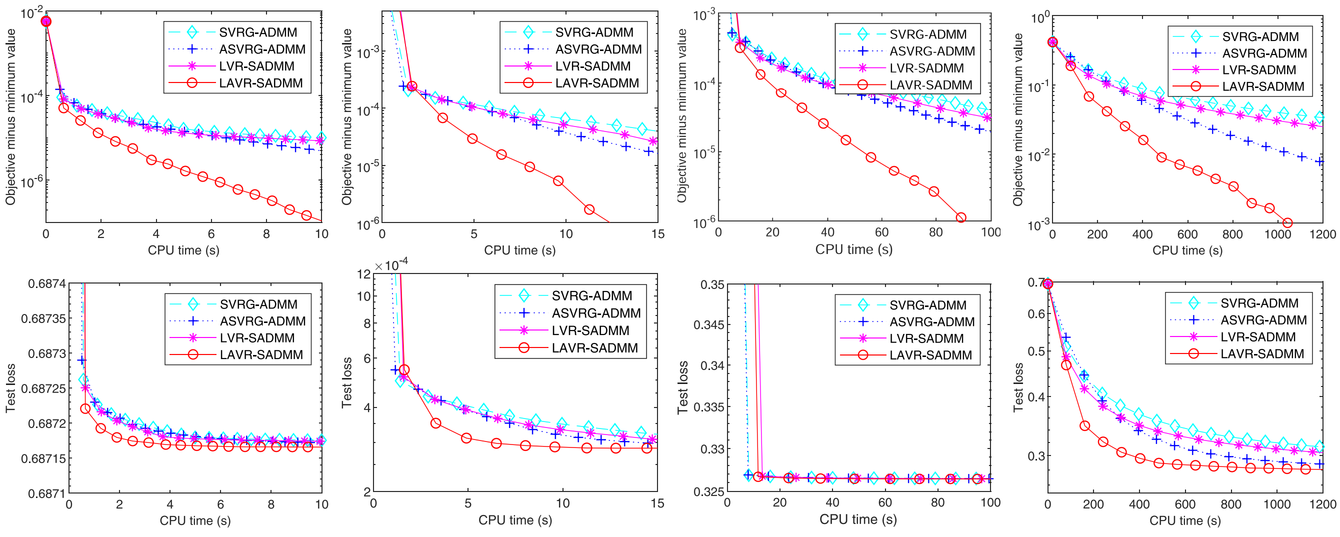
For the counterparts, SVRG-ADMM and ASVRG-ADMM, the snapshot and starting points are chosen according to their algorithms in the non-SC case.

Fig. 4 demonstrates the experimental results (including objective gap and test loss) of all the methods on the four data sets. All the results show that the accelerated algorithms, i.e., ASVRG-ADMM and LAVR-SADMM, usually outperform the nonaccelerated methods, SVRG-ADMM and LVR-SADMM, which are consistent with their convergence results, i.e., ASVRG-ADMM for $\mathcal{O}(1/K^2)$ versus SVRG-ADMM for $\mathcal{O}(1/K)$. In particular, LAVR-SADMM can achieve an average speedup of 5× over other algorithms, which verifies the effectiveness of LAVR-SADMM.

VI. Conclusion and Further Work

In this article, we first proposed an efficient loopless LVR-SADMM method for both SC and non-SC equality constrained optimization problems in various IoT applications. We also theoretically analyzed the convergence property of LVR-SADMM, which shows that it attains the best-known linear convergence as its loopy counterpart for SC problems. Our LVR-SADMM nontrivially extends to the non-SC setting. Moreover, we also proposed an accelerated LAVR-SADMM method. We also discussed our algorithms' applications in IoT. Various experimental results verified that the proposed methods converge much faster than their loopy counterparts.

Moreover, there is still some work that needs to be done for the proposed algorithms, such as a further improvement in theoretical results for non-SC problems, convergence analysis for our accelerated algorithm, and self-adaptive probability options in practice. An interesting direction of future work is the convergence research of the proposed accelerated algorithm, as well as the loopless variants of accelerated algorithms [40]–[42]. Furthermore, it is also interesting to extend



(a) (b) (c) (d)

Fig. 4. Comparison of all the four methods for GGFL on the four data sets. Top: Objective value minus minimum value versus CPU time (seconds); Bottom: Test loss versus CPU time (seconds). (a) Bio-train. (b) Covtype. (c) HIGGS. (d) Epsilon.

our algorithms and theoretical results from the two-block case as in (1) to the multiblock case [43].

ACKNOWLEDGMENT

The authors thank all the reviewers for their valuable comments.

REFERENCES

[1] D. Silver *et al.*, “Mastering the game of go without human knowledge,” *Nature*, vol. 550, pp. 354–359, Oct. 2017.

[2] Y. Liu, F. Shang, H. Liu, L. Kong, L. Jiao, and Z. Lin, “Accelerated variance reduction stochastic ADMM for large-scale machine learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 8, 2018, doi: 10.1109/TPAMI.2020.3000512.

[3] L. W. Zhong and J. T. Kwok, “Accelerated stochastic gradient method for composite regularization,” in *Proc. 17th Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2014, pp. 1086–1094.

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

[5] X. Li, R. Arora, H. Liu, J. Haupt, and T. Zhao, “Nonconvex sparse learning via stochastic optimization with progressive variance reduction,” May 2016. [Online]. Available: arXiv:1605.02711.

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

[7] W. Xie, X. Jia, L. Shen, and M. Yang, “Sparse deep feature learning for facial expression recognition,” *Pattern Recognit.*, vol. 96, Dec. 2019, Art. no. 106966.

[8] B. Recht and C. Ré, “Parallel stochastic gradient algorithms for large-scale matrix completion,” *Math. Program. Comput.*, vol. 5, no. 2, pp. 201–226, Jun. 2013.

[9] L. Wang, X. Zhang, and Q. Gu, “A unified variance reduction-based framework for nonconvex low-rank matrix recovery,” in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3712–3721.

[10] J. Fan and T. W. Chow, “Matrix completion by least-square, low-rank, and sparse self-representations,” *Pattern Recognit.*, vol. 71, pp. 290–305, Nov. 2017.

[11] Y. Liu, L. Jiao, and F. Shang, “A fast tri-factorization method for low-rank matrix recovery and completion,” *Pattern Recognit.*, vol. 46, no. 1, pp. 163–173, 2013.

[12] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 315–323.

[13] Z. Allen-Zhu and E. Hazan, “Variance reduction for faster non-convex optimization,” in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 699–707.

[14] 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.

[15] H. Robbins and S. Monro, “A stochastic approximation method,” *Ann. Math. Stat.*, vol. 22, no. 3, pp. 400–407, 1951.

[16] H. Wang and A. Banerjee, “Online alternating direction method,” in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, 2012, pp. 1119–1126.

[17] T. Suzuki, “Dual averaging and proximal gradient descent for online alternating direction multiplier method,” in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 392–400.

[18] N. L. Roux, M. Schmidt, and F. Bach, “A stochastic gradient method with an exponential convergence rate for finite training sets,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 2672–2680.

[19] S. Shalev-Shwartz and T. Zhang, “Stochastic dual coordinate ascent methods for regularized loss minimization,” *J. Mach. Learn. Res.*, vol. 14, pp. 567–599, Sep. 2013.

[20] 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.

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

[22] T. Suzuki, “Stochastic dual coordinate ascent with alternating direction method of multipliers,” in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 736–744.

[23] S. Zheng and J. T. Kwok, “Fast-and-light stochastic ADMM,” in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2407–2413.

[24] Y. Liu, F. Shang, and J. Cheng, “Accelerated variance reduced stochastic ADMM,” in *Proc. 31st AAAI Conf. Artif. Intell. (AAAI)*, 2017, pp. 2287–2293.

[25] D. Kovalev, S. Horvath, and P. Richtárik, “Don’t jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop,” in *Proc. 31st Int. Conf. Algorithmic Learn. Theory*, 2020, pp. 451–467.

[26] X. Qian, Z. Qu, and P. Richtárik, “L-SVRG and L-Katyusha with arbitrary sampling,” 2019. [Online]. Available: arXiv:1906.01481.

[27] Z. Allen-Zhu, “Katyusha: The first direct acceleration of stochastic gradient methods,” *J. Mach. Learn. Res.*, vol. 18, no. 221, pp. 1–51, 2018.

[28] F. Shang, H. Huang, J. Fan, Y. Liu, H. Liu, and J. Liu, “Asynchronous parallel, sparse approximated SVRG for high-dimensional machine learning,” *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 2, 2021, doi: 10.1109/TKDE.2021.3070539.

[29] 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.

[30] F. Shang, Y. Liu, J. Cheng, and H. Cheng, "Robust principal component analysis with missing data," in *Proc. 23rd Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2014, pp. 1149–1158.

[31] 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. (ICML)*, 2015, pp. 343–352.

[32] 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.

[33] H. Chen, Y. Ye, M. Xiao, M. Skoglund, and H. V. Poor, "Coded stochastic admm for decentralized consensus optimization with edge computing," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5360–5373, Apr. 2021.

[34] H. Wu, N. E. O'Connor, J. Bruton, and M. Liu, "An ADMM-based optimal transmission frequency management system for IoT edge intelligence," 2021. [Online]. Available: arXiv:2104.07614.

[35] D. Yan, R. Wang, E. Liu, and Q. Hou, "ADMM-based robust beamforming design for downlink cloud radio access networks," *IEEE Access*, vol. 6, pp. 27912–27922, 2018.

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

[37] S. Azadi and S. Sra, "Towards an optimal stochastic alternating direction method of multipliers," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 620–628.

[38] 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, Mar. 2008.

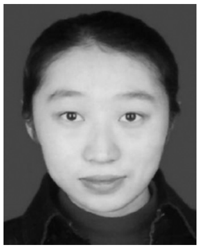
[39] S. Kim, K.-A. Sohn, and E. P. Xing, "A multivariate regression approach to association analysis of a quantitative trait network," *Bioinformatics*, vol. 25, no. 12, pp. i204–i212, May 2009.

[40] K. Zhou, F. Shang, and J. Cheng, "A simple stochastic variance reduced algorithm with fast convergence rates," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 5975–5984.

[41] 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. Stat. (AISTATS)*, 2018, pp. 1027–1036.

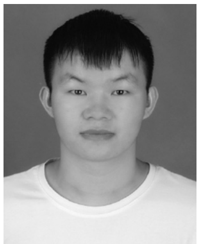
[42] Y. Liu, F. Shang, and L. Jiao, "Accelerated incremental gradient descent using momentum acceleration with scaling factor," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 3045–3051.

[43] 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, Oct. 2016.



Yuanyuan Liu (Member, IEEE) received the Ph.D. 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. Prior to joining Xidian University, she was a Postdoctoral Research Fellow with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, where she was a Postdoctoral Research Fellow with the Department of Systems Engineering and Engineering Management, from 2013 to 2014. Her current research interests include machine learning, pattern recognition, and image processing.



Jiacheng Geng received the B.S. degree in electronic information engineering from Hubei University, Wuhan, China, in 2016. He is currently pursuing the master's degree with the School of Artificial intelligence, Xidian University, Xi'an, China.

His current research interests include large-scale machine learning and stochastic optimization.

Fanhua Shang (Senior Member, IEEE) received the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 2012.

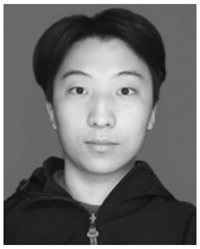
He is currently a Professor with the School of Artificial Intelligence, Xidian University. Prior to joining Xidian University, he was a Research Associate with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, where he was a Postdoctoral Research Fellow with the Department of Computer Science and Engineering, from 2013 to 2015. From 2012 to 2013, he was a Postdoctoral Research Associate with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. His current research interests include machine learning, data mining, pattern recognition, and computer vision.

Weixin An received the B.S. degree in information and computation science from Xi'an University of Technology, Xi'an, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Artificial intelligence, Xidian University, Xi'an.

His current research interests include large-scale machine learning and stochastic optimization.

Hongying Liu (Member, IEEE) received the B.E. and M.S. degrees in computer science and technology from Xi'an University of Technology, Xi'an, China, in 2006 and 2009, respectively, and the Ph.D. degree in engineering from Waseda University, Tokyo, Japan, in 2012.

She is currently 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, Xi'an. Her major research interests include image processing, intelligent signal processing, and machine learning.



Qi Zhu is currently pursuing the B.S. degree in telecommunication engineering from the School of Telecommunication Engineering, Xidian University, Xi'an, China.

His current research interests include stochastic optimization for machine learning, sparse signal recovery, and image processing.