Zeroth-Order Stochastic Alternating Direction Method of Multipliers for Nonconvex Nonsmooth Optimization

Feihu Huang[1], Shangqian Gao[1], Songcan Chen[2,3], Heng Huang[1,4*]
[1] Department of Electrical & Computer Engineering, University of Pittsburgh, USA
[2] College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics
[3] MIIT Key Laboratory of Pattern Analysis & Machine Intelligence, China
[4] JD Finance America Corporation
feh23@pitt.edu, shg84@pitt.edu, s.chen@nuaa.edu.cn, heng.huang@pitt.edu

Abstract

Alternating direction method of multipliers (ADMM) is a popular optimization tool for the composite and constrained problems in machine learning. However, in many machine learning problems such as black-box attacks and bandit feedback, ADMM could fail because the explicit gradients of these problems are difficult or infeasible to obtain. Zeroth-order (gradient-free) methods can effectively solve these problems due to that the objective function values are only required in the optimization. Recently, though there exist a few zeroth-order ADMM methods, they build on the convexity of objective function. Clearly, these existing zeroth-order methods are limited in many applications. In the paper, thus, we propose a class of fast zeroth-order stochastic ADMM methods (*i.e.*, ZO-SVRG-ADMM and ZO-SAGA-ADMM) for solving nonconvex problems with multiple nonsmooth penalties, based on the coordinate smoothing gradient estimator. Moreover, we prove that both the ZO-SVRG-ADMM and ZO-SAGA-ADMM have convergence rate of $O(1/T)$, where T denotes the number of iterations. In particular, our methods not only reach the best convergence rate $O(1/T)$ for the nonconvex optimization, but also are able to effectively solve many complex machine learning problems with multiple regularized penalties and constraints. Finally, we conduct the experiments of black-box binary classification and structured adversarial attack on black-box deep neural network to validate the efficiency of our algorithms.

1 Introduction

Alternating direction method of multipliers (ADMM [Gabay and Mercier, 1976; Boyd *et al.*, 2011]) is a popular optimization tool for solving the composite and constrained problems in machine learning. In particular, ADMM can efficiently optimize some problems with complicated structure regularization such as the graph-guided fused lasso [Kim *et al.*, 2009], which are too complicated for the other popular optimization methods such as proximal gradient methods [Beck and Teboulle, 2009]. Thus, ADMM has been widely studied in recent years [Boyd *et al.*, 2011]. For the large-scale optimization, the stochastic ADMM method [Ouyang *et al.*, 2013] has been proposed. Due to variances of the stochastic gradient, however, these methods suffer from a slow convergence rate. To speedup the convergence, recently, some faster stochastic ADMM methods [Suzuki, 2014; Zheng and Kwok, 2016] have been proposed by using the variance reduced (VR) techniques such as the SVRG [Johnson and Zhang, 2013]. In fact, ADMM is also highly successful in solving various nonconvex problems such as tensor decomposition [Jiang *et al.*, 2019] and learning neural networks [Taylor *et al.*, 2016]. Thus, some fast nonconvex stochastic ADMM methods have been developed in [Huang *et al.*, 2016].

Currently, most of the ADMM methods need to compute gradients of the loss functions over each iteration. However, in many machine learning problems, the explicit expression of gradient for objective function is difficult or infeasible to obtain. For example, in black-box situations, only prediction results (*i.e.*, function values) are provided [Chen *et al.*, 2017; Liu *et al.*, 2018b]. In bandit settings [Agarwal *et al.*, 2010], the player only receives partial feedback in terms of loss function values, so it is impossible to obtain expressive gradient of the loss function. Clearly, the classic optimization methods, based on the first-order gradient or second-order information, are not competent to these problems. Thus, zeroth-order optimization methods [Duchi *et al.*, 2015; Nesterov and Spokoiny, 2017] are developed by only using the function values in the optimization.

In the paper, we focus on using the zeroth-order methods to solve the following nonconvex nonsmooth problem:

$$\min_{x,\{y_j\}_{j=1}^k} F(x,y_{[k]}) =: \frac{1}{n}\sum_{i=1}^n f_i(x) + \sum_{j=1}^k \psi_j(y_j) \quad (1)$$

$$\text{s.t. } Ax + \sum_{j=1}^k B_j y_j = c,$$

where $A \in \mathbb{R}^{p\times d}$, $B_j \in \mathbb{R}^{p\times q}$ for all $j \in [k]$, $k \geq 1$, $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x) : \mathbb{R}^d \to \mathbb{R}$ is a *nonconvex* and smooth function, and each $\psi_j(y_j) : \mathbb{R}^q \to \mathbb{R}$ is a convex and *nonsmooth* function. In machine learning, function $f(x)$ can be

*Corresponding Author.

Table 1: Convergence properties comparison of the zeroth-order ADMM algorithms and other ones. C, NC, S, NS and mNS are the abbreviations of convex, non-convex, smooth, non-smooth and the sum of multiple non-smooth functions, respectively. T is the whole iteration number. Gaussian Smoothing Gradient Estimator (**GauSGE**), Uniform Smoothing Gradient Estimator (**UniSGE**) and Coordinate Smoothing Gradient Estimator (**CooSGE**).

Algorithm	Reference	Gradient Estimator	Problem	Convergence Rate
ZOO-ADMM	[Liu *et al.*, 2018a]	GauSGE	C(S) + C(NS)	$O(\sqrt{1/T})$
ZO-GADM	[Gao *et al.*, 2018]	UniSGE	C(S) + C(NS)	$O(\sqrt{1/T})$
RSPGF	[Ghadimi *et al.*, 2016]	GauSGE	NC(S) + C(NS)	$O(\sqrt{1/T})$
ZO-ProxSVRG ZO-ProxSAGA	[Huang *et al.*, 2019]	CooSGE	NC(S) + C(NS)	$O(1/T)$
ZO-SVRG-ADMM ZO-SAGA-ADMM	Ours	CooSGE	NC(S) + C(mNS)	$O(1/T)$

used for the empirical loss, $\sum_{j=1}^{k}\psi_j(y_j)$ for multiple structure penalties (*e.g.*, sparse + group sparse), and the constraint for encoding the structure pattern of model parameters such as graph structure. Due to the flexibility in splitting the objective function into loss $f(x)$ and each penalty $\psi_j(y_j)$, ADMM is an efficient method to solve the above constricted problem. However, in the problem (1), we only access the objective values rather than the explicit function $F(x, y_{[k]})$, thus the classic ADMM methods are unsuitable for this problem.

Recently, [Gao *et al.*, 2018; Liu *et al.*, 2018a] proposed the zeroth-order stochastic ADMM methods, which only use the objective values to optimize. However, these zeroth-order ADMM-based methods build on the convexity of objective function. Clearly, these methods are limited in many applications such as adversarial attack on black-box deep neural network (DNN). Due to that the problem (1) includes multiple nonsmooth regularization functions and constraint, the existing nonconvex zeroth-order algorithms [Liu *et al.*, 2018b; Ghadimi *et al.*, 2016; Huang *et al.*, 2019] are not suitable for this problem.

In the paper, thus, we propose a class of fast zeroth-order stochastic ADMM methods (*i.e.*, ZO-SVRG-ADMM and ZO-SAGA-ADMM) to solve the problem (1) based on the coordinate smoothing gradient estimator [Liu *et al.*, 2018b]. In particular, the ZO-SVRG-ADMM and ZO-SAGA-ADMM methods build on the SVRG [Johnson and Zhang, 2013] and SAGA [Defazio *et al.*, 2014], respectively. Moreover, we study the convergence properties of the proposed methods. Table 1 shows the convergence properties of the proposed methods and other related ones.

1.1 Challenges and Contributions

Although both SVRG and SAGA show good performances in the first-order and second-order methods, applying these techniques to the nonconvex zeroth-order ADMM method is *not trivial*. There exists at least two main **challenges**:

- Due to failure of the Fejér monotonicity of iteration, the convergence analysis of the nonconvex ADMM is generally quite difficult [Wang *et al.*, 2015]. With using the inexact zeroth-order estimated gradient, this difficulty becomes greater in the nonconvex zeroth-order ADMM methods.
- To guarantee convergence of our zeroth-order ADMM methods, we need to design a new effective *Lyapunov* function, which can not follow the existing nonconvex (stochastic) ADMM methods [Jiang *et al.*, 2019; Huang *et al.*, 2016].

Thus, we carefully establish the *Lyapunov* functions in the following theoretical analysis to ensure convergence of the proposed methods. In summary, our major **contributions** are given below:

1) We propose a class of fast zeroth-order stochastic ADMM methods (*i.e.*, ZO-SVRG-ADMM and ZO-SAGA-ADMM) to solve the problem (1).
2) We prove that both the ZO-SVRG-ADMM and ZO-SAGA-ADMM have convergence rate of $O(\frac{1}{T})$ for nonconvex nonsmooth optimization. In particular, our methods not only reach the existing best convergence rate $O(\frac{1}{T})$ for the nonconvex optimization, but also are able to effectively solve many machine learning problems with multiple complex regularized penalties.
3) Extensive experiments conducted on black-box classification and structured adversarial attack on black-box DNNs validate efficiency of the proposed algorithms.

2 Related Works

Zeroth-order (gradient-free) optimization is a powerful optimization tool for solving many machine learning problems, where the gradient of objective function is not available or computationally prohibitive. Recently, the zeroth-order optimization methods are widely applied and studied. For example, zeroth-order optimization methods have been applied to bandit feedback analysis [Agarwal *et al.*, 2010] and black-box attacks on DNNs [Chen *et al.*, 2017; Liu *et al.*, 2018b]. [Nesterov and Spokoiny, 2017] have proposed several random zeroth-order methods by using Gaussian smoothing gradient estimator. To deal with the nonsmooth regularization, [Gao *et al.*, 2018; Liu *et al.*, 2018a] have proposed the zeroth-order online/stochastic ADMM-based methods.

So far, the above algorithms mainly build on the convexity of problems. In fact, the zeroth-order methods are also highly successful in solving various nonconvex problems such as adversarial attack to black-box DNNs [Liu *et al.*, 2018b]. Thus, [Ghadimi and Lan, 2013; Liu *et al.*, 2018b; Gu *et al.*, 2018] have begun to study the zeroth-order stochastic methods for the nonconvex optimization. To deal with the nonsmooth regularization, [Ghadimi *et al.*, 2016; Huang *et al.*, 2019] have proposed some non-convex zeroth-order proximal stochastic gradient methods. However, these methods still are not well competent to some complex machine learning problems such as a task of structured adversarial attack to the black-box DNNs, which is described in the following experiment.

2.1 Notations

Let $y_{[k]} = \{y_1, \cdots, y_k\}$ and $y_{[j:k]} = \{y_j, \cdots, y_k\}$ for $j \in [k]$. Given a positive definite matrix G, $\|x\|_G^2 = x^T G x$; $\sigma_{\max}(G)$ and $\sigma_{\min}(G)$ denote the largest and smallest eigenvalues of G, respectively, and $\kappa_G = \frac{\sigma_{\max}(G)}{\sigma_{\min}(G)}$. $\sigma_{\max}^A$ and $\sigma_{\min}^A$ denote the largest and smallest eigenvalues of matrix $A^T A$.

3 Preliminaries

In the section, we begin with restating a standard ϵ-approximate stationary point of the problem (1), as in [Jiang *et al.*, 2019].

Definition 1. *Given $\epsilon > 0$, the point $(x^*, y^*_{[k]}, \lambda^*)$ is said to be an ϵ-approximate stationary point of the problems* (1)*, if it holds that*

$$\mathbb{E}\big[dist(0, \partial L(x^*, y^*_{[k]}, \lambda^*))^2\big] \leq \epsilon, \quad (2)$$

where $L(x, y_{[k]}, \lambda) = f(x) + \sum_{j=1}^k \psi_j(y_j) - \langle \lambda, Ax + \sum_{j=1}^k B_j y_j - c \rangle$*,*

$$\partial L(x, y_{[k]}, \lambda) = \begin{bmatrix} \nabla_x L(x, y_{[k]}, \lambda) \\ \partial_{y_1} L(x, y_{[k]}, \lambda) \\ \cdots \\ \partial_{y_k} L(x, y_{[k]}, \lambda) \\ -Ax - \sum_{j=1}^k B_j y_j + c \end{bmatrix},$$

$dist(0, \partial L) = \inf_{L' \in \partial L} \|0 - L'\|$.

Next, we make some mild assumptions regarding problem (1) as follows:

Assumption 1. *Each function $f_i(x)$ is L-smooth for $\forall i \in \{1, 2, \cdots, n\}$ such that*

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

which is equivalent to

$$f_i(x) \leq f_i(y) + \nabla f_i(y)^T (x - y) + \frac{L}{2}\|x - y\|^2.$$

Assumption 2. *Gradient of each function $f_i(x)$ is bounded, i.e., there exists a constant $\delta > 0$ such that for all x, it follows that $\|\nabla f_i(x)\|^2 \leq \delta^2$.*

Assumption 3. *$f(x)$ and $\psi_j(y_j)$ for all $j \in [k]$ are all lower bounded, and denote $f^* = \inf_x f(x)$ and $\psi_j^* = \inf_y \psi_j(y)$ for $j \in [k]$.*

Assumption 4. *A is a full row or column rank matrix.*

Assumption 1 has been commonly used in the convergence analysis of nonconvex algorithms [Ghadimi *et al.*, 2016]. Assumption 2 is widely used for stochastic gradient-based and ADMM-type methods [Boyd *et al.*, 2011]. Assumptions 3 and 4 are usually used in the convergence analysis of ADMM methods [Jiang *et al.*, 2019; Huang *et al.*, 2016]. Without loss of generality, we will use the full column rank of matrix A in the rest of this paper.

4 Fast Zeroth-Order Stochastic ADMMs

In this section, we propose a class of zeroth-order stochastic ADMM methods to solve the problem (1). First, we define an augmented Lagrangian function of the problem (1) as follows:

$$\mathcal{L}_\rho(x, y_{[k]}, \lambda) = f(x) + \sum_{j=1}^k \psi_j(y_j) - \langle \lambda, Ax + \sum_{j=1}^k B_j y_j - c \rangle + \frac{\rho}{2}\|Ax + \sum_{j=1}^k B_j y_j - c\|^2, \quad (3)$$

where $\lambda \in \mathbb{R}^p$ and $\rho > 0$ denotes the dual variable and penalty parameter, respectively.

In the problem (1), the explicit expression of objective function $f_i(x)$ is not available, and only the function value of $f_i(x)$ is available. To avoid computing explicit gradient, thus, we use the coordinate smoothing gradient estimator [Liu *et al.*, 2018b] to estimate gradients: for $i \in [n]$,

$$\hat{\nabla} f_i(x) = \sum_{j=1}^d \frac{1}{2\mu_j}\big(f_i(x + \mu_j e_j) - f_i(x - \mu_j e_j)\big)e_j, \quad (4)$$

where μ_j is a coordinate-wise smoothing parameter, and e_j is a standard basis vector with 1 at its j-th coordinate, and 0 otherwise.

Algorithm 1 Nonconvex ZO-SVRG-ADMM Algorithm

Input: $b, m, T, S = [T/m], \eta > 0$ and $\rho > 0$;
Initialize: $x_0^1, y_j^{0,1}$ for $j \in [k]$ and λ_0^1;
for $s = 1, 2, \cdots, S$ **do**
  $\tilde{x}^{s+1} = x_0^{s+1}$, $\hat{\nabla} f(\tilde{x}^s) = \frac{1}{n}\sum_{i=1}^n \hat{\nabla} f_i(\tilde{x}^s)$;
  for $t = 0, 1, \cdots, m-1$ **do**
    Uniformly randomly pick a mini-batch $\mathcal{I}_t$ (with replacement) from $\{1, 2, \cdots, n\}$, and $|\mathcal{I}_t| = b$;
    Using (4) to estimate stochastic gradient $\hat{g}_t^s = \hat{\nabla} f_{\mathcal{I}_t}(x_t^s) - \hat{\nabla} f_{\mathcal{I}_t}(\tilde{x}^s) + \hat{\nabla} f(\tilde{x}^s)$;
    $y_j^{s,t+1} = \arg\min_{y_j} \{\mathcal{L}_\rho(x_t^s, y_{[j-1]}^{s,t+1}, y_j, y_{[j+1:k]}^{s,t}, \lambda_t^s) + \frac{1}{2}\|y_j - y_j^{s,t}\|^2_{H_j}\}$, for all $j \in [k]$;
    $x_{t+1}^s = \arg\min_x \hat{\mathcal{L}}_\rho\big(x, y_{[k]}^{s,t+1}, \lambda_t^s, \hat{g}_t^s\big)$;
    $\lambda_{t+1}^s = \lambda_t^s - \rho(Ax_{t+1}^s + \sum_{j=1}^k B_j y_j^{s,t+1} - c)$;
  end for
  $x_0^{s+1} = x_m^s$, $y_j^{s+1,0} = y_j^{s,m}$ for $j \in [k]$, $\lambda_0^{s+1} = \lambda_m^s$;
end for
Output: Chosen uniformly from $\{(x_t^s, y_{[k]}^{s,t}, \lambda_t^s)_{t=1}^m\}_{s=1}^S$.

Based on the above estimated gradients, we propose a zeroth-order ADMM (ZO-ADMM) method to solve the problem (1) by executing the following iterations, for $t = 1, 2, \cdots$

$$\begin{cases} y_j^{t+1} = \arg\min_{y_j} \big\{\mathcal{L}_\rho(x_t, y_{[j-1]}^{t+1}, y_j, y_{[j+1:k]}^t, \lambda_t) + \frac{1}{2}\|y_j - y_j^t\|^2_{H_j}\big\}, \ \forall j \in [k] \\ x_{t+1} = \arg\min_x \hat{\mathcal{L}}_\rho(x, y_{t+1}, \lambda_t, \hat{\nabla} f(x)) \\ \lambda_{t+1} = \lambda_t - \rho(Ax_{t+1} + By_{t+1} - c), \end{cases} \quad (5)$$

where the term $\frac{1}{2}\|y_j - y_j^t\|^2_{H_j}$ with $H_j \succ 0$ to linearize the term $\|Ax+\sum_{j=1}^k B_j y_j - c\|^2$. Here, due to using the inexact zeroth-order gradient to update x, we define an approximate function over x_t as follows:

$$\hat{\mathcal{L}}_\rho\big(x, y_{[k]}^{t+1}, \lambda_t, \hat{\nabla} f(x)\big) = f(x_t) + \hat{\nabla} f(x)^T (x - x_t) + \frac{1}{2\eta}\|x - x_t\|^2_G + \sum_{j=1}^k \psi_j(y_j^{t+1}) - \lambda_t^T(Ax + \sum_{j=1}^k B_j y_j^{t+1} - c) + \frac{\rho}{2}\|Ax + \sum_{j=1}^k B_j y_j^{t+1} - c\|^2, \quad (6)$$

where $G \succ 0$, $\hat{\nabla} f(x)$ is the zeroth-order gradient and $\eta > 0$ is a step size. Considering the matrix A^TA is large, set $G = rI - \rho\eta A^TA \succ I$ with $r > \rho\eta\sigma_{\max}(A^TA) + 1$ to linearize the term $\|Ax + \sum_{j=1}^k B_j y_j^{t+1} - c\|^2$.

Algorithm 2 Nonconvex ZO-SAGA-ADMM Algorithm

Input: b, T, $\eta > 0$ and $\rho > 0$;
Initialize: $z_i^0 = x_0$ for $i \in \{1, 2, \cdots, n\}$, $\hat{\phi}_0 = \frac{1}{n}\sum_{i=1}^n \nabla f_i(z_i^0)$, y_j^0 for $j \in [k]$ and λ_0;
for $t = 0, 1, \cdots, T-1$ **do**
  Uniformly randomly pick a mini-batch $\mathcal{I}_t$ (with replacement) from $\{1, 2, \cdots, n\}$, and $|\mathcal{I}_t| = b$;
  Using (4) to estimate stochastic gradient $\hat{g}_t = \frac{1}{b}\sum_{i_t\in\mathcal{I}_t}\big(\nabla f_{i_t}(x_t) - \nabla f_{i_t}(z_{i_t}^t)\big) + \hat{\phi}_t$ with $\hat{\phi}_t = \frac{1}{n}\sum_{i=1}^n \nabla f_i(z_i^t)$;
  $y_j^{t+1} = \arg\min_{y_j} \big\{\mathcal{L}_\rho(x_t, y_{[j-1]}^{t+1}, y_j, y_{[j+1:k]}^t, \lambda_t) + \frac{1}{2}\|y_j - y_j^t\|^2_{H_j}\big\}$, for all $j \in [k]$;
  $x_{t+1} = \arg\min_x \hat{\mathcal{L}}_\rho\big(x, y_{[k]}^{t+1}, \lambda_t, \hat{g}_t\big)$;
  $\lambda_{t+1} = \lambda_t - \rho(Ax_{t+1} + \sum_{j=1}^k B_j y_j^{t+1} - c)$;
  $z_{i_t}^{t+1} = x_t$ for $i \in \mathcal{I}_t$ and $z_i^{t+1} = z_i^t$ for $i \notin \mathcal{I}_t$;
  $\hat{\phi}_{t+1} = \hat{\phi}_t - \frac{1}{n}\sum_{i_t\in\mathcal{I}_t}\big(\nabla f_{i_t}(z_{i_t}^t) - \nabla f_{i_t}(z_{i_t}^{t+1})\big)$;
end for
Output: Chosen uniformly from $\{x_t, y_{[k]}^t, \lambda_t\}_{t=1}^T$.

In the problem (1), not only the noisy gradient of $f_i(x)$ is not available, but also the sample size n is very large. Thus, we propose fast ZO-SVRG-ADMM and ZO-SAGA-ADMM to solve the problem (1), based on the SVRG and SAGA, respectively.

Algorithm 1 shows the algorithmic framework of ZO-SVRG-ADMM. In Algorithm 1, we use the estimated stochastic gradient $\hat{g}_t^s = \hat{\nabla} f_{\mathcal{I}_t}(x_t^s) - \hat{\nabla} f_{\mathcal{I}_t}(\tilde{x}^s) + \hat{\nabla} f(\tilde{x}^s)$ with $\hat{\nabla} f_{\mathcal{I}_t}(x_t^s) = \frac{1}{b}\sum_{i_t\in\mathcal{I}_t}\hat{\nabla} f_{i_t}(x_t^s)$. We have $\mathbb{E}_{\mathcal{I}_t}[\hat{g}_t^s] = \hat{\nabla} f(x_t^s) \neq \nabla f(x_t^s)$, *i.e.*, this stochastic gradient is a **biased** estimate of the true full gradient. Although the SVRG has shown a great promise, it relies upon the assumption that the stochastic gradient is an **unbiased** estimate of true full gradient. Thus, adapting the similar ideas of SVRG to zeroth-order ADMM optimization is not a trivial task. To handle this challenge, we choose the appropriate step size η, penalty parameter ρ and smoothing parameter μ to guarantee the convergence of our algorithms, which will be discussed in the following convergence analysis.

Algorithm 2 shows the algorithmic framework of ZO-SAGA-ADMM. In Algorithm 2, we use the estimated stochastic gradient $\hat{g}_t = \frac{1}{b}\sum_{i_t\in\mathcal{I}_t}\big(\hat{\nabla} f_{i_t}(x_t) - \nabla f_{i_t}(z_{i_t}^t)\big) + \hat{\phi}_t$ with $\hat{\phi}_t = \frac{1}{n}\sum_{i=1}^n \hat{\nabla} f_i(z_i^t)$. Similarly, we have $\mathbb{E}_{\mathcal{I}_t}[\hat{g}_t] = \hat{\nabla} f(x_t) \neq \nabla f(x_t)$.

5 Convergence Analysis

In this section, we will study the convergence properties of the proposed algorithms (ZO-SVRG-ADMM and ZO-SAGA-ADMM). For notational simplicity, let

$$\nu_1 = \frac{L}{4} + \frac{9L^2}{\sigma_{\min}^A},\ \nu_2 = k\big(\rho^2\sigma_{\max}^B\sigma_{\max}^A + \rho^2(\sigma_{\max}^B)^2 + \sigma_{\max}^2(H)\big),$$
$$\nu_3 = 6L^2 + \frac{3\sigma_{\max}^2(G)}{\eta^2},\ \nu_4 = \frac{18L^2}{\sigma_{\min}^A\rho^2} + \frac{3\sigma_{\max}^2(G)}{\sigma_{\min}^A\eta^2\rho^2}.$$

5.1 Convergence Analysis of ZO-SVRG-ADMM

In this subsection, we analyze convergence properties of the ZO-SVRG-ADMM.

Given the sequence $\{(x_t^s, y_{[k]}^{s,t}, \lambda_t^s)_{t=1}^m\}_{s=1}^S$ is generated from Algorithm 1, we define a *Lyapunov* function:

$$R_t^s = \mathbb{E}\big[\mathcal{L}_\rho(x_t^s, y_{[k]}^{s,t}, \lambda_t^s) + (\tfrac{3\sigma_{\max}^2(G)}{\sigma_{\min}^A\eta^2\rho} + \tfrac{9L^2}{\sigma_{\min}^A\rho})\|x_t^s - x_{t-1}^s\|^2 + \frac{18L^2 d}{\sigma_{\min}^A\rho b}\|x_{t-1}^s - \tilde{x}^s\|^2 + c_t\|x_t^s - \tilde{x}^s\|^2\big],$$

where the positive sequence $\{c_t\}$ satisfies

$$c_t = \begin{cases} \dfrac{36L^2 d}{\sigma_{\min}^A\rho b} + \dfrac{2Ld}{b} + (1+\beta)c_{t+1},\ 1 \le t \le m, \\ 0,\ t \ge m+1. \end{cases}$$

In addition, we definite a useful variable $\theta_t^s = \mathbb{E}\big[\|x_{t+1}^s - x_t^s\|^2 + \|x_t^s - x_{t-1}^s\|^2 + \frac{d}{b}(\|x_t^s - \tilde{x}^s\|^2 + \|x_{t-1}^s - \tilde{x}^s\|^2) + \sum_{j=1}^k \|y_j^{s,t} - y_j^{s,t+1}\|^2\ \big]$.

Theorem 1. *Suppose the sequence $\{(x_t^s, y_{[k]}^{s,t}, \lambda_t^s)_{t=1}^m\}_{s=1}^S$ is generated from Algorithm 1. Let $m = [n^{\frac{1}{3}}]$, $b = [d^{1-l}n^{\frac{2}{3}}]$, $l \in \{0, \frac{1}{2}, 1\}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{9d^l L}$ $(0 < \alpha \le 1)$ and $\rho = \frac{6\sqrt{71}\kappa_G d^l L}{\sigma_{\min}^A\alpha}$, then we have*

$$\min_{s,t}\mathbb{E}\big[dist(0, \partial L(x_t^s, y_{[k]}^{s,t}, \lambda_t^s))^2\big] \le O(\frac{d^{2l}}{T}) + O(d^{2+2l}\mu^2),$$

where $\gamma = \min(\sigma_{\min}^H, \chi_t, L)$ with $\chi_t \ge \frac{3\sqrt{71}\kappa_G d^l L}{2\alpha}$, $\nu_{\max} = \max(\nu_2, \nu_3, \nu_4)$ and R^* is a lower bound of function R_t^s. It follows that suppose the smoothing parameter μ and the whole iteration number $T = mS$ satisfy

$$\frac{1}{\mu^2} \ge \frac{2d^{2+2l}}{\epsilon}\max\big\{\nu_1\nu_2 + \frac{3L^2}{2}, \nu_1\nu_3 + \frac{9L^2}{\sigma_{\min}^A\rho^2}, \nu_1\nu_4\big\},$$
$$T = \frac{4\nu_{\max}(R_0^1 - R^*)}{\epsilon\gamma},$$

then $(x_{t^*}^{s^*}, y_{[k]}^{s^*,t^*}, \lambda_{t^*}^{s^*})$ is an ϵ-approximate stationary point of the problems (1)*, where $(t^*, s^*) = \arg\min_{t,s}\theta_t^s$.*

Remark 1. *Theorem 1 shows that given* $m = n^{\frac{1}{3}}$, $b = d^{1-l}n^{\frac{2}{3}}$, $l \in \{0, \frac{1}{2}, 1\}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{9d^lL}$ $(0 < \alpha \leq 1)$, $\rho = \frac{6\sqrt{71}\kappa_G d^l L}{\sigma^A_{\min}\alpha}$ *and* $\mu = O(\frac{1}{d\sqrt{T}})$, *the ZO-SVRG-ADMM has convergence rate of* $O(\frac{d^{2l}}{T})$. *Specifically, when* $1 \leq d < n^{\frac{1}{3}}$, *given* $l = 0$, *the ZO-SVRG-ADMM has convergence rate of* $O(\frac{1}{T})$; *when* $n^{\frac{1}{3}} \leq d < n^{\frac{2}{3}}$, *given* $l = \frac{1}{2}$, *it has convergence rate of* $O(\frac{\sqrt{d}}{T})$; *when* $n^{\frac{2}{3}} \leq d$, *given* $l = 1$, *it has convergence rate of* $O(\frac{d}{T})$. *Thus, the ZO-SVRG-ADMM has the optimal function query complexity of* $O(dn + d^2 n^{\frac{2}{3}}\epsilon^{-1})$ *for finding an* ϵ*-approximate local solution.*

5.2 Convergence Analysis of ZO-SAGA-ADMM

In this subsection, we provide the convergence analysis of the ZO-SAGA-ADMM.

Given the sequence $\{x_t, y^t_{[k]}, \lambda_t\}_{t=1}^T$ is generated from Algorithm 2, we define a *Lyapunov* function

$$\Omega_t = \mathbb{E}\big[\mathcal{L}_\rho(x_t, y^t_{[k]}, \lambda_t) + (\frac{3\sigma^2_{\max}(G)}{\sigma^A_{\min}\rho\eta^2} + \frac{9L^2}{\sigma^A_{\min}\rho})\|x_t - x_{t-1}\|^2 + \frac{18L^2 d}{\sigma^A_{\min}\rho b}\frac{1}{n}\sum_{i=1}^n \|x_{t-1} - z_i^{t-1}\|^2 + c_t \frac{1}{n}\sum_{i=1}^n \|x_t - z_i^t\|^2\big],$$

where the positive sequence $\{c_t\}$ satisfies

$$c_t = \begin{cases} \frac{36L^2 d}{\sigma^A_{\min}\rho b} + \frac{2Ld}{b} + (1-p)(1+\beta)c_{t+1}, \ 0 \leq t \leq T-1, \\ 0, \ t \geq T. \end{cases}$$

In addition, we definite a useful variable $\theta_t = \mathbb{E}\big[\|x_{t+1} - x_t\|^2 + \|x_t - x_{t-1}\|^2 + \frac{d}{bn}\sum_{i=1}^n(\|x_t - z_i^t\|^2 + \|x_{t-1} - z_i^{t-1}\|^2) + \sum_{j=1}^k \|y_j^t - y_j^{t+1}\|^2$.

Theorem 2. *Suppose the sequence* $\{x_t, y^t_{[k]}, \lambda_t\}_{t=1}^T$ *is generated from Algorithm 2. Let* $b = n^{\frac{2}{3}}d^{\frac{1-l}{3}}$, $l \in \{0, \frac{1}{2}, 1\}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{33d^lL}$ $(0 < \alpha \leq 1)$ *and* $\rho = \frac{6\sqrt{791}\kappa_G d^l L}{\sigma^A_{\min}\alpha}$ *then we have*

$$\min_{1\leq t\leq T} \mathbb{E}\big[dist(0, \partial L(x_t, y^t_{[k]}, \lambda_t))^2\big] \leq O(\frac{d^{2l}}{T}) + O(d^{2+2l}\mu^2),$$

where $\gamma = \min(\sigma^H_{\min}, \chi_t, L)$ *with* $\chi_t \geq \frac{3\sqrt{791}\kappa_G d^l L}{2\alpha}$, $\nu_{\max} = \max(\nu_2, \nu_3, \nu_4)$ *and* Ω^* *is a lower bound of function* Ω_t. *It follows that suppose the parameters* μ *and* T *satisfy*

$$\frac{1}{\mu^2} \geq \frac{2d^{2+2l}}{\epsilon}\max\big\{\nu_1\nu_2 + \frac{3L^2}{2}, \nu_1\nu_3 + \frac{9L^2}{\sigma^A_{\min}\rho^2}, \nu_1\nu_4\big\},$$
$$T = \frac{4\kappa_{\max}}{\epsilon\gamma}(\Omega_0 - \Omega^*),$$

then $(x_{t^*}, y^{t^*}_{[k]}, \lambda_{t^*})$ *is an* ϵ*-approximate stationary point of the problems* (1), *where* $t^* = \arg\min_{1\leq t\leq T}\theta_t$.

Remark 2. *Theorem 2 shows that* $b = n^{\frac{2}{3}}d^{\frac{1-l}{3}}$, $l \in \{0, \frac{1}{2}, 1\}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{33d^lL}$ $(0 < \alpha \leq 1)$, $\rho = \frac{6\sqrt{791}\kappa_G d^l L}{\sigma^A_{\min}\alpha}$ *and* $\mu = O(\frac{1}{d\sqrt{T}})$, *the ZO-SAGA-ADMM has the* $O(\frac{d^{2l}}{T})$ *of convergence rate. Specifically, when* $1 \leq d < n$, *given* $l = 0$, *the ZO-SAGA-ADMM has convergence rate of* $O(\frac{1}{T})$; *when* $n \leq d < n^2$, *given* $l = \frac{1}{2}$, *it has convergence rate of* $O(\frac{d}{T})$; *when* $n^2 \leq d$, *given* $l = 1$, *it has convergence rate of* $O(\frac{d^2}{T})$. *Thus, the ZO-SAGA-ADMM has the optimal function query complexity of* $O(dn + d^{\frac{4}{3}}n^{\frac{2}{3}}\epsilon^{-1})$ *for finding an* ϵ*-approximate local solution.*

6 Experiments

In this section, we compare our algorithms (ZO-SVRG-ADMM, ZO-SAGA-ADMM) with the ZO-ProxSVRG, ZO-ProxSAGA [Huang *et al.*, 2019], the deterministic zeroth-order ADMM (ZO-ADMM), and zeroth-order stochastic ADMM (ZO-SGD-ADMM) without variance reduction on two applications: 1) robust black-box binary classification, and 2) structured adversarial attacks on black-box DNNs.

Table 2: Real Datasets for Black-Box Binary Classification

datasets	#samples	#features	#classes
20news	16,242	100	2
a9a	32,561	123	2
w8a	64,700	300	2
covtype.binary	581,012	54	2

6.1 Robust Black-Box Binary Classification

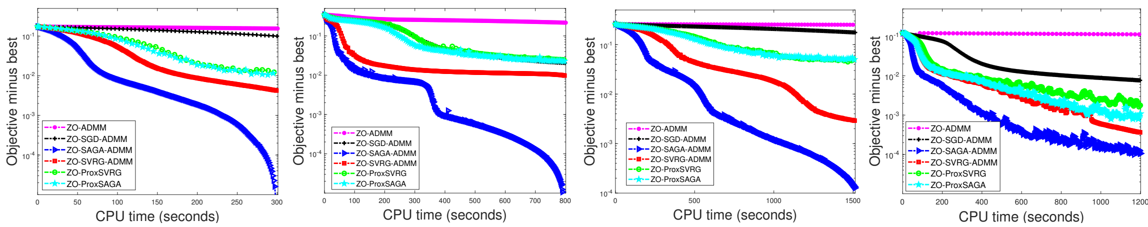
In this subsection, we focus on a robust black-box binary classification task with graph-guided fused lasso. Given a set of training samples $(a_i, l_i)_{i=1}^n$, where $a_i \in \mathbb{R}^d$ and $l_i \in \{-1, +1\}$, we find the optimal parameter $x \in \mathbb{R}^d$ by solving the problem:

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n f_i(x) + \tau_1\|x\|_1 + \tau_2\|\hat{G}x\|_1, \tag{7}$$

where $f_i(x)$ is the black-box loss function, that only returns the function value given an input. Here, we specify the loss function $f_i(x) = \frac{\sigma^2}{2}\big(1 - \exp(-\frac{(l_i - a_i^T x)^2}{\sigma^2})\big)$, which is the *nonconvex* robust correntropy induced loss [He *et al.*, 2011]. Matrix $\hat{G}$ decodes the sparsity pattern of graph obtained by sparse inverse covariance selection, as in [Ouyang *et al.*, 2013]. In the experiment, we give mini-batch size $b = 20$, smoothing parameter $\mu = \frac{1}{d\sqrt{t}}$ and penalty parameters $\tau_1 = \tau_2 = 10^{-5}$.

In the experiment, we use some public real datasets[1], which are summarized in Table 2. For each dataset, we use half of the samples as training data and the rest as testing data. Figure 1 shows that the objective values of our algorithms faster decrease than the other algorithms, as the CPU time increases. In particular, our algorithms show better performances than the zeroth-order proximal algorithms. It is relatively difficult that these zeroth-order proximal methods deal with the nonsmooth penalties in the problem (7). Thus, we have to use some iterative methods (such as the classic ADMM method) to solve the proximal operator in these proximal methods.

[1] *20news* is from https://cs.nyu.edu/~roweis/data.html; others are from www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.



(a) 20news (b) a9a (c) w8a (d) covtype.binary

Figure 1: Objective value gaps *versus* CPU time on benchmark datasets.

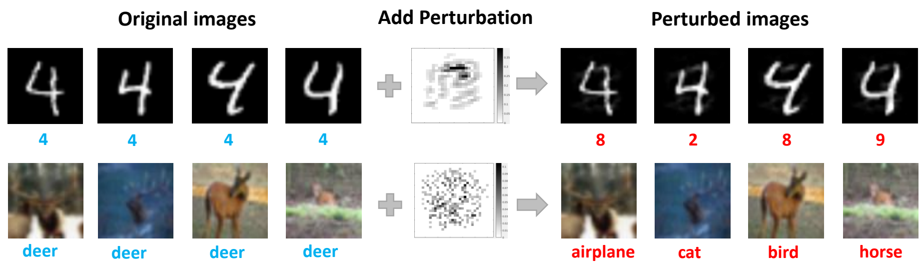


Figure 2: Group-sparsity perturbations are learned from MNIST and CIFAR-10 datasets. Blue and red labels denote the initial label, and the label after attack, respectively.

6.2 Structured Attacks on Black-Box DNNs

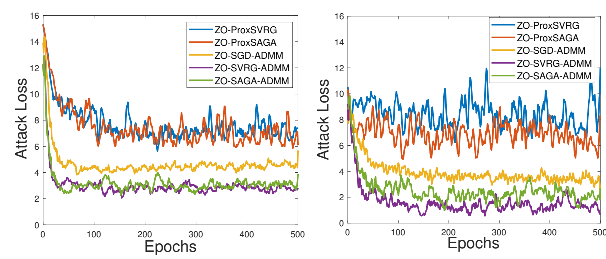
In this subsection, we use our algorithms to generate adversarial examples to attack the pre-trained DNN models, whose parameters are hidden from us and only its outputs are accessible. Moreover, we consider an interesting problem: "What possible structures could adversarial perturbations have to fool black-box DNNs ?" Thus, we use the zeroth-order algorithms to find an universal structured adversarial perturbation $x \in \mathbb{R}^d$ that could fool the samples $\{a_i \in \mathbb{R}^d,\ l_i \in \mathbb{N}\}_{i=1}^n$, which can be regarded as the following problem:

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max\left\{F_{l_i}(a_i+x) - \max_{j\neq l_i} F_j(a_i+x), 0\right\} + \tau_1 \sum_{p=1}^{P}\sum_{q=1}^{Q} \|x_{\mathcal{G}_{p,q}}\|_2 + \tau_2\|x\|_2^2 + \tau_3 h(x), \tag{8}$$

where $F(a)$ represents the final layer output before softmax of neural network, and $h(x)$ ensures the validness of created adversarial examples. Specifically, $h(x) = 0$ if $a_i + x \in [0,1]^d$ for all $i \in [n]$ and $\|x\|_\infty \leq \epsilon$, otherwise $h(x) = \infty$. Following [Xu *et al.*, 2018], we use the overlapping lasso to obtain structured perturbations. Here, the overlapping groups $\{\mathcal{G}_{p,q}\}$, $p = 1, \cdots, P$, $q = 1, \cdots, Q$ generate from dividing an image into sub-groups of pixels.

In the experiment, we use the pre-trained DNN models on MNIST and CIFAR-10 as the target black-box models, which can attain 99.4% and 80.8% test accuracy, respectively. For MNIST, we select 20 samples from a target class and set batch size $b = 4$; For CIFAR-10, we select 30 samples and set $b = 5$. In the experiment, we set $\mu = \frac{1}{d\sqrt{t}}$, where $d = 28\times 28$ and $d = 3\times 32\times 32$ for MNIST and CIFAR-10, respectively. At the same time, we set the parameters $\epsilon = 0.4$, $\tau_1 = 1$, $\tau_2 = 2$ and $\tau_3 = 1$. For both datasets, the kernel size for overlapping group lasso is set to 3×3 and the stride is one.

Figure 3 shows that attack losses (*i.e.* the first term of the problem (8)) of our methods faster decrease than the other methods, as the number of iteration increases. Figure 2 shows that our algorithms can learn some structure perturbations, and can successfully attack the corresponding DNNs.



(a) *MNIST* (b) *CIFAR-10*

Figure 3: Attack loss on adversarial attacks black-box DNNs.

7 Conclusions

In the paper, we proposed fast ZO-SVRG-ADMM and ZO-SAGA-ADMM methods based on the coordinate smoothing gradient estimator, which only uses the objective function values to optimize. Moreover, we prove that the proposed methods have a convergence rate of $O(\frac{1}{T})$. In particular, our methods not only reach the existing best convergence rate $O(\frac{1}{T})$ for the nonconvex optimization, but also are able to effectively solve many machine learning problems with the complex nonsmooth regularizations.

Acknowledgments

F.H., S.G., H.H. were partially supported by U.S. NSF IIS 1836945, IIS 1836938, DBI 1836866, IIS 1845666, IIS 1852606, IIS 1838627, IIS 1837956. S.C. was partially supported by the NSFC under Grant No. 61806093 and No. 61682281, and the Key Program of NSFC under Grant No. 61732006.

References

[Agarwal *et al.*, 2010] Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer, 2010.

[Beck and Teboulle, 2009] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[Boyd *et al.*, 2011] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[Chen *et al.*, 2017] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[Defazio *et al.*, 2014] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654, 2014.

[Duchi *et al.*, 2015] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE TIT*, 61(5):2788–2806, 2015.

[Gabay and Mercier, 1976] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[Gao *et al.*, 2018] Xiang Gao, Bo Jiang, and Shuzhong Zhang. On the information-adaptive variants of the admm: an iteration complexity perspective. *Journal of Scientific Computing*, 76(1):327–363, 2018.

[Ghadimi and Lan, 2013] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[Ghadimi *et al.*, 2016] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, 2016.

[Gu *et al.*, 2018] Bin Gu, Zhouyuan Huo, Cheng Deng, and Heng Huang. Faster derivative-free stochastic algorithm for shared memory machines. In *ICML*, pages 1807–1816, 2018.

[He *et al.*, 2011] Ran He, Wei-Shi Zheng, and Bao-Gang Hu. Maximum correntropy criterion for robust face recognition. *IEEE TPAMI*, 33(8):1561–1576, 2011.

[Huang *et al.*, 2016] Feihu Huang, Songcan Chen, and Zhaosong Lu. Stochastic alternating direction method of multipliers with variance reduction for nonconvex optimization. *arXiv preprint arXiv:1610.02758*, 2016.

[Huang *et al.*, 2019] Feihu Huang, Bin Gu, Zhouyuan Huo, Songcan Chen, and Heng Huang. Faster gradient-free proximal stochastic methods for nonconvex nonsmooth optimization. In *AAAI*, 2019.

[Jiang *et al.*, 2019] Bo Jiang, Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis. *Computational Optimization and Applications*, 72(1):115–157, 2019.

[Johnson and Zhang, 2013] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

[Kim *et al.*, 2009] Seyoung Kim, Kyung-Ah Sohn, and Eric P Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12):i204–i212, 2009.

[Liu *et al.*, 2018a] Sijia Liu, Jie Chen, Pin-Yu Chen, and Alfred Hero. Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications. In *AISTATS*, volume 84, pages 288–297, 2018.

[Liu *et al.*, 2018b] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. In *NIPS*, pages 3731–3741, 2018.

[Nesterov and Spokoiny, 2017] Yurii Nesterov and Vladimir G. Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

[Ouyang *et al.*, 2013] Hua Ouyang, Niao He, Long Tran, and Alexander G Gray. Stochastic alternating direction method of multipliers. *ICML*, 28:80–88, 2013.

[Suzuki, 2014] Taiji Suzuki. Stochastic dual coordinate ascent with alternating direction method of multipliers. In *ICML*, pages 736–744, 2014.

[Taylor *et al.*, 2016] Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit Patel, and Tom Goldstein. Training neural networks without gradients: a scalable admm approach. In *ICML*, pages 2722–2731, 2016.

[Wang *et al.*, 2015] Fenghui Wang, Wenfei Cao, and Zongben Xu. Convergence of multi-block bregman admm for nonconvex composite problems. *arXiv preprint arXiv:1505.03063*, 2015.

[Xu *et al.*, 2018] Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. *arXiv preprint arXiv:1808.01664*, 2018.

[Zheng and Kwok, 2016] Shuai Zheng and James T Kwok. Fast-and-light stochastic admm. In *IJCAI*, pages 2407–2613, 2016.