# Faster Stochastic Alternating Direction Method of Multipliers for Nonconvex Optimization

**Feihu Huang** [1] **Songcan Chen** [2 3] **Heng Huang** [1 4]

## Abstract

In this paper, we propose a faster stochastic alternating direction method of multipliers (ADMM) for nonconvex optimization by using a new stochastic path-integrated differential estimator (SPIDER), called as SPIDER-ADMM. Moreover, we prove that the SPIDER-ADMM achieves a record-breaking incremental first-order oracle (IFO) complexity of $\mathcal{O}(n + n^{1/2}\epsilon^{-1})$ for finding an $\epsilon$-approximate solution, which improves the deterministic ADMM by a factor $\mathcal{O}(n^{1/2})$, where $n$ denotes the sample size. As one of major contribution of this paper, we provide a new theoretical analysis framework for nonconvex stochastic ADMM methods with providing the optimal IFO complexity. Based on this new analysis framework, we study the unsolved optimal IFO complexity of the existing non-convex SVRG-ADMM and SAGA-ADMM methods, and prove they have the optimal IFO complexity of $\mathcal{O}(n + n^{2/3}\epsilon^{-1})$. Thus, the SPIDER-ADMM improves the existing stochastic ADMM methods by a factor of $\mathcal{O}(n^{1/6})$. Moreover, we extend SPIDER-ADMM to the online setting, and propose a faster online SPIDER-ADMM. Our theoretical analysis shows that the online SPIDER-ADMM has the IFO complexity of $\mathcal{O}(\epsilon^{-\frac{3}{2}})$, which improves the existing best results by a factor of $\mathcal{O}(\epsilon^{\frac{1}{2}})$. Finally, the experimental results on benchmark datasets validate that the proposed algorithms have faster convergence rate than the existing ADMM algorithms for nonconvex optimization.

[1]Department of Electrical & Computer Engineering, University of Pittsburgh, PA 15261, USA [2]College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China [3]MIIT Key Laboratory of Pattern Analysis & Machine Intelligence [4]JD Finance America Corporation. Correspondence to: Heng Huang <heng.huang@pitt.edu>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. 

## 1. Introduction

Alternating direction method of multipliers (ADMM) (Gabay & Mercier, 1976; Boyd et al., 2011) is a powerful optimization tool for the composite or constrained problems in machine learning. In general, it considers the following optimization problem:

$$\min_{x,y} f(x) + g(y), \quad \text{s.t. } Ax + By = c,$$

where $f(x) : \mathbb{R}^d \to \mathbb{R}$ and $g(y) : \mathbb{R}^p \to \mathbb{R}$ are convex functions. For example, in machine learning, $f(x)$ can be used for the empirical loss, $g(y)$ for the structure regularizer, and the constraint for encoding the structure pattern of model parameters. Due to the flexibility in splitting the objective function into loss $f(x)$ and regularizer $g(y)$, the ADMM can relatively easily solve some complicated structure problems in machine learning, such as the graph-guided fused lasso (Kim et al., 2009) and the overlapping group lasso, which are too complicated for the other popular optimization methods such as proximal gradient methods (Nesterov, 2005; Beck & Teboulle, 2009). Thus, the ADMM has been extensively studied in recent years (Boyd et al., 2011; Nishihara et al., 2015; Xu et al., 2017).

The above deterministic ADMM generally needs to compute the gradients of empirical loss function on all examples at each iteration, which makes it unsuitable for solving big data problems. Thus, the online and stochastic versions of ADMM (Wang & Banerjee, 2012; Suzuki, 2013; Ouyang et al., 2013) are developed. However, due to large variance of stochastic gradients, these stochastic methods suffer from a slow convergence rate. Recently, some fast stochastic ADMM methods (Zhong & Kwok, 2014; Suzuki, 2014; Zheng & Kwok, 2016a) have been proposed by using the variance reduced (VR) techniques.

So far, the above discussed ADMM methods build on the convexity of objective functions. In fact, ADMM is also highly successful in solving various nonconvex problems such as tensor decomposition (Kolda & Bader, 2009) and training neural networks (Taylor et al., 2016). Thus, some works (Li & Pong, 2015; Wang et al., 2015a;b; Hong et al., 2016; Jiang et al., 2019) have devoted to studying the nonconvex ADMM methods. More recently, for solving the big data problems, the nonconvex stochastic ADMMs (Huang et al., 2016; Zheng & Kwok, 2016b) have been proposed

*Table 1.* IFO complexity comparison of the non-convex ADMM methods for finding an $\epsilon$-approximate solution of the problem (1), i.e., $\mathbb{E}\|\nabla\mathcal{L}(x, y_{[m]}, z)\|^2 \leq \epsilon$. $n$ denotes the sample size.

| Problem | Algorithm | Reference | IFO |
|---|---|---|---|
| Finite-sum | ADMM | Jiang et al. (2019) | $\mathcal{O}(n\epsilon^{-1})$ |
| | SVRG-ADMM | Huang et al. (2016); Zheng & Kwok (2016b) | $\mathcal{O}(n + n^{\frac{2}{3}}\epsilon^{-1})$ |
| | SAGA-ADMM | Huang et al. (2016) | $\mathcal{O}(n + n^{\frac{2}{3}}\epsilon^{-1})$ |
| | SPIDER-ADMM | Ours | $\mathcal{O}(n + n^{\frac{1}{2}}\epsilon^{-1})$ |
| Online | SADMM | Huang & Chen (2018) | $\mathcal{O}(\epsilon^{-2})$ |
| | Online SPIDER-ADMM | Ours | $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ |

with the VR techniques such as the SVRG (Johnson & Zhang, 2013) and the SAGA (Defazio et al., 2014). In addition, Huang & Chen (2018) have extended the online/stochastic ADMM (Ouyang et al., 2013) to the nonconvex setting.

Although these works have studied the convergence of nonconvex stochastic ADMMs and proved these methods have $\mathcal{O}(\frac{c}{T})$ convergence rate, where $T$ denotes number of iteration and $c$ a constant independent on $T$, they have not provided the **optimal** incremental/stochastic first-order oracle (IFO/SFO (Ghadimi & Lan, 2013)) complexity for these methods yet. In other words, they have only proved these stochastic ADMMs have the same convergence rate to the deterministic ADMM (Jiang et al., 2019), but don't tell us whether these stochastic ADMMs have less IFO complexity than the deterministic ADMM, which is a key assessment criteria of the first-order stochastic methods (Reddi et al., 2016). For example, from the existing nonconvex SAGA-ADMM and SVRG-ADMM (Zheng & Kwok, 2016b; Huang et al., 2016), we only obtain a **rough** IFO complexity of $\mathcal{O}(n + bc\epsilon^{-1})$ for finding an $\epsilon$-approximate stationary point, where $b$ denotes the mini-batch size. In their convergence analysis, to ensure the convergence of these methods, they need to choose a small step size $\eta$ and a large penalty parameter $\rho$. Under this case, we maybe have $bc \geq n$, so that these stochastic ADMMs have no less IFO complexity than the deterministic ADMM. Thus, there still exist two important problems to be addressed:

- *Does the stochastic ADMM have less IFO complexity than the deterministic ADMM for nonconvex optimization?*
- *If the stochastic ADMM improves IFO complexity, how much can it improve?*

In the paper, we answer the above challenging questions with positive solutions and propose a new faster stochastic ADMM method (*i.e.,* SPIDER-ADMM) to solve the following nonconvex nonsmooth problem:

$$\min_{x,\{y_j\}_{j=1}^m} f(x) := \begin{cases} \frac{1}{n}\sum_{i=1}^n f_i(x) \text{ (finite-sum)} \\ \mathbb{E}_\zeta[f(x,\zeta)] \text{ (online)} \end{cases} + \sum_{j=1}^m g_j(y_j)$$
$$\text{s.t. } Ax + \sum_{j=1}^m B_j y_j = c, \tag{1}$$

where $A \in \mathbb{R}^{l\times d}$, $B_j \in \mathbb{R}^{l\times p}$ for all $j \in [m]$, $f(x) : \mathbb{R}^d \to \mathbb{R}$ is a *nonconvex* and smooth function, and $g_j(y_j) : \mathbb{R}^p \to \mathbb{R}$ is a convex and possibly *nonsmooth* function for all $j \in [m]$, $m \geq 1$. In machine learning, $f(x)$ can be used for losses such as activation functions of neural networks, $\sum_{j=1}^m g_j(y_j)$ can be used for not only single structure penalty (*e.g.,* sparse, low rank) but also superposition structures penalties (*e.g.*, sparse + low rank, sparse + group sparse), which are widely applied in robust PCA (Candès et al., 2011), subspace clustering (Liu et al., 2010), and dirty models (Jalali et al., 2010). For the problem (1), its finite-sum subproblem generally arises from the empirical loss minimization and M-estimation. While its online subproblem comes from the expected loss minimization. To address the online subproblem, we extend the SPIDER-ADMM to the online setting, and propose an online SPIDER-ADMM.

### 1.1. Challenges and Contributions

Our SPIDER-ADMM methods use a new stochastic path-integrated differential estimator (SPIDER), which was recently presented in (Fang et al., 2018) and was further improved by SpiderBoost in (Wang et al., 2018). Although the SPIDER/SpiderBoost have shown good performances in the stochastic gradient descent (SGD) and proximal SGD methods, applying these techniques to the nonconvex ADMM method *is not a trivial task*. There exist the following 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., 2015a). With using the inexact stochastic gradient, this difficulty is greater in the nonconvex stochastic ADMM methods;
- To obtain the optimal IFO complexity of our methods, we need to design a new effective *Lyapunov* function, which can not follow the existing nonconvex stochastic ADMM methods (Huang et al., 2016).

In this paper, thus, we will fill this gap between the nonconvex ADMM and the SPIDER/SpiderBoost methods. Our main **contributions** are summarized as follows:

1) We propose a faster stochastic ADMM ( *i.e.,* SPIDER-ADMM ) method for nonconvex optimization based on the SPIDER/SpiderBoost. Moreover, we prove that the SPIDER-ADMM achieves an optimal IFO complexity

of $\mathcal{O}(n+n^{1/2}\epsilon^{-1})$ for finding an $\epsilon$-approximate solution of nonconvex optimization, which improves the deterministic ADMM by a factor $\mathcal{O}(n^{1/2})$.

2) We extend the SPIDER-ADMM method to the online setting, and propose a faster online SPIDER-ADMM for nonconvex optimization. Moreover, we prove that the online SPIDER-ADMM achieves the optimal IFO complexity of $\mathcal{O}(\epsilon^{-\frac{3}{2}})$, which improves the existing best results by a factor of $\mathcal{O}(\epsilon^{\frac{1}{2}})$.

3) We provide an useful theoretical analysis framework for nonconvex stochastic ADMM methods with providing the optimal IFO complexity. Based on our new analysis framework, we also prove that the existing nonconvex SVRG-ADMM and SAGA-ADMM have the optimal IFO complexity of $\mathcal{O}(n+n^{2/3}\epsilon^{-1})$. Thus, our SPIDER-ADMM improves the existing stochastic ADMMs by a factor of $\mathcal{O}(n^{1/6})$.

### 1.2. Notations

Let $y_{[m]} = \{y_1, \cdots, y_m\}$ and $y_{[j:m]} = \{y_j, \cdots, y_m\}$ for $j \in [m] = \{1, 2, \cdots, m\}$. 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 matrix $G$, respectively; $\kappa_G = \frac{\sigma_{\max}(G)}{\sigma_{\min}(G)} \geq 1$. $\sigma^A_{\max}$ and $\sigma^A_{\min}$ denote the largest and smallest eigenvalues of matrix $A^T A$, respectively. Given positive definite matrices $\{H_j\}_{j=1}^m$, let $\sigma^H_{\min} = \min_j \sigma_{\min}(H_j)$ and $\sigma^H_{\max} = \max_j \sigma_{\max}(H_j)$. $I_d$ denotes a $d \times d$ identity matrix.

## 2. Preliminaries

In the section, we introduce some preliminaries regarding problem (1). First, we restate the standard $\epsilon$-approximate stationary point of the nonconvex problem (1) used in (Jiang et al., 2019; Zheng & Kwok, 2016b).

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

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

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

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

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

Next, we give some standard assumptions regarding problem (1) as follows:

**Assumption 1.** *Each loss function $f_i(x)$ is $L$-smooth 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 loss function $f_i(x)$ is bounded, i.e., there exists a constant $\delta > 0$ such that for all $x$, it follows $\|\nabla f_i(x)\|^2 \leq \delta^2$.*

**Assumption 3.** *$f(x)$ and $g_j(y_j)$ for all $j \in [m]$ are all lower bounded, and let $f^* = \inf_x f(x) > -\infty$ and $g_j^* = \inf_{y_j} g_j(y_j) > -\infty$.*

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

Assumption 1 imposes smoothness on the individual loss functions, which is commonly used in convergence analysis of the nonconvex algorithms (Ghadimi & Lan, 2013; Ghadimi et al., 2016). Assumption 2 shows the gradients of loss functions have a bounded norm, which is used in the stochastic gradient-based and ADMM-type methods (Boyd et al., 2011; Suzuki, 2013; Hazan et al., 2016). Assumptions 3 and 4 have been used in the study of nonconvex ADMMs (Hong et al., 2016; Jiang et al., 2019; Zheng & Kwok, 2016b). Assumptions 3 guarantees the feasibility of the problem (1). Assumption 4 guarantees the matrix $A^T A$ or $AA^T$ is non-singular. Since there exist multiple regularizers in the above problem (1), $A$ is general a full column rank matrix. Without loss of generality, we will use the full column rank matrix $A$ below.

## 3. Fast SPIDER-ADMM Method

In the section, we propose a new faster stochastic ADMM algorithm, *i.e.,* SPIDER-ADMM, to solve the finite-sum problem (1). We begin with giving the augmented Lagrangian function of the problem (1):

$$\begin{aligned}\mathcal{L}_\rho(x, y_{[m]}, z) =& f(x) + \sum_{j=1}^m g_j(y_j) - \langle z, Ax + \sum_{j=1}^m B_j y_j - c\rangle \\ &+ \frac{\rho}{2}\|Ax + \sum_{j=1}^m B_j y_j - c\|^2, \end{aligned} \tag{3}$$

where $z \in \mathbb{R}^l$ and $\rho > 0$ denote the dual variable and penalty parameter, respectively. Due to using stochastic gradient of the function $f(x)$ to update $x$, we define an approximated function over $x_k$ as follows:

$$\begin{aligned}\hat{\mathcal{L}}_\rho(x, y^{k+1}_{[m]}, z_k, v_k) =& f(x_k) + v_k^T(x - x_k) + \frac{1}{2\eta}\|x - x_k\|_G^2 \\ &+ \sum_{j=1}^m g_j(y_j^{k+1}) - z_k^T(Ax + \sum_{j=1}^m B_j y_j^{k+1} - c) \\ &+ \frac{\rho}{2}\|Ax + \sum_{j=1}^m B_j y_j^{k+1} - c\|^2, \end{aligned} \tag{4}$$

where $\eta > 0$ is a step size; $v_k$ is an unbiased stochastic gradient over $x_k$, *i.e.,* $\mathbb{E}[v_k] = \nabla f(x_k)$; $G \succ 0$ is a positive

matrix. In updating $x$, to avoid computing inverse of $\frac{G}{\eta}+A^TA$, we can set $G = rI_d - \rho\eta A^TA \succeq I_d$ with $r \geq \rho\eta\sigma^A_{\max}+1$ to linearize term $\frac{\rho}{2}\|Ax+\sum_{j=1}^m B_jy_j^{k+1}-c\|^2$. To use the following proximal operator to update $y_j$:

$$y_j^{k+1} = \arg\min_{y_j\in\mathbb{R}^p} \frac{1}{2}\|y_j - y_j^k\|^2 + g_j(y_j),\ \forall j\in[m] \quad (5)$$

we can set $H_j = \tau_j I_p - \rho B_j^TB_j \succeq I_p$ with $\tau_j \geq \rho\sigma_{\max}(B_j^TB_j)+1$ for all $j\in[m]$ to linearize term $\frac{\rho}{2}\|Ax_k+\sum_{i=1}^{j-1}B_iy_i^{k+1}+B_jy_j+\sum_{i=j+1}^m B_iy_i^k-c\|^2$.

Algorithm 1 gives the SPIDER-ADMM algorithmic framework. In Algorithm 1, after setting $v_0=\nabla f(x_0)$, for each subsequent iteration $k$, we have:

$$v_k = \nabla f_{\mathcal{I}_k}(x_k) - \nabla f_{\mathcal{I}_k}(x_{k-1}) + v_{k-1}, \quad (6)$$

where $\nabla f_{\mathcal{I}_k}(x_k) = \frac{1}{|\mathcal{I}_k|}\sum_{i\in\mathcal{I}_k}\nabla f_i(x_t)$. It is easy to check $\mathbb{E}[v_k|x_0]=\nabla f(x_k)$, *i.e.,* an unbiased estimate gradient over $x_k$. Comparing the existing SVRG-ADMM, our SPIDER-ADMM constructs stochastic gradient $v_k$ based on the information $x_{k-1}$ and $v_{k-1}$, while the SVRG-ADMM constructs $v_k$ based on the information $x_0$ and $v_0$ (*i.e.,* the initalization information of each outer loop). Due to using more fresh information, thus, SPIDER-ADMM can yield more accurate estimation of the full gradient than SVRG-ADMM. Simultaneously, it does not require to additional computation and memory, so it costs less memory than the existing SAGA-ADMM.

**Algorithm 1** SPIDER-ADMM Algorithm

**Input:** $b$, $q$, $K$, $\eta>0$ and $\rho>0$;
**Initialize:** $x_0\in\mathbb{R}^d$, $y_j^0\in\mathbb{R}^p$, $j\in[m]$ and $z_0\in\mathbb{R}^l$;
**for** $k=0,1,\cdots,K-1$ **do**
  **if** $\mathrm{mod}(k,q)=0$ **then**
    Compute $v_k=\nabla f(x_k)$;
  **else**
    Uniformly randomly pick a mini-batch $\mathcal{I}_k$ (with replacement) from $\{1,2,\cdots,n\}$ with $|\mathcal{I}_k|=b$, and compute

$$v_k = \nabla f_{\mathcal{I}_k}(x_k) - \nabla f_{\mathcal{I}_k}(x_{k-1}) + v_{k-1};$$

  **end if**
  $y_j^{k+1} = \arg\min_{y_j}\{\mathcal{L}_\rho(x_k, y_{[j-1]}^{k+1}, y_j, y_{[j+1:m]}^k, z_k) + \frac{1}{2}\|y_j-y_j^k\|^2_{H_j}\}$ for all $j\in[m]$;
  $x_{k+1} = \arg\min_x \hat{\mathcal{L}}_\rho\big(x, y_{[m]}^{k+1}, z_k, v_k\big)$;
  $z_{k+1} = z_k - \rho(Ax_{k+1}+\sum_{j=1}^m B_jy_j^{k+1}-c)$;
**end for**
**Output:** $\{x, y_{[m]}, z\}$ chosen uniformly random from $\{x_k, y_{[m]}^k, z_k\}_{k=1}^K$.

# 4. Fast Online SPIDER-ADMM Method

In the section, we propose an online SPIDER-ADMM to solve the online problem (1), which is equivalent to the following stochastic constrained problem:

$$\min \mathbb{E}_\zeta[f(x,\zeta)] + \sum_{j=1}^m g_j(y_j),\ \text{s.t. } Ax+\sum_{j=1}^m B_jy_j = c, \quad (7)$$

where $f(x)=\mathbb{E}_\zeta[f(x,\zeta)]$ denotes a population risk over an underlying data distribution. The problem (7) can be viewed as having infinite samples, so we cannot evaluate the full gradient $\nabla f(x)$. For the problem (7), we use stochastic sampling to evaluate the full gradient. Algorithm 2 shows the algorithmic framework of online SPIDER-ADMM method. In Algorithm 2, we use the mini-batch samples to estimate the full gradient.

**Algorithm 2** Online SPIDER-ADMM Algorithm

**Input:** $b_1$, $b_2$, $q$, $K$, $\eta>0$ and $\rho>0$;
**Initialize:** $x_0\in\mathbb{R}^d$, $y_j^0\in\mathbb{R}^p$, $j\in[m]$ and $z_0\in\mathbb{R}^l$;
**for** $k=0,1,\cdots,K-1$ **do**
  **if** $\mathrm{mod}(k,q)=0$ **then**
    Draw $S_1$ samples with $|S_1|=b_1$, and compute $v_k=\frac{1}{b_1}\sum_{i\in S_1}\nabla f_i(x_k)$;
  **else**
    Draw $S_2$ samples with $|S_2|=b_2=\sqrt{b_1}$, and compute

$$v_k = \frac{1}{b_2}\sum_{i\in S_2}\big(\nabla f_i(x_k) - f_i(x_{k-1})\big) + v_{k-1};$$

  **end if**
  $y_j^{k+1} = \arg\min_{y_j}\{\mathcal{L}_\rho(x_k, y_{[j-1]}^{k+1}, y_j, y_{[j+1:m]}^k, z_k) + \frac{1}{2}\|y_j-y_j^k\|^2_{H_j}\}$ for all $j\in[m]$;
  $x_{k+1} = \arg\min_x \hat{\mathcal{L}}_\rho\big(x, y_{[m]}^{k+1}, z_k, v_k\big)$;
  $z_{k+1} = z_k - \rho(Ax_{k+1}+\sum_{j=1}^m B_jy_j^{k+1}-c)$;
**end for**
**Output:** $\{x, y_{[m]}, z\}$ chosen uniformly random from $\{x_k, y_{[m]}^k, z_k\}_{k=1}^K$.

# 5. Convergence Analysis

In the section, we study the convergence properties of both the SPIDER-ADMM and online SPIDER-ADMM. At the same time, based on our new theoretical analysis framework, we afresh analyze the convergence properties of existing ADMM-based nonconvex optimization algorithms, *i.e.,* SVRG-ADMM and SAGA-ADMM, and derive their optimal IFO complexity for finding an $\epsilon$-approximate solution.

### 5.1. Convergence Analysis of SPIDER-ADMM

In the subsection, we study convergence properties of the SPIDER-ADMM algorithm. Throughout the paper, let $n_k = \lceil k/q \rceil$ such that $(n_k - 1)q \leq k \leq n_k q - 1$.

**Lemma 1.** *Suppose the sequence $\{x_k, y^k_{[m]}, z_k\}_{k=1}^K$ is generated from Algorithm 1, and define a* Lyapunov *function $R_k$ as follows:*

$$R_k = \mathcal{L}_\rho(x_k, y^k_{[m]}, z_k) + (\frac{9L^2}{\sigma^A_{\min}\rho} + \frac{3\sigma^2_{\max}(G)}{\sigma^A_{\min}\eta^2\rho})\|x_k - x_{k-1}\|^2 + \frac{2L^2}{\sigma^A_{\min}\rho b}\sum_{i=(n_k-1)q}^{k-1}\mathbb{E}\|x_{i+1} - x_i\|^2.$$

*Let $b = q$, $\eta = \frac{2\alpha\sigma_{\min}(G)}{3L}$ $(0 < \alpha \leq 1)$ and $\rho = \frac{\sqrt{170}\kappa_G L}{\sigma^A_{\min}\alpha}$, then we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}(\|x_{k+1} - x_k\|^2 + \sum_{j=1}^m \|y^k_j - y^{k+1}_j\|^2) \leq \frac{R_0 - R^*}{K\gamma},$$

*where $\gamma = \min(\chi, \sigma^H_{\min})$ with $\chi \geq \frac{\sqrt{170}\kappa_G L}{4\alpha}$ and $R^*$ is a lower bound of the function $R_k$.*

Let $\theta_k = \mathbb{E}[\|x_{k+1} - x_k\|^2 + \|x_k - x_{k-1}\|^2 + \frac{1}{q}\sum_{i=(n_k-1)q}^k \|x_{i+1} - x_i\|^2 + \sum_{j=1}^m \|y^k_j - y^{k+1}_j\|^2]$. Next, based on the above lemma, we give the convergence properties of SPIDER-ADMM.

**Theorem 1.** *Suppose the sequence $\{x_k, y^k_{[m]}, z_k)_{k=1}^K$ is generated from Algorithm 1. Let*

$$\nu_1 = m\big(\rho^2\sigma^B_{\max}\sigma^A_{\max} + \rho^2(\sigma^B_{\max})^2 + \sigma^2_{\max}(H)\big),$$
$$\nu_2 = 3(L^2 + \frac{\sigma^2_{\max}(G)}{\eta^2}), \nu_3 = \frac{18L^2}{\sigma^A_{\min}\rho^2} + \frac{3\sigma^2_{\max}(G)}{\sigma^A_{\min}\eta^2\rho^2},$$

*and let $b = q$, $\eta = \frac{2\alpha\sigma_{\min}(G)}{3L}$ $(0 < \alpha \leq 1)$, and $\rho = \frac{\sqrt{170}\kappa_G L}{\sigma^A_{\min}\alpha}$, then we have*

$$\min_{1\leq k\leq K}\mathbb{E}\big[dist(0, \partial L(x_k, y^k_{[m]}, z_k))^2\big] \leq \frac{\nu_{\max}}{K}\sum_{k=1}^{K-1}\theta_k \leq \frac{3\nu_{\max}(R_0 - R^*)}{K\gamma},$$

*where $\gamma = \min(\chi, \sigma^H_{\min})$ with $\chi \geq \frac{\sqrt{170}\kappa_G L}{4\alpha}$, $\nu_{\max} = \max\{\nu_1, \nu_2, \nu_3\}$ and $R^*$ is a lower bound of the function $R_k$. It implies that the iteration number $K$ satisfies*

$$K = \frac{3\nu_{\max}(R_0 - R^*)}{\epsilon\gamma},$$

*then $(x_{k^*}, y^{k^*}_{[m]}, z_{k^*})$ is an $\epsilon$-approximate stationary point of (1), where $k^* = \arg\min_k \theta_k$.*

**Remark 1.** *Theorem 1 shows that the SPIDER-ADMM has $O(1/K)$ convergence rate. Moreover, given $b = q = \sqrt{n}$, $\eta = \frac{2\alpha\sigma_{\min}(G)}{3L}$ $(0 < \alpha \leq 1)$ and $\rho = \frac{\sqrt{170}\kappa_G L}{\sigma^A_{\min}\alpha}$, the SPIDER-ADMM has the optimal IFO of $\mathcal{O}(n + n^{\frac{1}{2}}\epsilon^{-1})$ for finding an $\epsilon$-approximate solution. In particular, we can choose $\alpha \in (0, 1]$ according to different problems to obtain appropriate step-size $\eta$ and penalty parameter $\rho$,* e.g.*, set $\alpha = 1$, we have $\eta = \frac{2\sigma_{\min}(G)}{3L}$ and $\rho = \frac{\sqrt{170}\kappa_G L}{\sigma^A_{\min}}$.*

### 5.2. Convergence Analysis of Online SPIDER-ADMM

In the subsection, we study convergence properties of the online SPIDER-ADMM algorithm.

**Lemma 2.** *Suppose the sequence $\{x_k, y^k_{[m]}, z_k\}_{k=1}^K$ is generated from Algorithm 2, and define a* Lyapunov *function $\Phi_k$ as follows:*

$$\Phi_k = \mathcal{L}_\rho(x_k, y^k_{[m]}, z_k) + (\frac{9L^2}{\sigma^A_{\min}\rho} + \frac{3\sigma^2_{\max}(G)}{\sigma^A_{\min}\eta^2\rho})\|x_k - x_{k-1}\|^2 + \frac{2L^2}{\sigma^A_{\min}\rho b_2}\sum_{i=(n_k-1)q}^{k-1}\mathbb{E}\|x_{i+1} - x_i\|^2.$$

*Let $b_2 = q$, $\eta = \frac{2\alpha\sigma_{\min}(G)}{3L}$ $(0 < \alpha \leq 1)$ and $\rho = \frac{\sqrt{170}\kappa_G L}{\sigma^A_{\min}\alpha}$, then we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}(\|x_{k+1} - x_k\|^2 + \sum_{j=1}^m \|y^k_j - y^{k+1}_j\|^2) \leq \frac{\Phi_0 - \Phi^*}{K\gamma} + \frac{2\delta^2}{b_1 L\gamma} + \frac{72\delta^2}{\sigma^A_{\min}b_1\rho\gamma},$$

*where $\gamma = \min(\chi, \sigma^H_{\min})$ with $\chi \geq \frac{\sqrt{170}\kappa_G L}{4\alpha}$ and $\Phi^*$ is a lower bound of the function $\Phi_k$.*

Let $\theta_k = \mathbb{E}[\|x_{k+1} - x_k\|^2 + \|x_k - x_{k-1}\|^2 + \frac{1}{q}\sum_{i=(n_k-1)q}^k \|x_{i+1} - x_i\|^2 + \sum_{j=1}^m \|y^k_j - y^{k+1}_j\|^2]$.

**Theorem 2.** *Suppose the sequence $\{x_k, y^k_{[m]}, z_k)_{k=1}^K$ is generated from Algorithm 2. Let*

$$\nu_1 = m\big(\rho^2\sigma^B_{\max}\sigma^A_{\max} + \rho^2(\sigma^B_{\max})^2 + \sigma^2_{\max}(H)\big),$$
$$\nu_2 = 3(L^2 + \frac{\sigma^2_{\max}(G)}{\eta^2}), \nu_3 = \frac{18L^2}{\sigma^A_{\min}\rho^2} + \frac{3\sigma^2_{\max}(G)}{\sigma^A_{\min}\eta^2\rho^2},$$

*and let $b_2 = q = \sqrt{b_1}$, $\eta = \frac{2\alpha\sigma_{\min}(G)}{3L}$ $(0 < \alpha \leq 1)$ and $\rho = \frac{\sqrt{170}\kappa_G L}{\sigma^A_{\min}\alpha}$, then we have*

$$\min_{1\leq k\leq K}\mathbb{E}\big[dist(0, \partial L(x_k, y^k_{[m]}, z_k))^2\big] \leq \frac{\nu_{\max}}{K}\sum_{k=1}^{K-1}\theta_k + \frac{w}{b_1} \leq \frac{3\nu_{\max}(\Phi_0 - \Phi^*)}{K\gamma} + \frac{6\nu_{\max}\delta^2}{b_1\gamma}(\frac{1}{L} + \frac{36}{\sigma^A_{\min}\rho}) + \frac{w}{b_1},$$

*where* $w = 12\delta^2 \max\{1, \frac{6}{\sigma^A_{\min}\rho^2}\}$, $\gamma = \min(\chi, \sigma^H_{\min})$ *with* $\chi \geq \frac{\sqrt{170}\kappa_G L}{4\alpha}$, $\nu_{\max} = \max\{\nu_1, \nu_2, \nu_3\}$ *and* $\Phi^*$ *is a lower bound of the function* $\Phi_k$. *It implies that* $K$ *and* $b_1$ *satisfy*

$$K = \frac{6\nu_{\max}(\Phi_0 - \Phi^*)}{\epsilon\gamma},\ b_1 = \frac{12\nu_{\max}\delta^2}{\epsilon\gamma}(\frac{1}{L} + \frac{36}{\sigma^A_{\min}\rho}) + \frac{2w}{\epsilon},$$

*then* $(x_{k^*}, y^{k^*}_{[m]}, z_{k^*})$ *is an* $\epsilon$*-approximate stationary point of* (1), *where* $k^* = \arg\min_k \theta_k$.

**Remark 2.** *Theorem 2 shows that given* $b_2 = q = \sqrt{b_1}$, $\eta = \frac{2\alpha\sigma_{\min}(G)}{3L}$ $(0 < \alpha \leq 1)$, $\rho = \frac{\sqrt{170}\kappa_G L}{\sigma^A_{\min}\alpha}$ *and* $b_1 = \mathcal{O}(\epsilon^{-1})$, *the online SPIDER-ADMM has the optimal IFO of* $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ *for finding an* $\epsilon$*-approximate solution.*

### 5.3. Convergence Analysis of Non-convex SVRG-ADMM

In the subsection, we extend the existing nonconvex SVRG-ADMM method (Huang et al., 2016; Zheng & Kwok, 2016b) to the multiple variables setting for solving the problem (1). The SVRG-ADMM algorithm is described in Algorithm 3 given in the supplementary document. Next, we analyze convergence properties of the SVRG-ADMM algorithm, and derive its optimal IFO complexity.

**Lemma 3.** *Suppose the sequence* $\{(x^s_t, y^{s,t}_{[m]}, z^s_t)^M_{t=1}\}^S_{s=1}$ *is generated from Algorithm 3, and define a* Lyapunov *function:*

$$\Gamma^s_t = \mathbb{E}\big[\mathcal{L}_\rho(x^s_t, y^{s,t}_{[m]}, z^s_t) + (\frac{3\sigma^2_{\max}(G)}{\sigma^A_{\min}\eta^2\rho} + \frac{9L^2}{\sigma^A_{\min}\rho})\|x^s_t - x^s_{t-1}\|^2 + \frac{9L^2}{\sigma^A_{\min}\rho b}\|x^s_{t-1} - \tilde{x}^s\|^2 + c_t\|x^s_t - \tilde{x}^s\|^2\big],$$

*where the positive sequence* $\{c_t\}$ *satisfies, for* $s = 1, 2, \cdots, S$

$$c_t = \begin{cases} \frac{18L^2}{\sigma^A_{\min}\rho b} + \frac{L}{b} + (1+\beta)c_{t+1},\ 1 \leq t \leq M, \\ 0,\ t \geq M+1. \end{cases}$$

*Let* $M = n^{\frac{1}{3}}$, $b = n^{\frac{2}{3}}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{5L}$ $(0 < \alpha \leq 1)$ *and* $\rho = \frac{2\sqrt{231}\kappa_G L}{\sigma^A_{\min}\alpha}$, *we have*

$$\frac{1}{T}\sum_{s=1}^{S}\sum_{t=0}^{M-1}(\sigma^H_{\min}\sum_{j=1}^{m}\|y^{s,t}_j - y^{s,t+1}_j\|^2 + \chi_t\|x^s_{t+1} - x^s_t\|^2 + \frac{L}{2b}\|x^s_t - \tilde{x}^s\|^2) \leq \frac{\Gamma^1_0 - \Gamma^*}{T}. \quad (8)$$

*where* $T = MS$, $\chi_t \geq \frac{\sqrt{231}\kappa_G L}{2\alpha} > 0$ *and* $\Gamma^*$ *denotes a lower bound of function* $\Gamma^s_t$.

Let $\theta^s_t = \mathbb{E}[\|x^s_{t+1} - x^s_t\|^2 + \|x^s_t - x^s_{t-1}\|^2 + \frac{1}{b}(\|x^s_t - \tilde{x}^s\|^2 + \|x^s_{t-1} - \tilde{x}^s\|^2) + \sum_{j=1}^m \|y^{s,t}_j - y^{s,t+1}_j\|^2]$.

**Theorem 3.** *Suppose the sequence* $\{(x^s_t, y^{s,t}_{[m]}, z^s_t)^M_{t=1}\}^S_{s=1}$ *is generated from Algorithm 3. Let*

$$\nu_1 = m\big(\rho^2\sigma^B_{\max}\sigma^A_{\max} + \rho^2(\sigma^B_{\max})^2 + \sigma^2_{\max}(H)\big),$$
$$\nu_2 = 3L^2 + \frac{3\sigma^2_{\max}(G)}{\eta^2},\ \nu_3 = \frac{9L^2}{\sigma^A_{\min}\rho^2} + \frac{3\sigma^2_{\max}(G)}{\sigma^A_{\min}\eta^2\rho^2},$$

*and given* $M = n^{\frac{1}{3}}$, $b = n^{\frac{2}{3}}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{5L}$ $(0 < \alpha \leq 1)$ *and* $\rho = \frac{2\sqrt{231}\kappa_G L}{\sigma^A_{\min}\alpha}$, *then we have*

$$\min_{s,t}\mathbb{E}\big[dist(0, \partial L(x^s_t, y^{s,t}_{[m]}, z^s_t))^2\big] \leq \frac{2\nu_{\max}(\Gamma^1_0 - \Gamma^*)}{\gamma T},$$

*where* $\gamma = \min(\sigma^H_{\min}, \frac{L}{2}, \chi_t)$, $\nu_{\max} = \max(\nu_1, \nu_2, \nu_3)$ *and* $\Gamma^*$ *is a lower bound of function* $\Gamma^s_t$. *It implies that the whole iteration number* $T = MS$ *satisfies*

$$T = \frac{2\nu_{\max}(\Gamma^1_0 - \Gamma^*)}{\epsilon\gamma},$$

*then* $(x^{s^*}_{t^*}, y^{s^*,t^*}_{[m]}, z^{s^*}_{t^*})$ *is an* $\epsilon$*-stationary point of* (1), *where* $(t^*, s^*) = \arg\min_{t,s}\theta^s_t$.

**Remark 3.** *Theorem 3 shows that given* $M = n^{\frac{1}{3}}$, $b = n^{\frac{2}{3}}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{5L}$ $(0 < \alpha \leq 1)$ *and* $\rho = \frac{2\sqrt{231}\kappa_G L}{\sigma^A_{\min}\alpha}$, *the nonconvex SVRG-ADMM has the optimal IFO complexity of* $\mathcal{O}(n + n^{\frac{2}{3}}\epsilon^{-1})$ *for finding an* $\epsilon$*-approximate solution.*

### 5.4. Convergence Analysis of Non-convex SAGA-ADMM

In the subsection, we extend the existing nonconvex SAGA-ADMM method (Huang et al., 2016) to the multiple variables setting for solving the problem (1). The SAGA-ADMM algorithm is described in Algorithm 4 given in the supplementary document. Next, we analyze convergence properties of non-convex SAGA-ADMM, and derive its the optimal IFO complexity.

**Lemma 4.** *Suppose the sequence* $\{x_t, y^t_{[m]}, z_t\}^T_{t=1}$ *is generated from Algorithm 4, and define a* Lyapunov *function*

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

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

$$c_t = \begin{cases} \frac{18L^2}{\sigma^A_{\min}\rho b} + \frac{L}{b} + (1-p)(1+\beta)c_{t+1},\ 0 \leq t \leq T-1, \\ 0,\ t \geq T, \end{cases}$$

*where* $p$ *denotes probability of an index* $i$ *being in* $\mathcal{I}_t$. *Further, let* $b = n^{\frac{2}{3}}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{17L}$ $(0 < \alpha \leq 1)$ *and*

$\rho = \frac{2\sqrt{2031}\kappa_G}{\sigma^A_{\min}\alpha}$ *we have*

$$\frac{1}{T}\sum_{t=1}^{T}(\sigma^H_{\min}\sum_{j=1}^{m}\|y_j^t - y_j^{t+1}\|^2 + \chi_t\|x_t - x_{t+1}\|^2 + \frac{L}{2b}\frac{1}{n}\sum_{i=1}^{n}\|x_t - u_i^t\|^2) \leq \frac{\Omega_0 - \Omega^*}{T},$$

*where* $\chi_t \geq \frac{\sqrt{2031}\kappa_G L}{2\alpha} > 0$ *and* $\Omega^*$ *denotes a lower bound of function* $\Omega_t$.

Let $\theta_t = \mathbb{E}[\|x_{t+1} - x_t\|^2 + \|x_t - x_{t-1}\|^2 + \frac{1}{bn}\sum_{i=1}^{n}(\|x_t - u_i^t\|^2 + \|x_{t-1} - u_i^{t-1}\|^2) + \sum_{j=1}^{m}\|y_j^t - y_j^{t+1}\|^2]$.

**Theorem 4.** *Suppose the sequence* $\{x_t, y^t_{[m]}, z_t\}_{t=1}^T$ *is generated from Algorithm 4. Let*

$$\nu_1 = m\big(\rho^2\sigma^B_{\max}\sigma^A_{\max} + \rho^2(\sigma^B_{\max})^2 + \sigma^2_{\max}(H)\big),$$
$$\nu_2 = 3L^2 + \frac{3\sigma^2_{\max}(G)}{\eta^2},\ \nu_3 = \frac{9L^2}{\sigma^A_{\min}\rho^2} + \frac{3\sigma^2_{\max}(G)}{\sigma^A_{\min}\eta^2\rho^2},$$

*and given* $b = n^{\frac{2}{3}}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{17L}$ $(0 < \alpha \leq 1)$ *and* $\rho = \frac{2\sqrt{2031}\kappa_G}{\sigma^A_{\min}\alpha}$, *then we have*

$$\min_{1\leq t\leq T}\mathbb{E}\big[dist(0, \partial L(x_t, y^t_{[m]}, z_t))^2\big] \leq \frac{2\nu_{\max}(\Omega_0 - \Omega^*)}{\gamma T},$$

*where* $\gamma = \min(\sigma^H_{\min}, \frac{L}{2}, \chi_t)$ *with* $\chi_t \geq \frac{\sqrt{2031}\kappa_G L}{2\alpha} > 0$, $\nu_{\max} = \max(\nu_1, \nu_2, \nu_3)$ *and* $\Omega^*$ *is a lower bound of function* $\Omega_t$. *It implies that the iteration number* $T$ *satisfies*

$$T = \frac{2\nu_{\max}}{\epsilon\gamma}(\Omega_0 - \Omega^*),$$

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

**Remark 4.** *Theorem 4 shows that given* $b = n^{\frac{2}{3}}$, $\eta = \frac{\alpha\sigma_{\min}(G)}{17L}$ $(0 < \alpha \leq 1)$ *and* $\rho = \frac{2\sqrt{2031}\kappa_G L}{\sigma^A_{\min}\alpha}$, *the non-convex SAGA-ADMM has the optimal IFO of* $\mathcal{O}(n + n^{\frac{2}{3}}\epsilon^{-1})$ *for finding an* $\epsilon$*-approximate solution.*

**Remark 5.** ***Our contributions*** *on convergence analysis of both the non-convex SVRG-ADMM and SAGA-ADMM are given as follows:*

- *We extend both the existing non-convex SVRG-ADMM and SAGA-ADMM to the multi-block setting for solving the problem* (1);
- *We not only give its optimal IFO complexity of* $\mathcal{O}(n + n^{\frac{2}{3}}\epsilon^{-1})$, *but also provide the* specific and simple *choice on the step-size* $\eta$ *and penalty parameter* $\rho$.

*All related proofs are provided in the supplementary document.*

*Table 2.* Real datasets

| datasets | $\#samples$ | $\#features$ | $\#classes$ |
|---|---|---|---|
| *a9a* | 32,561 | 123 | 2 |
| *w8a* | 64,700 | 300 | 2 |
| *ijcnn1* | 126,702 | 22 | 2 |
| *covtype.binary* | 581,012 | 54 | 2 |
| *letter* | 15,000 | 16 | 26 |
| *sensorless* | 58,509 | 48 | 11 |
| *mnist* | 60,000 | 780 | 10 |
| *covtype* | 581,012 | 54 | 7 |

## 6. Experiments

In this section, we will compare the proposed algorithm (SPIDER-ADMM) with the existing non-convex algorithms ( nc-ADMM (Jiang et al., 2019), nc-SVRG-ADMM (Huang et al., 2016; Zheng & Kwok, 2016b), nc-SAGA-ADMM (Huang et al., 2016) and nc-SADMM (Huang & Chen, 2018) ) on two applications: 1) Graph-guided binary classification; 2) Multi-task learning. In the experiment, we use some publicly available datasets[1], which are summarized in Table 2. All algorithms are implemented in MATLAB, and all experiments are performed on a PC with an Intel i7-4790 CPU and 16GB memory.

### 6.1. Graph-Guided Binary Classification

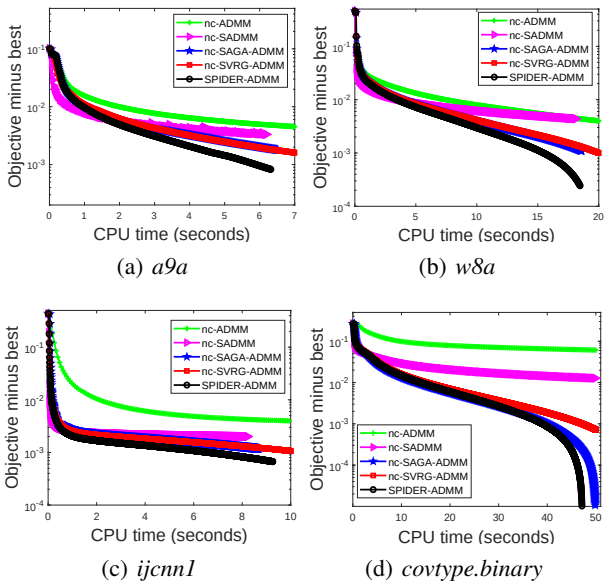
In the subsection, we focus on the binary classification task. Specifically, given a set of training samples $(a_i, b_i)_{i=1}^n$, where $a_i \in \mathbb{R}^d$, $b_i \in \{-1, 1\}$, then we solve the following nonconvex empirical loss minimization problem:

$$\min_{x\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^{n} f_i(x) + \lambda\|Ax\|_1, \tag{9}$$

where $f_i(x) = \frac{1}{1+\exp(b_i a_i^T x)}$ is the nonconvex sigmoid loss function. We use the nonsmooth regularizer *i.e.,* graph-guided fused lasso (Kim et al., 2009), and $A$ decodes the sparsity pattern of graph, which is obtained by sparse precision matrix estimation (Friedman et al., 2008). To solve the problem (9), we give an auxiliary variable $y$ with the constraint $y = Ax$. In the experiment, we fix the parameter $\lambda = 10^{-5}$, and use the same initial solution $x_0$ from the standard normal distribution for all algorithms.

Figure 1 shows that the objective values of our SPIDER-ADMM method faster decrease than those of other methods, as CPU time consumed increases. Thus, these results demonstrate that our method has a relatively faster convergence rate than other methods.

[1] These data are from the LIBSVM website (www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/).




(a) *a9a* (b) *w8a*

(c) *ijcnn1* (d) *covtype.binary*

*Figure 1.* Objective value *versus* CPU time of the *nonconvex* graph-guided binary classification model on some real datasets.

### 6.2. Multi-Task Learning

In this subsection, we focus on the multi-task learning task with sparse and low-rank structures. Specifically, given a set of training samples $(a_i, b_i)_{i=1}^n$, where $a_i \in \mathbb{R}^d$ and $b_i \in \{1, 2, \cdots, c\}$, then let $D \in \mathbb{R}^{n\times c}$ with $D_{ij} = 1$ if $j = b_i$, and $D_{ij} = 0$ otherwise. This multi-task learning is equivalent to solving the following nonconvex problem:

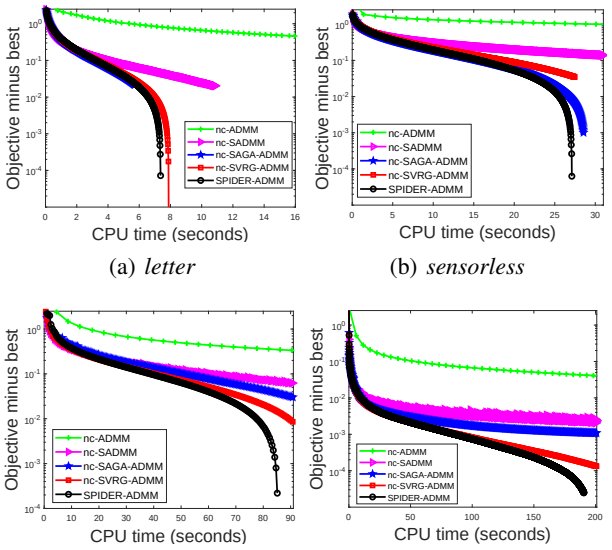
$$\min_{X\in\mathbb{R}^{c\times d}} \frac{1}{n}\sum_{i=1}^{n} f_i(X) + \lambda_1 \sum_{ij} \kappa(|X_{ij}|) + \lambda_2 \|X\|_*, \quad (10)$$

where $f_i(X) = \log(\sum_{j=1}^c \exp(X_{j,.}a_i)) - \sum_{j=1}^c D_{ij}X_{j,.}a_i$ is a multinomial logistic loss function, $\kappa(|X_{ij}|) = \beta\log(1+\frac{|X_{ij}|}{\alpha})$ is the nonconvex log-sum penalty function (Candes et al., 2008). Next, we change the above problem into the following form:

$$\begin{aligned} &\min \frac{1}{n}\sum_{i=1}^{n} \bar{f}_i(X) + \lambda_1\kappa_0\|Y_1\|_1 + \lambda_2\|Y_2\|_* \\ &\text{s.t. } AX + B_1Y_1 + B_2Y_2 = 0, \end{aligned} \quad (11)$$

where $\bar{f}_i(X) = f_i(X) + \lambda_1\big(\sum_{ij}\kappa(|X_{ij}|) - \kappa_0\|X\|_1\big)$, and $\kappa_0 = \kappa'(0)$. Here $A = [I_c; I_c] \in \mathbb{R}^{2c\times c}$, $B_1 = [-I_c; 0] \in \mathbb{R}^{2c\times c}$ and $B_2 = [0; -I]$. By the Proposition 2.3 in Yao & Kwok (2016), $\bar{f}_i(X)$ is nonconvex and smooth. In the experiment, we fix the parameters $\lambda_1 = 10^{-5}$ and $\lambda_2 = 10^{-4}$, and use the same initial solution $x_0$ from the standard normal distribution for all algorithms.

Figure 2 shows that objective values of our SPIDER-ADMM faster decrease than those of the other methods, as CPU time consumed increases. Similarly, these results also demonstrate that our method has a relatively faster convergence rate than other methods.




(a) *letter* (b) *sensorless*

(c) *mnist* (d) *covtype*

*Figure 2.* Objective value *versus* CPU time of the *nonconvex* multi-task learning on some real datasets.

## 7. Conclusion

In the paper, we propose a faster stochastic ADMM method (*i.e.,* SPIDER-ADMM) for nonconvex optimization. Moreover, we prove that the SPIDER-ADMM achieves a record-breaking IFO complexity of $\mathcal{O}(n + n^{1/2}\epsilon^{-1})$. Further, we extend the SPIDER-ADMM to the online setting, and propose a faster online ADMM method (*i.e.,* online SPIDER-ADMM). As one of major contribution of this paper, we give a new theoretical analysis framework for the nonconvex stochastic ADMM methods with providing the optimal IFO complexity. Based on our new theoretical analysis framework, we study the unsolved optimal IFO complexity of the existing non-convex SVRG-ADMM and SAGA-ADMM methods, and prove they have the optimal IFO complexity of $\mathcal{O}(n + n^{2/3}\epsilon^{-1})$. In the future work, we will apply the stage-wise stochastic momentum technique (Chen et al., 2018) to our methods.

## Acknowledgments

We thank the anonymous reviewers for their helpful comments. F.H. and 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


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

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Candes, E. J., Wakin, M. B., and Boyd, S. P. Enhancing sparsity by reweighted $\ell_1$ minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3): 11, 2011.

Chen, Z., Yang, T., Yi, J., Zhou, B., and Chen, E. Universal stagewise learning for non-convex problems with convergence on averaged solutions. *arXiv preprint arXiv:1808.06296*, 2018.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654, 2014.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator. *arXiv preprint arXiv:1807.01695*, 2018.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

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

Ghadimi, S. and Lan, G. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23:2341–2368, 2013.

Ghadimi, S., Lan, G., and Zhang, H. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155 (1-2):267–305, 2016.

Hazan, E. et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Hong, M., Luo, Z.-Q., and Razaviyayn, M. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

Huang, F. and Chen, S. Mini-batch stochastic admms for nonconvex nonsmooth optimization. *arXiv preprint arXiv:1802.03284*, 2018.

Huang, F., Chen, S., and Lu, Z. Stochastic alternating direction method of multipliers with variance reduction for nonconvex optimization. *arXiv preprint arXiv:1610.02758*, 2016.

Jalali, A., Sanghavi, S., Ruan, C., and Ravikumar, P. K. A dirty model for multi-task learning. In *Advances in neural information processing systems*, pp. 964–972, 2010.

Jiang, B., Lin, T., Ma, S., and Zhang, S. Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis. *Computational Optimization and Applications*, 72(1):115–157, 2019.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pp. 315–323, 2013.

Kim, S., Sohn, K.-A., and Xing, E. P. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12):i204–i212, 2009.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Li, G. and Pong, T. K. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015.

Liu, G., Lin, Z., and Yu, Y. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 663–670, 2010.

Nesterov, Y. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

Nishihara, R., Lessard, L., Recht, B., Packard, A., and Jordan, M. A general analysis of the convergence of admm. In *International Conference on Machine Learning*, pp. 343–352, 2015.

Ouyang, H., He, N., Tran, L., and Gray, A. G. Stochastic alternating direction method of multipliers. *ICML*, 28: 80–88, 2013.

Reddi, S., Sra, S., Poczos, B., and Smola, A. J. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Advances in Neural Information Processing Systems*, pp. 1145–1153, 2016.

Suzuki, T. Dual averaging and proximal gradient descent for online alternating direction multiplier method. In *ICML*, pp. 392–400, 2013.

Suzuki, T. Stochastic dual coordinate ascent with alternating direction method of multipliers. In *ICML*, pp. 736–744, 2014.

Taylor, G., Burmeister, R., Xu, Z., Singh, B., Patel, A., and Goldstein, T. Training neural networks without gradients: a scalable admm approach. In *ICML*, pp. 2722–2731, 2016.

Wang, F., Cao, W., and Xu, Z. Convergence of multi-block bregman admm for nonconvex composite problems. *arXiv preprint arXiv:1505.03063*, 2015a.

Wang, H. and Banerjee, A. Online alternating direction method. In *ICML*, pp. 1119–1126, 2012.

Wang, Y., Yin, W., and Zeng, J. Global convergence of admm in nonconvex nonsmooth optimization. *arXiv preprint arXiv:1511.06324*, 2015b.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., and Tarokh, V. Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv preprint arXiv:1810.10690*, 2018.

Xu, Y., Liu, M., Lin, Q., and Yang, T. Admm without a fixed penalty parameter: Faster convergence with new adaptive penalization. In *Advances in Neural Information Processing Systems*, pp. 1267–1277, 2017.

Yao, Q. and Kwok, J. Efficient learning with a family of nonconvex regularizers by redistributing nonconvexity. In *ICML*, pp. 2645–2654, 2016.

Zheng, S. and Kwok, J. T. Fast and light stochastic admm. In *IJCAI*, 2016a.

Zheng, S. and Kwok, J. T. Stochastic variance-reduced admm. *arXiv preprint arXiv:1604.07070*, 2016b.

Zhong, W. and Kwok, J. Fast stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*, pp. 46–54, 2014.