MULTI-AGENT DISTRIBUTED LARGE-SCALE OPTIMIZATION BY INEXACT CONSENSUS ALTERNATING DIRECTION METHOD OF MULTIPLIERS

Tsung-Hui Chang[⋆]*, Mingyi Hong*[†] *and Xiangfeng Wang*[‡]

[⋆] Dept. of Elec. & Compt. Eng.
Nat. Taiwan Univ. of Sci. and Tech.,
Taipei, Taiwan 10607
E-mail: tsunghui.chang@ieee.org

[†] Dept. of Elect. & Compt. Eng.
Univ. of Minnesota,
Twin Cities, MN 55455, USA,
E-mail: mhong@umn.edu

[‡] Dept. of Mathematics
Nanjing University,
Nanjing 210093, China
E-mail: xfwang.nju@gmail.com

ABSTRACT

The multi-agent distributed consensus optimization problem arises in many engineering applications. Recently, the alternating direction method of multipliers (ADMM) has been applied to distributed consensus optimization which, referred to as the consensus ADMM (C-ADMM), can converge much faster than conventional consensus subgradient methods. However, C-ADMM can be computationally expensive when the cost function to optimize has a complicated structure or when the problem dimension is large. In this paper, we propose an inexact C-ADMM (IC-ADMM) where each agent only performs one proximal gradient (PG) update at each iteration. The PGs are often easy to obtain especially for structured sparse optimization problems. Convergence conditions for IC-ADMM are analyzed. Numerical results based on a sparse logistic regression problem show that IC-ADMM, though converges slower than the original C-ADMM, has a considerably reduced computational complexity.

Index Terms Distributed consensus optimization, multi-agent network, ADMM, logistic regression

1. INTRODUCTION

Consider a multi-agent network, e.g., a wireless sensor network with distributed sensors, a data cloud network with distributed servers or a computer system with distributed microprocessors. The agents seek to collaborate with each other to accomplish a task [1]. For example, distributed servers in a data cloud network may cooperate for data mining or for parameter learning in order to fully exploit the data collected by individual servers. The agents are assumed able to perform local computation and exchange messages with their neighbors. In general, the task problems can be cast as the following form

$$(\mathsf{P}) \quad \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{N} \Big(f_i(\boldsymbol{A}_i\boldsymbol{x}) + g_i(\boldsymbol{x}) \Big) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^K$ is the decision variable, $\mathcal{X} \subseteq \mathbb{R}^K$ is a feasible set of $\boldsymbol{x}$, and $f_i(\boldsymbol{A}_i\boldsymbol{x}) + g_i(\boldsymbol{x})$ is a cost function associated with agent i where $\boldsymbol{A}_i \in \mathbb{R}^{M\times K}$. The cost function is composed of one smooth component $f_i(\boldsymbol{A}_i\boldsymbol{x})$ and one non-smooth component $g_i(\boldsymbol{x})$ which is often used for regularization purpose [2].

It is assumed that each agent i has knowledge about local information only, i.e., f_i, g_i and $\boldsymbol{A}_i$. Nevertheless, all agents are interested in obtaining the optimal $\boldsymbol{x}$ of (P). Distributed consensus optimization methods, based on the average consensus technique [3] and the subgradient method [4], have been proposed [5, 6]. The consensus subgradient method is appealing due to its simplicity and ability to handle a wide range of applications. However, the convergence of the consensus subgradient method is usually slow.

Recently, the alternating direction method of multipliers (ADMM) [7] has been proposed for distributed consensus optimization, which we refer to as the consensus ADMM (C-ADMM). In [8], several C-ADMM alternatives were developed for solving the sparse LASSO problem [9]. Linear convergence rate of C-ADMM was further analyzed in [10˘12]. One issue that C-ADMM faces in practice is that, at each iteration, each agent has to solve a subproblem globally, which, however, may not always be easy. This is particularly true when the cost functions f_i s have complicated structures or when the problem dimension is large. While a low-accuracy suboptimal solution may be adopted for these subproblems, the convergence behavior of C-ADMM may be greatly impaired.

In this paper, we propose an *inexact consensus ADMM* (IC-ADMM) where agents at each iteration perform one proximal gradient (PG) update [13] only. The PG step is obtained by linearizing the smooth functions f_i s in C-ADMM, which is different from the existing inexact ADMM methods [14,15] which linearize the quadratic term caused by the augmented Lagrangian function. It is known that PGs are efficiently computable in many situations, especially when g_i s are so called sparse promoting functions [2, 13]. We present conditions under which the proposed IC-ADMM can converge globally and have a linear convergence rate. Numerical results will show that the proposed IC-ADMM converges much faster than the consensus subgradient method. While it converges slower than the original C-ADMM, the traded complexity reduction is quite significant, demonstrating the potentials for multi-agent big data applications.

2. NETWORK MODEL AND ASSUMPTIONS

We let a graph $\mathcal{G}$ denote the multi-agent network, which associates with a node set $V = \{1, \ldots, N\}$ and an edge set $\mathcal{E}$. Here, $(i, j) \in \mathcal{E}$ if and only if agent i and agent j are neighbors to each other and can communicate with each other. According to $\mathcal{E}$, an adjacency matrix $\boldsymbol{W} \in \{0, 1\}^{N\times N}$ can be defined, where $[\boldsymbol{W}]_{i,j} = 1$ if $(i, j) \in \mathcal{E}$ and zero otherwise. In addition, one can define an index subset $\mathcal{N}_i = \{j \in V \mid (i, j) \in \mathcal{E}\}$ for the neighbors of each agent i, and a degree matrix $\boldsymbol{D} = \mathrm{diag}\{|\mathcal{N}_1|, \ldots, |\mathcal{N}_N|\}$ which is a diagonal matrix. We assume that

Assumption 1 *The network graph $\mathcal{G}$ of the multi-agent system is connected.*

Assumption 1 implies that neighborhood-wise consensus is sufficient for global consensus in the network. We also make the standard

This work is supported in part by National Science Council, Taiwan (R.O.C.), under grants NSC 102-2221-E-011-005-MY3.

convexity assumption for (P).

Assumption 2 (P) *is a convex problem, that is, f_i's and g_i's are proper closed convex functions and $\mathcal{X}$ is a closed convex set. Moreover, strong duality holds for* (P) *and its dual (e.g., Slater's condition holds).*

3. DISTRIBUTED CONSENSUS ADMM

In this section, we briefly review the consensus ADMM (C-ADMM) method in [8] for solving (P). The C-ADMM method is based on the observation that, under Assumption 1, (P) can be equivalently written as

$$\min_{\boldsymbol{x}_1\in\mathcal{X},\ldots,\boldsymbol{x}_N\in\mathcal{X}} \sum_{i=1}^{N}\Big(f_i(\boldsymbol{A}_i\boldsymbol{x}_i)+g_i(\boldsymbol{x}_i)\Big) \tag{2a}$$

$$\text{s.t. } \boldsymbol{x}_i=\boldsymbol{t}_{ij}\ \forall j\in\mathcal{N}_i,\ i\in V, \tag{2b}$$

$$\boldsymbol{x}_j=\boldsymbol{t}_{ij}\ \forall j\in\mathcal{N}_i,\ i\in V, \tag{2c}$$

where $\{\boldsymbol{t}_{ij}\}$ are slack variables that ensure the consensus between agent i and its neighboring agent j for all $j\in\mathcal{N}_i$ and $i\in V$. A distributed optimization algorithm then can be obtained by applying the standard ADMM algorithm [7] to problem (2). Specifically, let $\{\boldsymbol{u}_{ij}\}$ and $\{\boldsymbol{v}_{ij}\}$ be the dual variables associated with constraints (2b) and (2c), respectively. It has been shown in [8] that, given that $\boldsymbol{u}_{ij}^{(0)}+\boldsymbol{v}_{ij}^{(0)}=\boldsymbol{0}\ \forall i,j$, the ADMM iterations for problem (2) at each iteration k are given by

$$\boldsymbol{u}_{ij}^{(k)}=\boldsymbol{u}_{ij}^{(k-1)}+\frac{c}{2}(\boldsymbol{x}_i^{(k-1)}-\boldsymbol{x}_j^{(k-1)})\ \forall j\in\mathcal{N}_i, i\in V, \tag{3}$$

$$\boldsymbol{v}_{ij}^{(k)}=\boldsymbol{v}_{ij}^{(k-1)}+\frac{c}{2}(\boldsymbol{x}_j^{(k-1)}-\boldsymbol{x}_i^{(k-1)})\ \forall j\in\mathcal{N}_i, i\in V, \tag{4}$$

$$\boldsymbol{x}_i^{(k)}=\arg\min_{\boldsymbol{x}_i} f_i(\boldsymbol{A}_i\boldsymbol{x}_i)+g_i(\boldsymbol{x}_i)+\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)}+\boldsymbol{v}_{ji}^{(k)})^T\boldsymbol{x}_i + c\sum_{j\in\mathcal{N}_i}\Big\|\boldsymbol{x}_i-\frac{\boldsymbol{x}_i^{(k-1)}+\boldsymbol{x}_j^{(k-1)}}{2}\Big\|_2^2\ \forall i\in V, \tag{5}$$

where $c>0$ is a penalty parameter. By further letting $\boldsymbol{p}_i^{(k)}=\sum_{j\in\mathcal{N}_i}(\boldsymbol{u}_{ij}^{(k)}+\boldsymbol{v}_{ji}^{(k)})\ \forall i\in V$, steps in (3) to (5) boil down to Algorithm 1.

Algorithm 1 Consensus ADMM (C-ADMM)

1: **Given** initial variables $\boldsymbol{x}_i^{(0)}\in\mathbb{R}^K$ and $\boldsymbol{p}_i^{(0)}=\boldsymbol{0}$ for each agent i, $i\in V$. Set $k=1$.
2: **repeat**
3: For all $i\in V$
4: $\boldsymbol{p}_i^{(k)}=\boldsymbol{p}_i^{(k-1)}+c\sum_{j\in\mathcal{N}_i}(\boldsymbol{x}_i^{(k-1)}-\boldsymbol{x}_j^{(k-1)})$.

$$\boldsymbol{x}_i^{(k)}=\arg\min_{\boldsymbol{x}_i\in\mathcal{X}} f_i(\boldsymbol{A}_i\boldsymbol{x}_i)+g_i(\boldsymbol{x}_i)+\boldsymbol{x}_i^T\boldsymbol{p}_i^{(k)} + c\sum_{j\in\mathcal{N}_i}\Big\|\boldsymbol{x}_i-\frac{\boldsymbol{x}_i^{(k-1)}+\boldsymbol{x}_j^{(k-1)}}{2}\Big\|_2^2. \tag{6}$$

5: **Set** $k=k+1$.
6: **until** a predefined stopping criterion (e.g., a maximum iteration number) is satisfied.

One can see from Step 4 of Algorithm 1 that each agent i updates the variables $(\boldsymbol{x}_i^{(k)},\boldsymbol{p}_i^{(k)})$ in a fully parallel manner. Moreover, each agent i uses the local function $f_i(\boldsymbol{A}_i\boldsymbol{x}_i)+g_i(\boldsymbol{x}_i)$ and messages $\boldsymbol{x}_j^{(k-1)}\ j\in\mathcal{N}_i$ from its neighbors only. It has been shown in [8] that, under Assumptions 1 and 2, Algorithm 1 is guaranteed to converge and $\lim_{k\to\infty}\boldsymbol{x}_i^{(k)}=\boldsymbol{x}^\star\ \forall i\in V$, where $\boldsymbol{x}^\star$ denotes an optimal solution to (P). Algorithm 1 can also converge linearly, e.g., when $f_i(\boldsymbol{A}_i\boldsymbol{x}_i)$ s are strongly convex and g_i s are absent [10, 11] or when g_i s satisfy certain error bound assumption [12].

It is important to note that Algorithm 1 may not be easy to implement since, at each iteration, agent i has to solve subproblem (6) globally. For example, consider the following sparse logistic regression (LR) problem [16]

$$\min_{\boldsymbol{x}\in\mathcal{X}}\sum_{i=1}^{N}\Big(\sum_{m=1}^{M}\log\Big(1+\exp(-b_{im}\boldsymbol{a}_{im}^T\boldsymbol{x})\Big)+\frac{\lambda}{N}\|\boldsymbol{x}\|_1\Big), \tag{7}$$

where $\lambda>0$ is a regularization parameter, $\boldsymbol{A}_i=[\boldsymbol{a}_{i1},\ldots,\boldsymbol{a}_{iM}]^T$ contains M training data collected by agent i and $b_{im}\in\{\pm1\}$, $m=1,\ldots,M$, are the associated binary labels. The LR problem as in (7) arises in many applications, including document classification, computer vision and language processing, to name a few. When C-ADMM is applied to (7), the associated subproblem (6) would not yield simple solutions, and additional numerical solver has to be employed. When the problem dimension is large, obtaining a high-accuracy solution of (6) can be computationally expensive. While a low-accuracy solution can be adopted for complexity reduction, it may impair the convergence behavior of C-ADMM considerably.

4. PROPOSED INEXACT CONSENSUS ADMM

In this section, aiming at reducing the computation overhead, we propose an inexact consensus ADMM (IC-ADMM). In IC-ADMM, instead of solving subproblem (6) directly, we consider the following update of $\boldsymbol{x}_i^{(k)}$:

$$\boldsymbol{x}_i^{(k)}=\arg\min_{\boldsymbol{x}_i\in\mathcal{X}} \nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)})^T\boldsymbol{A}_i(\boldsymbol{x}_i-\boldsymbol{x}_i^{(k-1)}) + \frac{\beta}{2}\|\boldsymbol{x}_i-\boldsymbol{x}_i^{(k-1)}\|_2^2+g_i(\boldsymbol{x}_i)+\boldsymbol{x}_i^T\boldsymbol{p}_i^{(k)} + c\sum_{j\in\mathcal{N}_i}\Big\|\boldsymbol{x}_i-\frac{\boldsymbol{x}_i^{(k-1)}+\boldsymbol{x}_j^{(k-1)}}{2}\Big\|^2, \tag{8}$$

where $\beta>0$ is a penalty parameter. Equation (8) is obtained by replacing $f_i(\boldsymbol{A}_i\boldsymbol{x}_i)$ in (6) with its linearized and regularized counterpart $\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)})^T\boldsymbol{A}_i(\boldsymbol{x}_i-\boldsymbol{x}_i^{(k-1)})+\frac{\beta}{2}\|\boldsymbol{x}_i-\boldsymbol{x}_i^{(k-1)}\|_2^2$. Define

$$\text{prox}_{g_i}(\boldsymbol{s})\triangleq\arg\min_{\boldsymbol{x}\in\mathcal{X}} g_i(\boldsymbol{x})+\frac{\gamma_i}{2}\|\boldsymbol{x}-\boldsymbol{s}\|^2 \tag{9}$$

as a proximity operator [13], where $\gamma_i=\beta+2c|\mathcal{N}_i|$. Equation (8) can be shown to be equivalent to the following proximal gradient (PG) step

$$\boldsymbol{x}_i^{(k)}=\text{prox}_{g_i}\Big[\frac{1}{\gamma_i}\Big(\beta\boldsymbol{x}_i^{(k-1)}-\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)})\boldsymbol{x}_i^{(k-1)}-\boldsymbol{p}_i^{(k)} + c\sum_{j\in\mathcal{N}_i}(\boldsymbol{x}_i^{(k-1)}+\boldsymbol{x}_j^{(k-1)})\Big)\Big]. \tag{10}$$

It is known that a PG update like (10) can often have close-form expressions, especially when g_i s are so called sparse promoting functions. For example, when $g_i(\boldsymbol{x})=\|\boldsymbol{x}\|_1$ and $\mathcal{X}=\mathbb{R}^K$, (9) has a closed-form solution known as the soft thresholding operator [13]:

$$\mathcal{S}[\boldsymbol{s},\frac{1}{\gamma_i}]=(\boldsymbol{s}-\frac{1}{\gamma_i}\boldsymbol{1})^+-(-\boldsymbol{s}-\frac{1}{\gamma_i}\boldsymbol{1})^+, \tag{11}$$

where $(x)^+ = \max\{x, 0\}$ and $\mathbf{1}$ is an all-one vector. The proposed IC-ADMM method is presented in Algorithm 2.

Algorithm 2 Proposed Inexact Consensus ADMM (IC-ADMM)

1: **Given** initial variables $\boldsymbol{x}_i^{(0)} \in \mathbb{R}^K$ and $\boldsymbol{p}_i^{(0)} = \mathbf{0}$ for each agent i, $i \in V$. Set $k = 1$.
2: **repeat**
3: For all $i \in V$
4: $\boldsymbol{p}_i^{(k)} = \boldsymbol{p}_i^{(k-1)} + c\sum_{j\in\mathcal{N}_i}(\boldsymbol{x}_i^{(k-1)} - \boldsymbol{x}_j^{(k-1)})$.
5: Update $\boldsymbol{x}_i^{(k)}$ by (10).
6: **Set** $k = k + 1$.
7: **until** a predefined stopping criterion is satisfied.

To show the convergence of IC-ADMM, we make the following assumption on $f_i's$.

Assumption 3 *For all $i \in V$, the smooth function f_i is strongly convex, i.e., for some $\sigma_f^2 > 0$,*

$$(\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y}))^T(\boldsymbol{x} - \boldsymbol{y}) \geq \sigma_f^2\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^M,$$

Moreover, f_i has Lipschitz continuous gradients, i.e.,

$$\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\|_2 \leq L_{f,i}\|\boldsymbol{x} - \boldsymbol{y}\|_2 \ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^M, \quad (12)$$

for some $L_f > 0$.

Note that, even under Assumption 3, $f_i(\boldsymbol{A}_i\boldsymbol{x})$ is not necessarily strongly convex in $\boldsymbol{x}$ since the mapping matrix $\boldsymbol{A}_i$ is allowed to be fat and rank deficient. Both the LASSO problem [8, 9] and the LR problem in (7) (given that $\mathcal{X}$ is compact) satisfy Assumption 3. Our main convergence result of IC-ADMM is given below.

Theorem 1 *Suppose that Assumptions 1, 2 and 3 hold and let*

$$\beta > \frac{L_f^2}{\sigma_f^2}\lambda_{\max}(\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}}) - c\lambda_{\min}(\boldsymbol{D} + \boldsymbol{W}) > 0, \quad (13)$$

where $\tilde{\boldsymbol{A}} = \text{blkdiag}\{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N\}$ (block diagonal) and $\lambda_{\max}$ and $\lambda_{\min}$ denote the maximum and minimum eigenvalues, respectively.

(a) Then, $\{\boldsymbol{x}_1^{(k)}, \ldots, \boldsymbol{x}_N^{(k)}\}$ in Algorithm 2 converges to a common point $\boldsymbol{x}^\star$ that is optimal to (P).

(b) If g_i's are removed from (1) *and $\boldsymbol{A}_i$'s have full column rank (i.e., $f_i(\boldsymbol{A}_i\boldsymbol{x})$ is strongly convex in $\boldsymbol{x}$ for all i), then the sequence $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_{\mathbf{M}}^2 + \frac{1}{c}\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^\star\|^2$ converges linearly, where $\boldsymbol{x}^{(k)} = [(\boldsymbol{x}_1^{(k)})^T, \ldots, (\boldsymbol{x}_N^{(k)})^T]^T$, $\boldsymbol{u}_i^{(k)} \in \mathbb{R}^{K|\mathcal{N}_i|}$ is a vector that stacks $\boldsymbol{u}_{ij}^{(k)}$ $\forall j \in \mathcal{N}_i$ [see (3)], $\boldsymbol{u}^{(k)} = [(\boldsymbol{u}_1^{(k)})^T, \ldots, (\boldsymbol{u}_N^{(k)})^T]^T$, and*

$$\boldsymbol{G} = \beta\boldsymbol{I}_{KN} + c((\boldsymbol{D} + \boldsymbol{W}) \otimes \boldsymbol{I}_K) \succ \mathbf{0}, \quad (14)$$

$$\mathbf{M} = \left[\frac{1}{2}\boldsymbol{G} + \alpha(\sigma_f^2 - \frac{\rho}{2})\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}}\right]^{1/2} \succ \mathbf{0}, \quad (15)$$

for some $0 < \alpha < 1$ and $\rho > 0$. Here, $\boldsymbol{I}_K$ is the $K \times K$ identity matrix.

Due to the space limitation, the proof of Theorem 1 is presented in [17]. Theorem 1(a) implies that, given a β satisfying (13), IC-ADMM ensures that all agents achieve consensus and attain the optimal solution of (P). Theorem 1(b) further asserts that IC-ADMM can converge linearly if $f_i(\boldsymbol{A}_i\boldsymbol{x}_i)$ is strongly convex in $\boldsymbol{x}_i$ for all i and the non-smooth g_i s are not present. Theorem 1(b) thus extends the analysis result in [10] of C-ADMM to the IC-ADMM.

Two remarks regarding the proposed IC-ADMM are in order.

Remark 1 In [8], several alternatives were proposed to reduce the complexity of C-ADMM for solving the LASSO problem. However, these approaches are specifically devised for the least squared error function of LASSO, and may not provide simple solutions for, for instance, the LR problem in (7). Our IC-ADMM by contrast are applicable to a wider range of problems. On the other hand, one should note that the proposed IC-ADMM is different from the existing inexact ADMM in the optimization literature; see [14, 15] where the inexact update is obtained by linearizing the quadratic terms caused by the augmented Lagrangian function. The work [18] considered linearizing the cost function but requires additional back substitution steps and is not designed for multi-agent distributed optimization.

Remark 2 The value of $\lambda_{\min}(\boldsymbol{D} + \boldsymbol{W})$ in (13) depends on the network topology. Let $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ be the Laplacian matrix of $\mathcal{G}$. Then $\boldsymbol{D} + \boldsymbol{W} = 2\boldsymbol{D} - \boldsymbol{L}$. By the graph theory [19], the normalized Laplacian matrix, i.e., $\tilde{\boldsymbol{L}} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}}$, have $\lambda_{\max}(\tilde{\boldsymbol{L}}) \leq 2$ and $\lambda_{\max}(\tilde{\boldsymbol{L}}) < 2$ if and only if the connected graph $\mathcal{G}$ is not bipartite. Thus, we have $\lambda_{\min}(\boldsymbol{D} + \boldsymbol{W}) = \lambda_{\min}(\boldsymbol{D}^{\frac{1}{2}}(2\boldsymbol{I}_N - \tilde{\boldsymbol{L}})\boldsymbol{D}^{\frac{1}{2}}) \geq 0$, and $\lambda_{\min}(\boldsymbol{D} + \boldsymbol{W}) > 0$ whenever $\mathcal{G}$ is non-bipartite.

5. NUMERICAL RESULTS AND DISCUSSIONS

In this section, we examine the numerical performances of C-ADMM (Algorithm 1) and the proposed IC-ADMM (Algorithm 2) by considering the LR problem in (7). To generate the training data $\boldsymbol{A}_i$ s, we considered images D24 and D68 of the Brodatz data set (`http://www.ux.uis.no/~tranden/brodatz.html`). We randomly extracted $MN/2$ overlapping patches with dimension $\sqrt{K} \times \sqrt{K}$ from the two images, respectively, followed by vectorizing the MN patches into vectors and stacking them into an $MN \times K$ matrix. After a random shuffle the order of rows of this matrix, we partitioned the matrix into N submatrices each with dimension $M \times K$, which were then used as the training data $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N$. The binary labels were generated accordingly with 1 for one image and -1 for the other. The feasible set was set to $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^K \mid |x_i| \leq 1 \ \forall i\}$.

To implement Algorithm 1, we employed the fast iterative shrinkage-thresholding algorithm (FISTA) [20, 21] to solve subproblem (6). For (6), the associated steps can be shown as

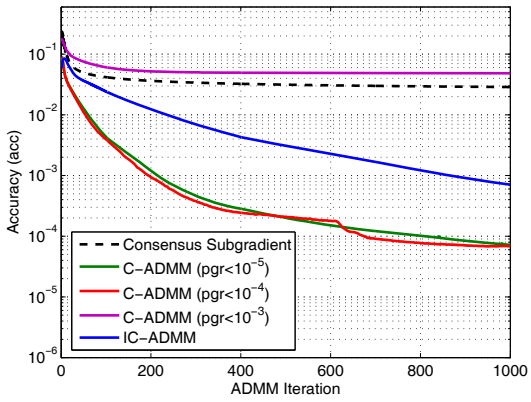
$$\tilde{\boldsymbol{x}}_i^{(\ell)} = \mathcal{S}\Bigg[\boldsymbol{z}_i^{(\ell-1)} - \rho_i^{(\ell)}\Bigg[\boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{z}_i^{(\ell-1)}) + \boldsymbol{p}_i^{(k)} + 2c\sum_{j\in\mathcal{N}_i}(\boldsymbol{z}_i^{(\ell-1)} - \frac{\boldsymbol{x}_i^{(k-1)} + \boldsymbol{x}_j^{(k-1)}}{2})\Bigg], \frac{\lambda\rho_i^{(\ell)}}{N}\Bigg], \quad (16a)$$

$$\boldsymbol{z}_i^{(\ell)} = \tilde{\boldsymbol{x}}_i^{(\ell)} + \frac{\ell - 1}{\ell + 2}(\tilde{\boldsymbol{x}}_i^{(\ell)} - \tilde{\boldsymbol{x}}_i^{(\ell-1)}), \quad (16b)$$

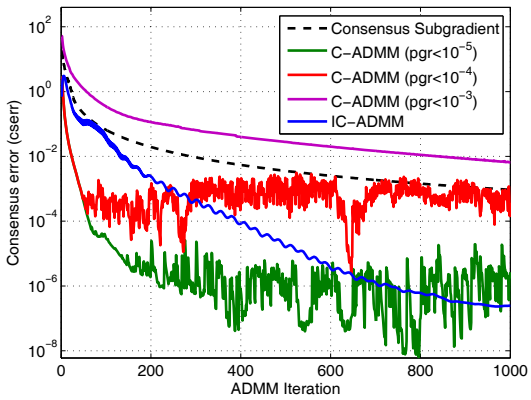
where ℓ denotes the inner iteration index of FISTA, $\rho_i^{(\ell)} > 0$ is a step size and $\mathcal{S}$ is the soft thresholding operator defined in (11). Suppose that FISTA stops at iteration $\ell_i(k)$. We then set $\boldsymbol{x}_i^{(k)} = \tilde{\boldsymbol{x}}_i^{(\ell_i(k))}$ for subproblem (6). The stopping criterion of (16) was based on the PG residue (pgr) $\text{pgr} = \|\boldsymbol{z}_i^{(\ell-1)} - \tilde{\boldsymbol{x}}_i^{(\ell)}\|/(\rho_i^{(\ell)}\sqrt{K})$ [20, 21]. For obtaining a high-accuracy solution of (6), one may set the stopping criterion as, e.g., $\text{pgr} < 1\text{e}^{-5}$.

By applying the proposed IC-ADMM to (7), the corresponding (10) can be shown to be

$$\boldsymbol{x}_i^{(k)} = \frac{1}{\gamma_i}\mathcal{S}\Bigg[\beta\boldsymbol{x}_i^{(k-1)} - \boldsymbol{A}_i^T\nabla f_i(\boldsymbol{A}_i\boldsymbol{x}_i^{(k-1)}) - \boldsymbol{p}_i^{(k)} + c\sum_{j\in\mathcal{N}_i}(\boldsymbol{x}_i^{(k-1)} + \boldsymbol{x}_j^{(k-1)}), \frac{\lambda}{N}\Bigg]. \quad (17)$$



(a) Normalized accuracy



(b) Consensus error

Fig. 1. Convergence curves for the example of $N = 10$, $K = 10,000$, $M = 10$, $\lambda = 0.1$.

By comparing (17) with (16a), one can see that, for each agent i, the computational complexity of Algorithm 1 per iteration k (we refer this as the ADMM iteration (ADMM Ite.)*f*) is roughly $\ell_i(k)$ times that of Algorithm 2. To measure the computational complexity of Algorithm 1, we count the total average number of FISTA iterations implemented by each agent before Algorithm 1 stops. More precisely, suppose that the total number of ADMM iterations of Algorithm 1 is $\bar{k}$. Then the complexity per agent due to Algorithm 1 is measured as

$$\text{Computation iteration (Compt. Ite.)} = \frac{1}{N}\sum_{k=1}^{\bar{k}}\sum_{i=1}^{N}\ell_i(k).$$

By contrast, the complexity per agent due to Algorithm 2 is simply given by $\bar{k}$ if the total number of ADMM iterations of Algorithm 2 is $\bar{k}$. The stopping criterion of Algorithms 1 and 2 was based on measuring the solution accuracy $\mathsf{acc} = (\mathsf{obj}(\hat{\boldsymbol{x}}^{(k)}) - \mathsf{obj}^\star)/\mathsf{obj}^\star$ and variable consensus error $\mathsf{cserr} = \sum_{i=1}^N \|\hat{\boldsymbol{x}}^{(k)} - \boldsymbol{x}_i^{(k)}\|^2/N$, where $\hat{\boldsymbol{x}}^{(k)} = (\sum_{i=1}^N \boldsymbol{x}_i^{(k)})/N$, $\mathsf{obj}(\hat{\boldsymbol{x}}^{(k)})$ denotes the objective value of (P) given $\boldsymbol{x} = \hat{\boldsymbol{x}}^{(k)}$, and $\mathsf{obj}^\star$ is the optimal value of (P) which was obtained by using FISTA [20,21]. The two algorithms are set to stop whenever acc and cserr are both smaller than preset target values.

In Table 1(a), we consider a simulation example of $N = 10$, $K = 10,000$, $M = 10$, $\lambda = 0.1$ and display the comparison results. The stopping conditions of C-ADMM and IC-ADMM are acc $< 10^{-4}$, cserr $< 10^{-5}$. For C-ADMM, we considered two cases, one with the stopping condition of FISTA for solving subproblem (6) set to pgr $< 10^{-5}$ and one with that set to pgr $< 10^{-4}$. The penalty parameter c for C-ADMM was set to $c = 0.03$ and the step size $\rho_i^{(\ell)}$ of FISTA (see (16)) was set to a constant $\rho_i^{(\ell)} = 0.1$. The penalty parameters c and β of IC-ADMM were set to $c = 0.01$ and $\beta = 1.2$. We observe from Table 1 that IC-ADMM in general requires more ADMM iterations than C-ADMM (around 4 times); however, the required computation complexity is significantly lower. Specifically, the number of computation iterations of IC-ADMM is around $81,459/2973 \approx 27.4$ times lower than that of C-ADMM (pgr $< 10^{-5}$). We also observe that C-ADMM (pgr $< 10^{-4}$) consumes a smaller number of computation iterations for achieving acc $< 10^{-4}$. However, the associated cserr=3.425×10^{-4} is quite large. In fact, C-ADMM (pgr $< 10^{-4}$) cannot reduce cserr properly. To see more clearly, we plot the acc and cserr curves of C-ADMM and IC-ADMM in Figure 1. One can see from Figure 1(b) that the cserr curve of C-ADMM (pgr $< 1e^{-4}$) keeps relatively high and does not decrease along the iterations. When one further reduces the accuracy of FISTA to pgr $< 10^{-3}$, C-ADMM converges very slowly, as shown in Fig. 1. In the figure, we also plot the convergence curves of the consensus subgradient method in [5], where the diminishing step size $10/k$ was used. While the consensus subgradient method is also simple, it converges much slower than IC-ADMM.

Table 1. Comparison of C-ADMM and IC-ADMM.

(a) $N = 10$, $K = 10,000$, $M = 10$, $\lambda = 0.1$.

	C-ADMM (pgr $< 10^{-5}$)	**C-ADMM** (pgr $< 10^{-4}$)	**IC-ADMM**
ADMM Ite.	810	675	2973
Compt. Ite.	**81,459**	**30,648**	**2973**
acc$< 10^{-4}$	9.982×10^{-5}	9.91×10^{-5}	9.99×10^{-5}
cserr$< 10^{-5}$	1.53×10^{-6}	$\mathbf{3.425 \times 10^{-4}}$	$\mathbf{3.859 \times 10^{-9}}$

(b) $N = 50$, $K = 10,000$, $M = 10$, $\lambda = 0.15$.

	C-ADMM (pgr $< 10^{-5}$)	**C-ADMM** (pgr $< 10^{-4}$)	**IC-ADMM**
ADMM Ite.	952	N/A	7,251
Compt. Ite.	$\mathbf{1.432 \times 10^5}$	N/A	**7,251**
acc$< 10^{-4}$	9.99×10^{-5}	N/A	9.999×10^{-5}
cserr$< 10^{-5}$	1.305×10^{-7}	N/A	$\mathbf{1.169 \times 10^{-10}}$

In Table 1(b), we considered another example with the network size increased to $N = 50$. We set $c = 0.004$ for C-ADMM and $\rho_i^{(\ell)} = 0.1$ for FISTA; while for IC-ADMM, we set $c = 0.008$ and $\beta = 1.2$. We can observe similar comparison results from Table 1(b). Specifically, the number of computation iterations of IC-ADMM is around 19.7 times lower than C-ADMM (pgr $< 1e^{-5}$); when considering a lower accuracy of pgr $< 10^{-4}$, it is found that C-ADMM cannot properly converge.

Since one ADMM iteration corresponds to one time of communication between neighboring agents, the numerical results presented above indicate that IC-ADMM gains complexity reduction in the expense of communication overhead. Therefore, IC-ADMM is suitable for applications where message exchanges between neighboring agents can be achieved cheaply; for example, distributed data servers connected via dedicated fiber links or distributed microprocessors in a computer system.

6. REFERENCES

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, Cloud computing and grid computing 360-degree compared,*f* in *Proc. Grid Computing Environments Workshop*, Austin, TX, USA, Nov. 12-16, 2008, pp. 1˘10.

[2] M. Elad, *Sparse and Redundant Rerpesentations*. New York, NY, USA: Springer Science + Business Media, 2010.

[3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, Randomized gossip algorithms,*f IEEE Trans. Info. Theory*, vol. 52, pp. 2508˘2530, June 2006.

[4] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex analysis and optimization*. Cambridge, Massachusetts: Athena ScientiŒe, 2003.

[5] A. Nedić, A. Ozdaglar, , and A. Parrilo, Constrained consensus and optimization in multi-agent networks,*f IEEE Trans. Automatic Control*, vol. 55, no. 4, pp. 922˘938, April 2010.

[6] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson, Subgradient methods and consensus algorithms for solving convex optimization problems,*f* in *Proc. IEEE CDC*, Cancun, Mexica, Dec. 9-11, 2008, pp. 4185˘4190.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[8] G. Mateos, J. A. Bazerque, and G. B. Giannakis, Distributed sparse linear regression,*f IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262˘5276, Dec. 2010.

[9] R. Tibshirani, Regression shrinkage and selection via the LASSO,*f J. Roy. Stat. Soc. B*, vol. 58, pp. 267˘288, 1996.

[10] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, Linearly convergent decentralized consensus optimization with alternating direction method of multipliers,*f* in *Proc. IEEE ICASSP*, Vancouver, Canada, May 26-31, 2013, pp. 4613˘4616.

[11] , On the linear convergence of the ADMM in decentralized consensus optimization,*f* available on arxiv.org.

[12] M. Hong and Z.-Q. Luo, On the linear convergence of the alternating direction method of multipliers,*f* available on arxiv.org.

[13] Y. Nesterov, Smooth minimization of nonsmooth functions,*f Math. Program.*, vol. 103, no. 1, pp. 127˘152, 2005.

[14] B. He and X. Yuan, Linearized alternating direction method of multipliers with gaussian back substitution for separable convex programming,*f Numerial Algebra. Control and Optimization*, vol. 3, no. 2, pp. 247˘260, 2013.

[15] S. Ma, Alternating proximal gradient method for convex minimization,*f* available on http://www.optimization-online.org/.

[16] I. Foster, Y. Zhao, I. Raicu, and S. Lu, Large-scale sparse logistic regression,*f* in *Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, June 28 - July 1, 2009, pp. 547˘556.

[17] T.-H. Chang, M. Hong, and X. Wang, Multi-agent distributed optimization via inexact consensus ADMM,*f* manuscript, available on arxiv.org.

[18] B. He, Z. Peng, and X. Wang, Proximal alternating direction-based contraction methods for separable linearly constrained convex optimization,*f Frontiers of Math. in China*, vol. 6, no. 1, pp. 79˘114, 2011.

[19] F. R.-K. Chung, *Spectral graph theory*. CBMS Regional Conference Series in Mathematics, No. 92, 1996.

[20] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems,*f SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183˘202, 2009.

[21] S. Becker, E. Candes, and M. Grant, Templates for convex cone problems with applications to sparse signal recovery,*f Math. Program. Compt.*, pp. 1˘54, 2011.