Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Accelerated Variance Reduced Stochastic ADMM

Yuanyuan Liu, Fanhua Shang,* James Cheng
Department of Computer Science and Engineering, The Chinese University of Hong Kong
{yyliu, fhshang, jcheng}@cse.cuhk.edu.hk

Abstract

Recently, many variance reduced stochastic alternating direction method of multipliers (ADMM) methods (e.g. SAG-ADMM, SDCA-ADMM and SVRG-ADMM) have made exciting progress such as linear convergence rates for strongly convex problems. However, the best known convergence rate for general convex problems is $\mathcal{O}(1/T)$ as opposed to $\mathcal{O}(1/T^2)$ of accelerated batch algorithms, where T is the number of iterations. Thus, there still remains a gap in convergence rates between existing stochastic ADMM and batch algorithms. To bridge this gap, we introduce the momentum acceleration trick for batch optimization into the stochastic variance reduced gradient based ADMM (SVRG-ADMM), which leads to an accelerated (ASVRG-ADMM) method. Then we design two different momentum term update rules for strongly convex and general convex cases. We prove that ASVRG-ADMM converges linearly for strongly convex problems. Besides having a low per-iteration complexity as existing stochastic ADMM methods, ASVRG-ADMM improves the convergence rate on general convex problems from $\mathcal{O}(1/T)$ to $\mathcal{O}(1/T^2)$. Our experimental results show the effectiveness of ASVRG-ADMM.

Introduction

In this paper, we consider a class of composite convex optimization problems

$$\min_{x\in\mathbb{R}^{d_1}} f(x) + h(Ax), \tag{1}$$

where $A\in\mathbb{R}^{d_2\times d_1}$ is a given matrix, $f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, each $f_i(x)$ is a convex function, and $h(Ax)$ is convex but possibly non-smooth. With regard to $h(\cdot)$, we are interested in a sparsity-inducing regularizer, e.g. ℓ_1-norm, group Lasso and nuclear norm. When A is an identity matrix, i.e. $A = I_{d_1}$, the above formulation (1) arises in many places in machine learning, statistics, and operations research (Bubeck 2015), such as logistic regression, Lasso and support vector machine (SVM). We mainly focus on the large sample regime. In this regime, even first-order batch methods, e.g. FISTA (Beck and Teboulle 2009), become computationally burdensome due to their per-iteration complexity of $O(nd_1)$. As a result, stochastic gradient descent (SGD) with per-iteration complexity of $O(d_1)$ has witnessed tremendous progress in the recent years. Especially, a number of stochastic variance reduced gradient methods such as SAG (Roux, Schmidt, and Bach 2012), SDCA (Shalev-Shwartz and Zhang 2013) and SVRG (Johnson and Zhang 2013) have been proposed to successfully address the problem of high variance of the gradient estimate in ordinary SGD, resulting in a linear convergence rate (for strongly convex problems) as opposed to sub-linear rates of SGD. More recently, the Nesterov's acceleration technique (Nesterov 2004) was introduced in (Allen-Zhu 2016; Hien et al. 2016) to further speed up the stochastic variance-reduced algorithms, which results in the best known convergence rates for both strongly convex and general convex problems. This motivates us to integrate the momentum acceleration trick into the stochastic alternating direction method of multipliers (ADMM) below.

When A is a more general matrix, i.e. $A\neq I_{d_1}$, the formulation (1) becomes many more complicated problems arising from machine learning, e.g. graph-guided fuzed Lasso (Kim, Sohn, and Xing 2009) and generalized Lasso (Tibshirani and Taylor 2011). To solve this class of composite optimization problems with an auxiliary variable $y = Ax$, which are the special case of the general ADMM form,

$$\min_{x\in\mathbb{R}^{d_1}, y\in\mathbb{R}^{d_2}} f(x) + h(y), \text{ s.t. } Ax + By = c, \tag{2}$$

the ADMM is an effective optimization tool (Boyd et al. 2011), and has shown attractive performance in a wide range of real-world problems, such as big data classification (Nie et al. 2014). To tackle the issue of high per-iteration complexity of batch (deterministic) ADMM (as a popular first-order optimization method), Wang and Banerjee (2012), Suzuki (2013) and Ouyang et al. (2013) proposed some online or stochastic ADMM algorithms. However, all these variants only achieve the convergence rate of $\mathcal{O}(1/\sqrt{T})$ for general convex problems and $\mathcal{O}(\log T/T)$ for strongly convex problems, respectively, as compared with the $\mathcal{O}(1/T^2)$ and linear convergence rates of accelerated batch algorithms (Nesterov 1983), e.g. FISTA, where T is the number of iterations. By now several accelerated and faster converging versions of stochastic ADMM, which are all based on variance reduction techniques, have been proposed, e.g. SAG-ADMM (Zhong and Kwok 2014b),

*Corresponding author.

Table 1: Comparison of convergence rates and memory requirements of some stochastic ADMM algorithms.

	General convex	Strongly-convex	Space requirement
SAG-ADMM	$\mathcal{O}(1/T)$	unknown	$O(d_1d_2+nd_1)$
SDCA-ADMM	unknown	linear rate	$O(d_1d_2+n)$
SCAS-ADMM	$\mathcal{O}(1/T)$	$\mathcal{O}(1/T)$	$O(d_1d_2)$
SVRG-ADMM	$\mathcal{O}(1/T)$	linear rate	$O(d_1d_2)$
ASVRG-ADMM	$\mathcal{O}(1/T^2)$	linear rate	$O(d_1d_2)$

SDCA-ADMM (Suzuki 2014) and SVRG-ADMM (Zheng and Kwok 2016). With regard to strongly convex problems, Suzuki (2014) and Zheng and Kwok (2016) proved that linear convergence can be obtained for the special ADMM form (i.e. $B = -I_{d_2}$ and $c = \mathbf{0}$) and the general ADMM form, respectively. In SAG-ADMM and SVRG-ADMM, an $\mathcal{O}(1/T)$ convergence rate can be guaranteed for general convex problems, which implies that there still remains a gap in convergence rates between the stochastic ADMM and accelerated batch algorithms.

To bridge this gap, we integrate the momentum acceleration trick in (Tseng 2010) for deterministic optimization into the stochastic variance reduction gradient (SVRG) based stochastic ADMM (SVRG-ADMM). Naturally, the proposed method has low per-iteration time complexity as existing stochastic ADMM algorithms, and does not require the storage of all gradients (or dual variables) as in SCAS-ADMM (Zhao, Li, and Zhou 2015) and SVRG-ADMM (Zheng and Kwok 2016), as shown in Table 1. We summarize our main contributions below.

- We propose an accelerated variance reduced stochastic ADMM (ASVRG-ADMM) method, which integrates both the momentum acceleration trick in (Tseng 2010) for batch optimization and the variance reduction technique of SVRG (Johnson and Zhang 2013).
- We prove that ASVRG-ADMM achieves a linear convergence rate for strongly convex problems, which is consistent with the best known result in SDCA-ADMM (Suzuki 2014) and SVRG-ADMM (Zheng and Kwok 2016).
- We also prove that ASVRG-ADMM has a convergence rate of $\mathcal{O}(1/T^2)$ for non-strongly convex problems, which is a factor of T faster than SAG-ADMM and SVRG-ADMM, whose convergence rates are $\mathcal{O}(1/T)$.
- Our experimental results further verified that our ASVRG-ADMM method has much better performance than the state-of-the-art stochastic ADMM methods.

Related Work

Introducing $y = Ax \in \mathbb{R}^{d_2}$, problem (1) becomes

$$\min_{x\in\mathbb{R}^{d_1}, y\in\mathbb{R}^{d_2}} f(x) + h(y), \text{ s.t. } Ax - y = \mathbf{0}. \tag{3}$$

Although (3) is only a special case of the general ADMM form (2), when $B = -I_{d_2}$ and $c = \mathbf{0}$, the stochastic (or online) ADMM algorithms and theoretical results in (Wang and Banerjee 2012; Ouyang et al. 2013; Zhong and Kwok 2014b; Zheng and Kwok 2016) and this paper are all for the more general problem (2). To minimize (2), together with the dual variable λ, the update steps of batch ADMM are

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

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

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

where $\beta > 0$ is a penalty parameter.

To extend the batch ADMM to the online and stochastic settings, the update steps for y_k and λ_k remain unchanged. In (Wang and Banerjee 2012; Ouyang et al. 2013), the update step of x_k is approximated as follows:

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

where we draw i_k uniformly at random from $[n] := \{1, \ldots, n\}$, $\eta_k \propto 1/\sqrt{k}$ is the step-size, and $\|z\|_G^2 = z^TGz$ with given positive semi-definite matrix G, e.g. $G = I_{d_1}$ in (Ouyang et al. 2013). Analogous to SGD, the stochastic ADMM variants use an unbiased estimate of the gradient at each iteration. However, all those algorithms have much slower convergence rates than their batch counterpart, as mentioned above. This barrier is mainly due to the variance introduced by the stochasticity of the gradients. Besides, to guarantee convergence, they employ a decaying sequence of step sizes η_k, which in turn impacts the rates.

More recently, a number of variance reduced stochastic ADMM methods (e.g. SAG-ADMM, SDCA-ADMM and SVRG-ADMM) have been proposed and made exciting progress such as linear convergence rates. SVRG-ADMM in (Zheng and Kwok 2016) is particularly attractive here because of its low storage requirement compared with the algorithms in (Zhong and Kwok 2014b; Suzuki 2014). Within each epoch of 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

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

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

Accelerated Variance Reduced Stochastic ADMM

In this section, we design an accelerated variance reduced stochastic ADMM method for both strongly convex and general convex problems. We first make the following assumptions: Each convex $f_i(\cdot)$ is L_i-smooth, i.e. there exists a constant $L_i > 0$ such that $\|\nabla f_i(x) - \nabla f_i(y)\| \le L_i\|x-y\|$, $\forall x, y \in \mathbb{R}^d$, and $L \triangleq \max_i L_i$; $f(\cdot)$ is μ-strongly convex, i.e. there is $\mu > 0$ such that $f(x) \ge f(y) + \nabla f(y)^T(x-y) + \frac{\mu}{2}\|x-y\|^2$ for all $x, y \in \mathbb{R}^d$; The matrix A has full row rank. The first two assumptions are common in the analysis of first-order

Algorithm 1 ASVRG-ADMM for strongly-convex case

Input: m, η, $\beta > 0$, $1 \leq b \leq n$.

Initialize: $\widetilde{x}^0 = \widetilde{z}^0$, $\widetilde{y}^0$, θ, $\widetilde{\lambda}^0 = -\frac{1}{\beta}(A^T)^\dagger \nabla f(\widetilde{x}_0)$.

for $s = 1, 2, \ldots, T$ **do**
  $x_0^s = z_0^s = \widetilde{x}^{s-1}$, $y_0^s = \widetilde{y}^{s-1}$, $\lambda_0^s = \widetilde{\lambda}^{s-1}$;
  $\widetilde{p} = \nabla f(\widetilde{x}^{s-1})$;
  for $k = 1, 2, \ldots, m$ **do**
    Choose $I_k \subseteq [n]$ of size b, uniformly at random;
    $y_k^s = \arg\min_y h(y) + \frac{\beta}{2}\|Az_{k-1}^s + By - c + \lambda_{k-1}^s\|^2$;
    $z_k^s = z_{k-1}^s - \frac{\eta\left(\widetilde{\nabla} f_{I_k}(x_{k-1}^s) + \beta A^T(Az_{k-1}^s + By_k^s - c + \lambda_{k-1}^s)\right)}{\gamma\theta}$;
    $x_k^s = (1-\theta)\widetilde{x}^{s-1} + \theta z_k^s$;
    $\lambda_k^s = \lambda_{k-1}^s + Az_k^s + By_k^s - c$;
  end for
  $\widetilde{x}^s = \frac{1}{m}\sum_{k=1}^m x_k^s$, $\widetilde{y}^s = (1-\theta)\widetilde{y}^{s-1} + \frac{\theta}{m}\sum_{k=1}^m y_k^s$,
  $\widetilde{\lambda}^s = -\frac{1}{\beta}(A^T)^\dagger \nabla f(\widetilde{x}^s)$;
end for

Output: $\widetilde{x}^T$, $\widetilde{y}^T$.

optimization methods, while the last one has been used in the convergence analysis of batch ADMM (Shang et al. 2014; Nishihara et al. 2015; Deng and Yin 2016) and stochastic ADMM (Zheng and Kwok 2016).

The Strongly Convex Case

In this part, we consider the case of (2) 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 SVM as notable examples. To efficiently solve this class of problems, we incorporate both the momentum acceleration and variance reduction techniques into stochastic ADMM. Our algorithm is divided into T epochs, and each epoch consists of m stochastic updates, where m is usually chosen to be $O(n)$ as in (Johnson and Zhang 2013).

Let z be an important auxiliary variable, its update rule is given as follows. Similar to (Zhong and Kwok 2014b; Zheng and Kwok 2016), we also use the inexact Uzawa method (Zhang, Burger, and Osher 2011) to approximate the sub-problem (7), which can avoid computing the inverse of the matrix $(\frac{1}{\eta}I_{d_1} + \beta A^T A)$. Moreover, the momentum weight $0 \leq \theta_s \leq 1$ (the update rule for θ_s is provided below) is introduced into the proximal term $\frac{1}{2\eta}\|x - x_{k-1}\|_G^2$ similar to that of (7), and then the sub-problem with respect to z is formulated as follows:

$$\begin{aligned}\min_z\ & (z - z_{k-1}^s)^T \widetilde{\nabla} f_{I_k}(x_{k-1}^s) + \frac{\theta_{s-1}}{2\eta}\|z - z_{k-1}^s\|_G^2 \\ & + \frac{\beta}{2}\|Az + By_k^s - c + \lambda_{k-1}^s\|^2,\end{aligned} \tag{9}$$

where $\widetilde{\nabla} f_{I_k}(x_{k-1}^s)$ is defined in (8), $\eta < \frac{1}{2L}$, and $G = \gamma I_{d_1} - \frac{\eta\beta}{\theta_{s-1}}A^T A$ with $\gamma \geq \gamma_{\min} \equiv \frac{\eta\beta\|A^T A\|_2}{\theta_{s-1}} + 1$ to ensure that $G \succeq I$ similar to (Zheng and Kwok 2016), where $\|\cdot\|_2$ is the spectral norm, i.e. the largest singular value of the matrix. Furthermore, the update rule for x is given by

$$x_k^s = \widetilde{x}^{s-1} + \theta_{s-1}(z_k^s - \widetilde{x}^{s-1}) = (1-\theta_{s-1})\widetilde{x}^{s-1} + \theta_{s-1}z_k^s, \tag{10}$$

where $\theta_{s-1}(z_k^s - \widetilde{x}^{s-1})$ is the key momentum term (similar to those in accelerated batch methods (Nesterov 2004)), which helps accelerate our algorithm by using the iterate of the previous epoch, i.e. $\widetilde{x}^{s-1}$. Similar to x_k^s, $\widetilde{y}^s = (1 - \theta_{s-1})\widetilde{y}^{s-1} + \frac{\theta_{s-1}}{m}\sum_{k=1}^m y_k^s$. Moreover, θ_s can be set to a constant θ in all epochs of our algorithm, which must satisfy $0 \leq \theta \leq 1 - \delta(b)/(\alpha - 1)$, where $\alpha = \frac{1}{L\eta} > 1 + \delta(b)$, and $\delta(b)$ is defined below. The optimal value of θ is provided in Proposition 1 below. The detailed procedure is shown in Algorithm 1, where we adopt the same initialization technique for $\widetilde{\lambda}^s$ as in (Zheng and Kwok 2016), and $(\cdot)^\dagger$ is the pseudo-inverse. Note that, when $\theta = 1$, ASVRG-ADMM degenerates to SVRG-ADMM in (Zheng and Kwok 2016).

The Non-Strongly Convex Case

In this part, we consider general convex problems of the form (2) when each $f_i(\cdot)$ is convex, L-smooth, and $h(\cdot)$ is not necessarily strongly convex (but possibly non-smooth). Different from the strongly convex case, the momentum weight θ_s is required to satisfy the following inequalities:

$$\frac{1 - \theta_s}{\theta_s^2} \leq \frac{1}{\theta_{s-1}^2} \text{ and } 0 \leq \theta_s \leq 1 - \frac{\delta(b)}{\alpha - 1}, \tag{11}$$

where $\delta(b) := \frac{n-b}{b(n-1)}$ is a decreasing function with respect to the mini-batch size b. The condition (11) allows the momentum weight to decease, but not too fast, similar to the requirement on the step-size η_k in classical SGD and stochastic ADMM (Tseng 1998). Unlike batch acceleration methods, the weight must satisfy both inequalities in (11).

Motivated by the momentum acceleration techniques in (Tseng 2010; Nesterov 2004) for batch optimization, we give the update rule of the weight θ_s for the mini-batch case:

$$\theta_s = \frac{\sqrt{\theta_{s-1}^4 + 4\theta_{s-1}^2} - \theta_{s-1}^2}{2} \text{ and } \theta_0 = 1 - \frac{\delta(b)}{\alpha - 1}. \tag{12}$$

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. batch version), $\delta(n) = 0$ and $\theta_0 = 1$. Since $\{\theta_s\}$ is decreasing, then $\theta_s \leq 1 - \frac{\delta(b)}{\alpha - 1}$ is satisfied. The detailed procedure is shown in Algorithm 2, which has many 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 weight θ_s. That is, θ_s in Algorithm 1 can be set to a constant, while that in Algorithm 2 is adaptively adjusted as in (12).

Convergence Analysis

This section provides the convergence analysis of our ASVRG-ADMM algorithms (i.e. Algorithms 1 and 2) for strongly convex and general convex problems, respectively. Following (Zheng and Kwok 2016), we first introduce the following function $P(x, y) := f(x) - f(x^*) - \nabla f(x^*)^T(x - x^*) + h(y) - h(y^*) - h'(y^*)^T(y - y^*)$ as a convergence criterion, where $h'(y)$ denotes the (sub)gradient of $h(\cdot)$ at y. Indeed, $P(x, y) \geq 0$ for all $x, y \in \mathbb{R}^d$. In the following, we give the intermediate key results for our analysis.

Algorithm 2 ASVRG-ADMM for general convex case

Input: m, η, $\beta > 0$, $1 \le b \le n$.

Initialize: $\widetilde{x}^0 = \widetilde{z}^0, \widetilde{y}^0, \widetilde{\lambda}^0, \theta_0 = 1 - \frac{L\eta\delta(b)}{1-L\eta}$.

for $s = 1, 2, \ldots, T$ **do**
  $x_0^s = (1-\theta_{s-1})\widetilde{x}^{s-1} + \theta_{s-1}\widetilde{z}^{s-1}$, $y_0^s = \widetilde{y}^{s-1}$, $\lambda_0^s = \widetilde{\lambda}^{s-1}$;
  $\widetilde{p} = \nabla f(\widetilde{x}^{s-1})$, $z_0^s = \widetilde{z}^{s-1}$;
  for $k = 1, 2, \ldots, m$ **do**
    Choose $I_k \subseteq [n]$ of size b, uniformly at random;
    $y_k^s = \arg\min_y h(y) + \frac{\beta}{2}\|Az_{k-1}^s + By - c + \lambda_{k-1}^s\|^2$;
    $z_k^s = z_{k-1}^s - \frac{\eta\left(\widetilde{\nabla} f_{I_k}(x_{k-1}^s) + \beta A^T(Az_{k-1}^s + By_k^s - c + \lambda_{k-1}^s)\right)}{\gamma\theta_{s-1}}$;
    $x_k^s = (1-\theta_{s-1})\widetilde{x}^{s-1} + \theta_{s-1}z_k^s$;
    $\lambda_k^s = \lambda_{k-1}^s + Az_k^s + By_k^s - c$;
  end for
  $\widetilde{x}^s = \frac{1}{m}\sum_{k=1}^m x_k^s$, $\widetilde{y}^s = (1-\theta_{s-1})\widetilde{y}^{s-1} + \frac{\theta_{s-1}}{m}\sum_{k=1}^m y_k^s$,
  $\widetilde{\lambda}^s = \lambda_m^s$, $\widetilde{z}^s = z_m^s$, $\theta_s = \frac{\sqrt{\theta_{s-1}^4 + 4\theta_{s-1}^2} - \theta_{s-1}^2}{2}$;
end for

Output: $\widetilde{x}^T$, $\widetilde{y}^T$.

Lemma 1.

$$\mathbb{E}[\|\widetilde{\nabla} f_{I_k}(x_{k-1}^s) - \nabla f(x_{k-1}^s)\|^2]$$
$$\le 2L\delta(b)\left[f(\widetilde{x}^{s-1}) - f(x_{k-1}^s) + (x_{k-1}^s - \widetilde{x}^{s-1})^T\nabla f(x_{k-1}^s)\right],$$

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

Lemma 2. *Using the same notation as in Lemma 1, let* (x^*, y^*, λ^*) *denote an optimal solution of problem (2), and* $\{(z_k^s, x_k^s, y_k^s, \lambda_k^s, \widetilde{x}^s, \widetilde{y}^s)\}$ *be the sequence generated by Algorithm 1 or 2 with* $\theta_s \le 1 - \frac{\delta(b)}{\alpha-1}$, *where* $\alpha = \frac{1}{L\eta}$. *Then the following holds for all k,*

$$\mathbb{E}\left[P(\widetilde{x}^s, \widetilde{y}^s) - \frac{\theta_{s-1}}{m}\sum_{k=1}^m\left((x^* - z_k^s)^T A^T\varphi_k^s + (y^* - y_k^s)^T B^T\varphi_k^s\right)\right]$$
$$\le \mathbb{E}\left[\frac{P(\widetilde{x}^{s-1}, \widetilde{y}^{s-1})}{1/(1-\theta_{s-1})} + \frac{\theta_{s-1}^2\left(\|x^* - z_0^s\|_G^2 - \|x^* - z_m^s\|_G^2\right)}{2m\eta}\right]$$
$$+ \frac{\beta\theta_{s-1}}{2m}\mathbb{E}\left[\|Az_0^s - Ax^*\|^2 - \|Az_m^s - Ax^*\|^2 + \sum_{k=1}^m\|\lambda_k^s - \lambda_{k-1}^s\|^2\right]$$

where $\varphi_k^s = \beta(\lambda_k^s - \lambda^*)$.

The detailed proofs of Lemmas 1 and 2 are provided in the Supplementary Material.

Linear Convergence

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

Theorem 1. *Using the same notation as in Lemma 2 with given* $\theta \le 1 - \frac{\delta(b)}{\alpha-1}$, *and suppose* $f(\cdot)$ *is* μ*-strongly convex and* L_f*-smooth, and* m *is sufficiently large so that*

$$\rho = \underbrace{\frac{\theta\|\theta G + \eta\beta A^TA\|_2}{\eta m\mu}}_{1} + \underbrace{1-\theta}_{2} + \underbrace{\frac{L_f\theta}{\beta m\sigma_{\min}(AA^T)}}_{3} < 1, \quad (13)$$

where $\sigma_{\min}(AA^T)$ *is the smallest eigenvalue of the positive semi-definite matrix* AA^T, *and* G *is defined in (9). Then*

$$\mathbb{E}\left[P(\widetilde{x}^T, \widetilde{y}^T)\right] \le \rho^T P(\widetilde{x}^0, \widetilde{y}^0).$$

The proof of Theorem 1 is provided in the Supplementary Material. From Theorem 1, one 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 *in (Zheng and Kwok 2016) with the same* η *and* β. *Specifically, as shown in (13),* ρ *consists of three components, corresponding to those of Theorem 1 in (Zheng and Kwok 2016). In Algorithm 1, recall that here* $\theta \le 1$ *and* G *is defined in (9). Thus, both the first and third terms in (13) are slightly smaller than those of Theorem 1 in (Zheng and Kwok 2016). In addition, one can set* $\eta = 1/8L$ *(i.e.* $\alpha = 8$*) and* $\theta = 1 - \delta(b)/(\alpha-1) = 1 - \delta(b)/7$. *Thus, the second term in (13) equals to* $\delta(b)/7$, *while that of SVRG-ADMM is approximately equal to* $4L\eta\delta(b)/(1 - 4L\eta\delta(b)) \ge \delta(b)/2$. *In summary, the convergence bound of SVRG-ADMM can be slightly improved by ASVRG-ADMM.*

Selecting Scheme of θ

The rate ρ in (13) of Theorem 1 can be expressed as the function with respect to the parameters θ and β with given m, η, L_f, L, A, μ. Similar to (Nishihara et al. 2015; Zheng and Kwok 2016), one can obtain the optimal parameter $\beta^* = \sqrt{L_f\mu/(\sigma_{\min}(AA^T)\|A^TA\|_2)}$, which produces a smaller rate ρ. In addition, as shown in (13), all the three terms are with respect to the weight θ. Therefore, we give the following selecting scheme for θ.

Proposition 1. *Given* $\kappa_f = L_f/\mu, \beta^*, \kappa = L/\mu, b, A$, *and let* $\omega = \|A^TA\|_2/\sigma_{\min}(AA^T)$, *we set* $m > 2\kappa + 2\sqrt{\kappa_f\omega}$ *and* $\eta = 1/(L\alpha)$, *where* $\alpha = \frac{m - 2\sqrt{\kappa_f\omega}}{2\kappa} + \delta(b) + 1$. *Then the optimal* θ^* *of Algorithm 1 is given by*

$$\theta^* = \frac{m - 2\sqrt{\kappa_f\omega}}{m - 2\sqrt{\kappa_f\omega} + 2\kappa(\delta(b) + 1)}.$$

The proof of Proposition 1 is provided in the Supplementary Material.

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

We first assume that $z \in \mathcal{Z}$, where $\mathcal{Z}$ is a convex compact set with diameter $D_{\mathcal{Z}} = \sup_{z_1, z_2 \in \mathcal{Z}}\|z_1 - z_2\|$, and the dual variable λ is also bounded with $D_\lambda = \sup_{\lambda_1, \lambda_2}\|\lambda_1 - \lambda_2\|$. For Algorithm 2, we give the following result.

Theorem 2. *Using the same notation as in Lemma 2 with* $\theta_0 = 1 - \frac{\delta(b)}{\alpha-1}$, *then we have*

$$\mathbb{E}\left[P(\widetilde{x}^T, \widetilde{y}^T) + \gamma\|A\widetilde{x}^T + B\widetilde{y}^T - c\|\right]$$
$$\le \frac{4(\alpha-1)\delta(b)\left(P(\widetilde{x}^0, \widetilde{y}^0) + \gamma\|A\widetilde{x}^0 + B\widetilde{y}^0 - c\|\right)}{(\alpha - 1 - \delta(b))^2(T+1)^2} \quad (14)$$
$$+ \frac{2L\alpha\|x^* - \widetilde{x}^0\|_G^2}{m(T+1)^2} + \frac{2\beta\left(\|A^TA\|_2D_{\mathcal{Z}}^2 + 4D_\lambda^2\right)}{m(T+1)}.$$

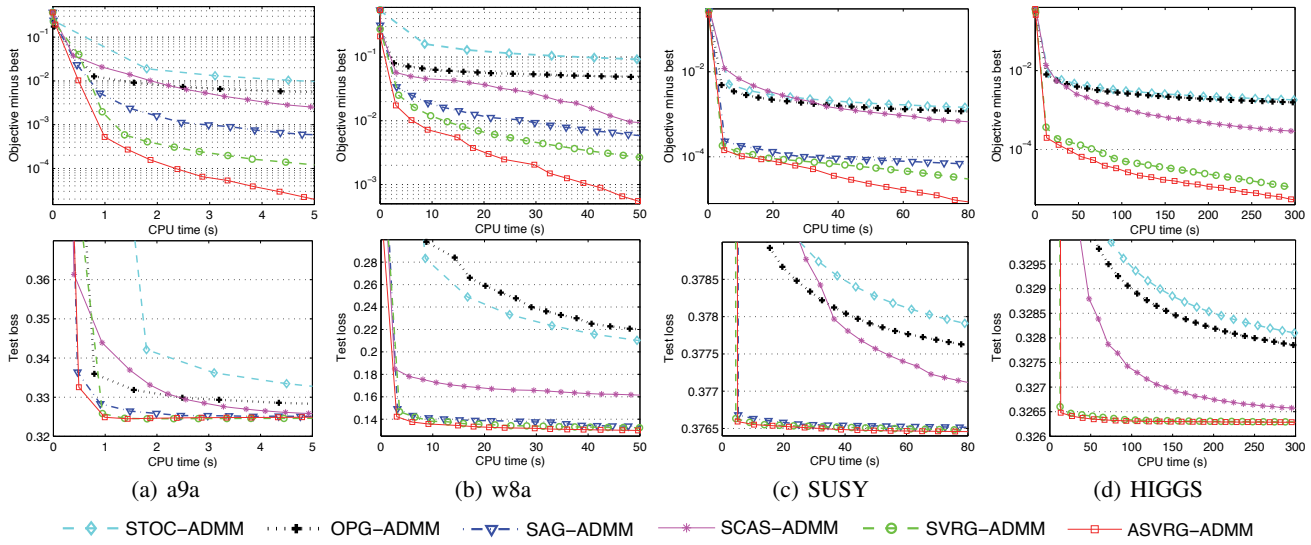

Figure 1: Comparison of different stochastic ADMM methods for graph-guided fuzed Lasso problems on the four data sets. The x-axis represents the objective value minus the minimum (top) or testing loss (bottom), and the y-axis corresponds to the running time (seconds).

The proof of Theorem 2 is provided in the Supplementary Material. Theorem 2 shows that the convergence bound consists of the three components, which converge as $\mathcal{O}(1/T^2)$, $\mathcal{O}(1/mT^2)$ and $\mathcal{O}(1/mT)$, respectively, while the three components of SVRG-ADMM converge as $\mathcal{O}(1/T)$, $\mathcal{O}(1/mT)$ and $\mathcal{O}(1/mT)$. 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 ($m \gg T$). All the components in the convergence bound of SCAS-ADMM converge as $\mathcal{O}(1/T)$. Thus, it is clear from this comparison that ASVRG-ADMM is a factor of T faster than SAG-ADMM, SVRG-ADMM and SCAS-ADMM.

Connections to Related Work

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 (14) vanishes, and ASVRG-ADMM degenerates to the batch version. Its convergence rate becomes $\mathcal{O}(D_{x^*}^2/(T+1)^2 + D_{\mathcal{Z}}^2/(T+1) + D_{\lambda}^2/(T+1))$ (which is consistent with the optimal result for accelerated deterministic ADMM methods (Goldstein et al. 2014; Lu et al. 2016)), where $D_{x^*} = \|x^* - \widetilde{x}^0\|_G$. Many empirical risk minimization problems can be viewed as the special case of (1) when $A = I$. Thus, our method can be extended to solve them, and has an $\mathcal{O}(1/T^2 + 1/(mT^2))$ rate, which is consistent with the best known result as in (Allen-Zhu 2016; Hien et al. 2016).

Experiments

In this section, we use our ASVRG-ADMM method to solve the general convex graph-guided fuzed Lasso, strongly convex graph-guided logistic regression and graph-guided SVM problems. We compare ASVRG-ADMM with the following state-of-the-art methods: STOC-ADMM (Ouyang et al. 2013), OPG-ADMM (Suzuki 2013), SAG-ADMM (Zhong and Kwok 2014b), and SCAS-ADMM (Zhao, Li, and Zhou 2015) and SVRG-ADMM (Zheng and Kwok 2016). All methods were implemented in MATLAB, and the experiments were performed on a PC with an Intel i5-2400 CPU and 16GB RAM.

Graph-Guided Fused Lasso

We first evaluate the empirical performance of the proposed method for solving the graph-guided fuzed Lasso problem:

$$\min_x \frac{1}{n}\sum_{i=1}^{n} \ell_i(x) + \lambda_1 \|Ax\|_1, \tag{15}$$

where ℓ_i is the logistic loss function on the feature-label pair (a_i, b_i), i.e., $\log(1+\exp(-b_i a_i^T x))$, and $\lambda_1 \geq 0$ is the regularization parameter. Here, we set $A = [G; I]$ as in (Ouyang et al. 2013; Zhong and Kwok 2014b; Azadi and Sra 2014; Zheng and Kwok 2016), where G is the sparsity pattern of the graph obtained by sparse inverse covariance selection (Banerjee, Ghaoui, and d'Aspremont 2008). We used four publicly available data sets[1] in our experiments, as listed in Table 2. Note that except STOC-ADMM, all the other algorithms adopted the linearization of the penalty term $\frac{\beta}{2}\|Ax - y + z\|^2$ 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. The parameters of ASVRG-ADMM are set as follows: $m = 2n/b$ and $\gamma = 1$ as in (Zhong and Kwok 2014b; Zheng and Kwok 2016), as well as η and β.

Figure 1 shows the training error (i.e. the training objective value minus the minimum) and testing loss of all the algorithms for the general convex problem on the four data sets. SAG-ADMM could not generate experimental results

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

Table 2: Summary of data sets and regularization parameters used in our experiments.

Data sets	♯ training	♯ test	♯ mini-batch	λ_1	λ_2
a9a	16,281	16,280	20	1e-5	1e-2
w8a	32,350	32,350	20	1e-5	1e-2
SUSY	3,500,000	1,500,000	100	1e-5	1e-2
HIGGS	7,700,000	3,300,000	150	1e-5	1e-2

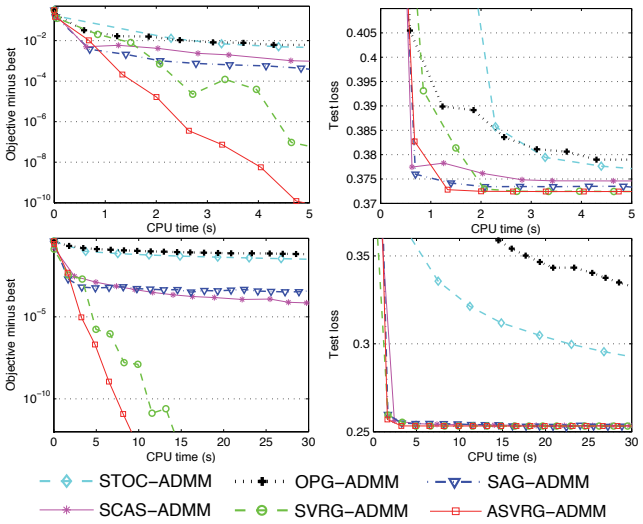

Figure 2: Comparison of different stochastic ADMM methods for graph-guided logistic regression problems on the two data sets: a9a (top) and w8a (bottom).

on the HIGGS data set because it ran out of memory. These figures clearly indicate that the variance reduced stochastic ADMM algorithms (including 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. Notably, ASVRG-ADMM consistently outperforms all other algorithms in terms of the convergence rate under all settings, which empirically verifies our theoretical result that ASVRG-ADMM has a faster convergence rate of $\mathcal{O}(1/T^2)$, as opposed to the best known rate of $\mathcal{O}(1/T)$.

Graph-Guided Logistic Regression

We further discuss the performance of ASVRG-ADMM for solving the strongly convex graph-guided logistic regression problem (Ouyang et al. 2013; Zhong and Kwok 2014a):

$$\min_x \frac{1}{n}\sum_{i=1}^{n}\left(\ell_i(x) + \frac{\lambda_2}{2}\|x\|_2^2\right) + \lambda_1\|Ax\|_1. \qquad (16)$$

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 Figure 2, from which we observe that SVRG-ADMM and ASVRG-ADMM achieve comparable performance, and they significantly outperform the other methods in terms of the convergence rate, which is consistent with their linear (geometric) convergence guarantees. Moreover, ASVRG-ADMM converges slightly faster

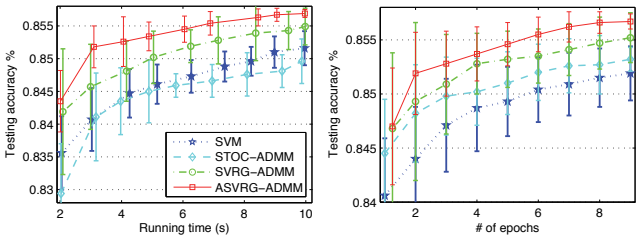

Figure 3: Comparison of accuracies multi-class classification on the 20newsgroups data set: accuracy v.s. running time (left) or number of epochs (right).

than SVRG-ADMM, which shows the effectiveness of the momentum trick to accelerate variance reduced stochastic ADMM, as we expected.

Graph-Guided SVM

Finally, we evaluate the performance of ASVRG-ADMM for solving the graph-guided SVM problem,

$$\min_x \frac{1}{n}\sum_{i=1}^{n}\left([1 - b_i a_i^T x]_+ + \frac{\lambda_2}{2}\|x\|_2^2\right) + \lambda_1\|Ax\|_1, \quad (17)$$

where $[x]_+ = \max(0, x)$ is the non-smooth hinge loss. To effectively solve problem (17), we used the smooth Huberized hinge loss in (Rosset and Zhu 2007) to approximate the hinge loss. For the 20newsgroups dataset[2], we randomly divide it into 80% training set and 20% test set. Following (Ouyang et al. 2013), we set $\lambda_1 = \lambda_2 = 10^{-5}$, and use the one-vs-rest scheme for the multi-class classification.

Figure 3 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 observe 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 again verifies the effectiveness of our ASVRG-ADMM method.

Conclusions

In this paper, we proposed an accelerated stochastic variance reduced ADMM (ASVRG-ADMM) method, in which we combined both the momentum acceleration trick for batch optimization and the variance reduction technique. We designed two different momentum term update rules for strongly convex and general convex cases, respectively. Moreover, we also theoretically analyzed the convergence properties of ASVRG-ADMM, from which it is clear that ASVRG-ADMM achieves linear convergence and $\mathcal{O}(1/T^2)$ rates for both cases. Especially, ASVRG-ADMM is at least a factor of T faster than existing stochastic ADMM methods for general convex problems.

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

Acknowledgements

We thank the reviewers for their valuable comments. The authors are supported by the Hong Kong GRF 2150851 and 2150895, and Grants 3132964 and 3132821 funded by the Research Committee of CUHK.

References

Allen-Zhu, Z. 2016. Katyusha: Accelerated variance reduction for faster SGD. *arXiv:1603.05953v4*.

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

Banerjee, O.; Ghaoui, L. E.; and d'Aspremont, A. 2008. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.* 9:485–516.

Beck, A., and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* 2(1):183–202.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* 3(1):1–122.

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

Deng, W., and Yin, W. 2016. On the global and linear convergence of the generalized alternating direction method of multipliers. *J. Sci. Comput.* 66:889–916.

Goldstein, T.; ODonoghue, B.; Setzer, S.; and Baraniuk, R. 2014. Fast alternating direction optimization methods. *SIAM J. Imaging Sciences* 7(3):1588–1623.

Hien, L. T. K.; Lu, C.; Xu, H.; and Feng, J. 2016. Accelerated stochastic mirror descent algorithms for composite non-strongly convex optimization. *arXiv:1605.06892v2*.

Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 315–323.

Kim, S.; Sohn, K. A.; and Xing, E. P. 2009. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics* 25:i204–i212.

Lu, C.; Li, H.; Lin, Z.; and Yan, S. 2016. Fast proximal linearized alternating direction method of multiplier with parallel splitting. In *Proc. 30th AAAI Conf. Artif. Intell. (AAAI)*, 739–745.

Nesterov, Y. 1983. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady* 27(2):372–376.

Nesterov, Y. 2004. *Introductory Lectures on Convex Optimization: A Basic Course*. Boston: Kluwer Academic Publ.

Nie, F.; Huang, Y.; XiaoqianWang; and Huang, H. 2014. Linear time solver for primal SVM. In *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 505–513.

Nishihara, R.; Lessard, L.; Recht, B.; Packard, A.; and Jordan, M. I. 2015. A general analysis of the convergence of ADMM. In *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 343–352.

Ouyang, H.; He, N.; Tran, L. Q.; and Gray, A. 2013. Stochastic alternating direction method of multipliers. In *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, 80–88.

Rosset, S., and Zhu, J. 2007. Piecewise linear regularized solution paths. *Ann. Statist.* 35(3):1012–1030.

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

Shalev-Shwartz, S., and Zhang, T. 2013. Stochastic dual coordinate ascent methods for regularized loss minimization. *J. Mach. Learn. Res.* 14:567–599.

Shang, F.; Liu, Y.; Cheng, J.; and Cheng, H. 2014. Robust principal component analysis with missing data. In *Proc. 23rd Int. Conf. Inform. Knowl. Manag. (CIKM)*, 1149–1158.

Suzuki, T. 2013. Dual averaging and proximal gradient descent for online alternating direction multiplier method. In *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, 392–400.

Suzuki, T. 2014. Stochastic dual coordinate ascent with alternating direction method of multipliers. In *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 736–744.

Tibshirani, R. J., and Taylor, J. 2011. The solution path of the generalized lasso. *Annals of Statistics* 39(3):1335–1371.

Tseng, P. 1998. An incremental gradient(-projection) method with momentum term and adaptive step size rule. *SIAM J. Optim.* 8(2):506–531.

Tseng, P. 2010. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Math. Program.* 125:263–295.

Wang, H., and Banerjee, A. 2012. Online alternating direction method. In *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, 1119–1126.

Zhang, X.; Burger, M.; and Osher, S. 2011. A unified primal-dual algorithm framework based on Bregman iteration. *J. Sci. Comput.* 46(1):20–46.

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

Zheng, S., and Kwok, J. T. 2016. Fast-and-light stochastic ADMM. In *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2407–2613.

Zhong, L. W., and Kwok, J. T. 2014a. Accelerated stochastic gradient method for composite regularization. In *Proc. 17th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 1086–1094.

Zhong, L. W., and Kwok, J. T. 2014b. Fast stochastic alternating direction method of multipliers. In *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 46–54.