Faster Stochastic Alternating Direction Method of Multipliers for Nonconvex Optimization

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

1. Department of Electrical and Computer Engineering, University of Pittsburgh, USA
2. College of Computer Science & Technology, Nanjing University of Aeronautics & Astronautics
3. MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
4. JD Finance America Corporation

* heng.huang@pitt.edu

ICML-2019

Long Beach, USA

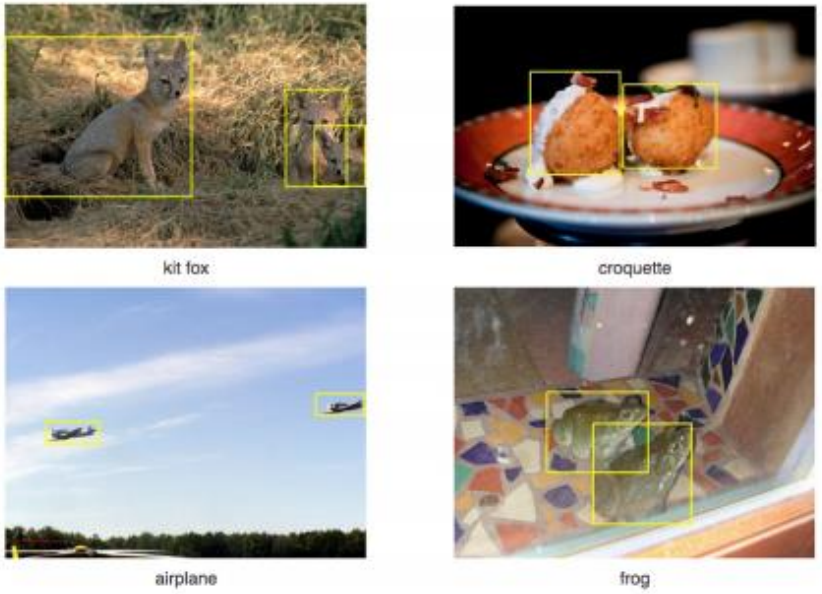
Outline

- Background
- Faster Stochastic ADMM Methods
- Convergence Analysis: Lower IFO Complexity
- Experiments

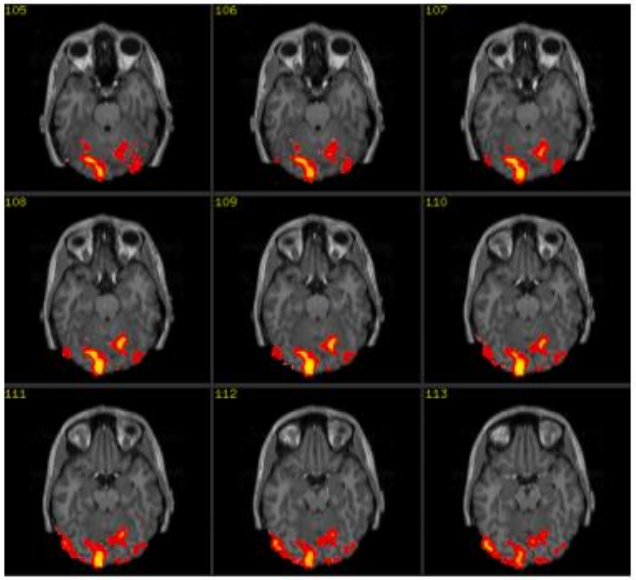
Current Data

Current data not only has large sample size, but also contains some complex structures.

Image classification



fMRI



Problem Statement

- A finite/infinite-sum constraint problem

encoding empirical or expected loss for big data

encoding some complex structures

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

where $f(x) : \mathbb{R}^d \to \mathbb{R}$ 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 $j \in [m]$, $m \geq 1$.

Contributions

In the paper, our main contributions are summarized as follows:

1) We propose a faster stochastic ADMM (*i.e.,* SPIDER-ADMM) method based on the SPIDER method. Moreover, we prove that the SPIDER-ADMM achieves better optimal IFO complexity of $\mathcal{O}(n + n^{1/2}\epsilon^{-1})$ for finding an ϵ-approximate stationary point of the problem (1), 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 give 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 $\mathcal{O}(n^{1/6})$.

IFO Complexity

Table 1. IFO complexity comparison of the non-convex ADMM methods for finding an ϵ-approximate stationary point 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	$\mathcal{O}(\epsilon^{-\frac{3}{2}})$

Faster Stochastic ADMM (SPIDER-ADMM)

Algorithm 1 SPIDER-ADMM Algorithm

Input: b, K, $\rho > 0$ and $\eta > 0$;
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 $\mod(k, q) = 0$ **then**
    Compute $v_k = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x_k)$;
  else
    Uniformly randomly pick a mini-batch $\mathcal{S}$ ($|\mathcal{S}| = b$) from $\{1, 2, \cdots, n\}$ with replacement, then update $v_k = \frac{1}{b}\sum_{i\in\mathcal{S}} \big(\nabla f_i(x_k) - \nabla f_i(x_{k-1})\big) + v_{k-1}$;
  end if
  $y_j^{k+1} = \arg\min_{y_j} \big\{\mathcal{L}_\rho(x_k, y_{[j-1]}^{k+1}, y_j, y_{[j+1:m]}^{k}, z_k) + \frac{1}{2}\|y_j - y_j^k\|_{H_j}^2\big\}$ for all $j \in [m]$;
  $x_{k+1} = \arg\min_x \hat{\mathcal{L}}_\rho\big(x, y_{[m]}^{k+1}, z_k, v_k\big)$;
  $z_{k+1} = z_k - \rho(Ax_{k+1} + \sum_{j=1}^{m} B_j y_j^{k+1} - c)$;
end for
Output: $\{x, y_{[m]}, z\}$ chosen uniformly random from $\{x_k, y_{[m]}^k, z_k\}_{k=1}^K$.

Convergence Analysis

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

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

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

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

where $\gamma = \min(\chi, \sigma_{\min}^H)$ *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 $(x_{k^*}, y_{[m]}^{k^*}, z_{k^*})$ *is an* ϵ*-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* $b = q = \lceil n^{\frac{1}{2}}\rceil$, $\eta = \frac{2\alpha\sigma_{\min}(G)}{3L}$ $(0 < \alpha \leq 1)$ *and* $\rho = \frac{\sqrt{170}\kappa_G L}{\sigma_{\min}^A\alpha}$, *the SPIDER-ADMM has the optimal IFO of* $\mathcal{O}(n + n^{\frac{1}{2}}\epsilon^{-1})$ *for finding an* ϵ*-approximate stationary point. In particular, we can choose* $\alpha \in (0, 1]$ *according to different problems to obtain appropriate step-size* η *and penalty parameter* ρ, e.g., *set* $\alpha = 1$, *we have* $\eta = \frac{2\sigma_{\min}(G)}{3L}$ *and* $\rho = \frac{\sqrt{170}\kappa_G L}{\sigma_{\min}^A}$.

Online SPIDER-ADMM

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

Experiments

1. Graph-Guided Binary Classification

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:

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

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, and A decodes the sparsity pattern of graph, which is obtained by sparse precision matrix estimation.

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

Experiments

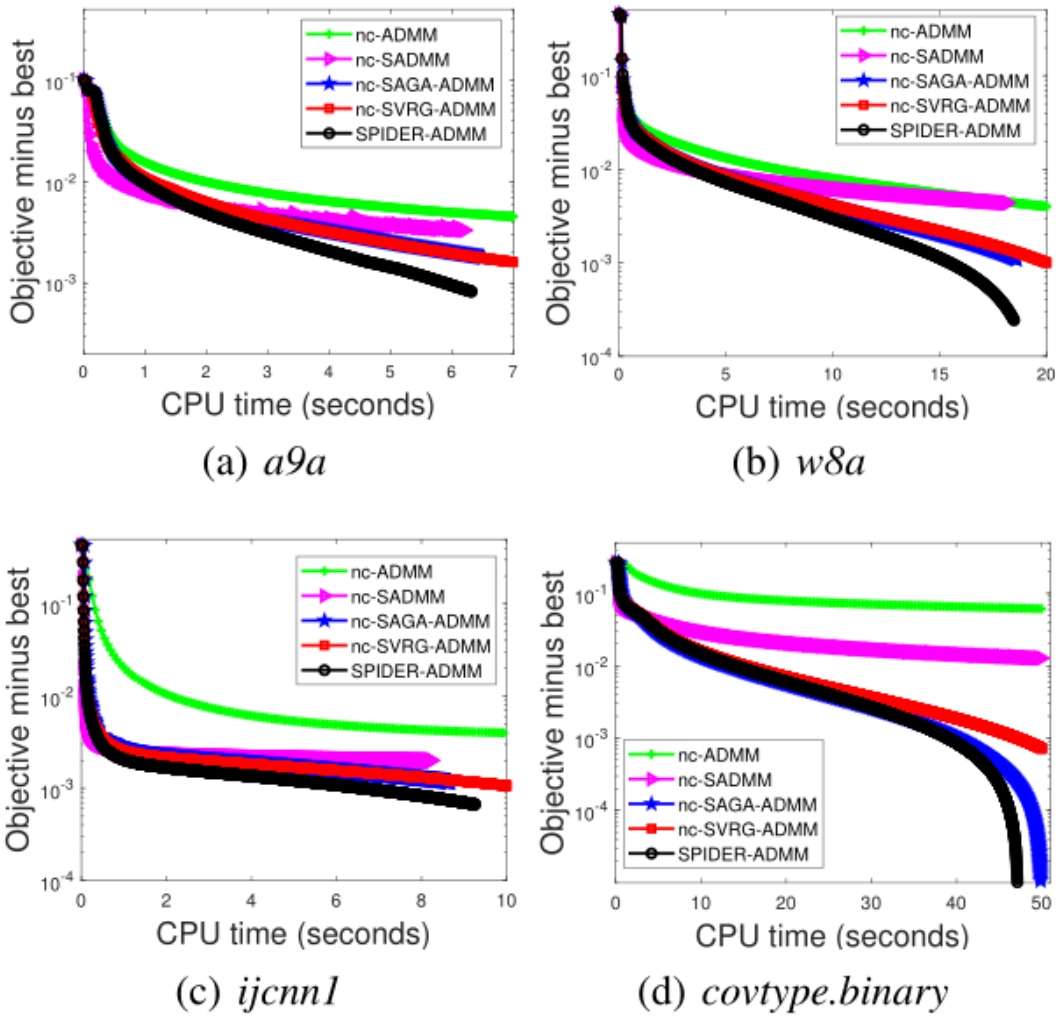


Figure 1. Objective value *versus* CPU time of the *nonconvex* graph-guided binary classification model on some real datasets.

Experiments

2. Multi-Task Learning

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:

$$\min_{X\in\mathbb{R}^{c\times d}} \frac{1}{n}\sum_{i=1}^{n} f_i(X) + \lambda_1 \sum_{ij} \kappa(|X_{ij}|) + \lambda_2 \|X\|_*, \tag{8}$$

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

$$\begin{aligned} \min \frac{1}{n}\sum_{i=1}^{n} \bar{f}_i(X) + \lambda_1\kappa_0\|Y_1\|_1 + \lambda_2\|Y_2\|_* \\ \text{s.t. } AX + B_1Y_1 + B_2Y_2 = 0, \end{aligned} \tag{9}$$

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 $B_2 = [0; -I]$.

Experiments

datasets	# samples	# features	# classes
letter	15,000	16	26
sensorless	58,509	48	11
mnist	60,000	780	10
covtype	581,012	54	7

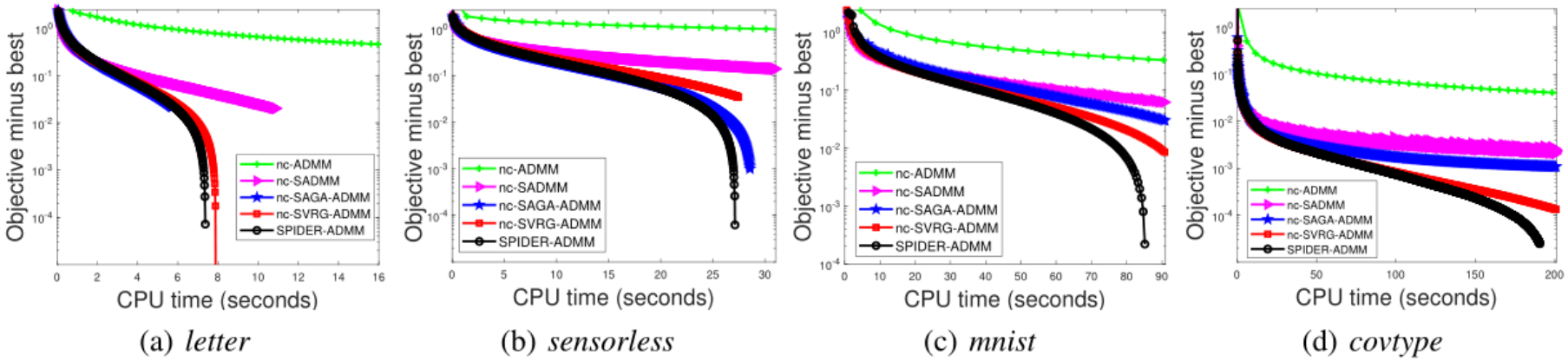


Figure 2. Objective value *versus* CPU time of the *nonconvex* multi-task learning on some real datasets.

Thanks!

Poster: Pacific ballroom #93