Distributed Learning with Non-Smooth Objective Functions

Cristiano Gratton*, Naveen K. D. Venkategowda*, Reza Arablouei†, Stefan Werner*
* Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway
† CSIRO's Data61, Pullenvale QLD 4069, Australia

***Abstract*—We develop a new distributed algorithm to solve a learning problem with non-smooth objective functions when data are distributed over a multi-agent network. We employ a zeroth-order method to minimize the associated augmented Lagrangian in the primal domain using the alternating direction method of multipliers (ADMM) to develop the proposed algorithm, named distributed zeroth-order based ADMM (D-ZOA). Unlike most existing algorithms for non-smooth optimization, which rely on calculating subgradients or proximal operators, D-ZOA only requires function values to approximate gradients of the objective function. Convergence of D-ZOA to the centralized solution is confirmed via theoretical analysis and simulation results.**

I. Introduction

Performing learning tasks at a central processing unit in a large distributed network can be prohibitive due to communication/computation costs or privacy issues. Therefore, it is important to develop algorithms that are able to distributedly process the data collected by agents scattered over a large geographical area [1]–[8]. In this context, each agent in the network only possesses information of a local cost function and the agents aim to collaboratively minimize the sum of the local objective functions. Such optimization problems are relevant to several applications in statistics [3]–[5], signal processing [6]–[8] and control [1], [2].

There have been several works developing algorithms for solving distributed convex optimization problems over ad-hoc networks. However, many existing algorithms only offer solutions for problems with smooth objective functions, see, e.g., [5], [9], [10]. Distributed optimization problems with non-smooth objectives have been considered in [1], [2], [4], [11]–[16]. The approaches taken in [2], [11], [12] are based on subgradient methods. The works of [13], [14] are based on dual decomposition techniques while the algorithms in [4], [15] are developed using soft-thresholding operations. However, all the aforementioned algorithms require either the computation of subgradients, which might be hard to achieve for some objectives, or derivation of proximal operators, which might not be feasible in some scenarios.

Moreover, there are some real-world problems where obtaining first-order information is impossible due to the lack of the complete loss function. For example, in bandit optimization [17], an adversary generates a sequence of loss functions and the goal is to minimize such sequence that is only available at some points. In addition, in simulation-based optimization, the objective is available only using repeated simulation [18], and in adversarial black-box machine learning models, only the function values are given [19]. This motivates the use of zeroth-order methods requiring only function values to approximate gradients.

The works in [1], [16] are based on zeroth-order methods within the distributed optimization setting. While the approach of [16] relies on approximate projections for dealing with constraints, the algorithm ZONE-S proposed in [1] is based on a primal-dual approach and deals with non-convex objectives. However, ZONE-S addresses only consensus problems with a non-smooth regularization that is handled by a central collector making the algorithm not fully distributed.

In this paper, we develop a fully-distributed algorithm to solve an optimization problem with a non-smooth convex objective function over an ad-hoc network. We utilize the alternating direction method of multipliers (ADMM) for distributed optimization. Furthermore, we employ the zeroth-order method called the two-point stochastic gradient algorithm [20] that is suitable for non-smooth objectives to obtain an approximate minimizer of the augmented Lagrangian in the ADMM's primal update step. The proposed algorithm, called distributed zeroth-order based ADMM (D-ZOA), is fully distributed in the sense that each agent in the network communicates only with its neighbors and no central coordinator is necessary. Furthermore, D-ZOA does not compute any subgradient and only requires the objective function values to approximate the gradient of the augmented Lagrangian. The simulations show that D-ZOA is competitive even on a problem that can be easily solved with a subgradient-based algorithm. Furthermore, the experiments show the usefulness of D-ZOA on a problem where calculating any subgradient is impractical. Convergence of D-ZOA to the centralized solution at all agents is verified through theoretical analysis and simulation results.

Mathematical Notations: The set of natural and real numbers are denoted by $\mathbb{N}$ and $\mathbb{R}$, respectively. Scalars, column vectors and matrices are respectively denoted by lowercase, bold lowercase, and bold uppercase letters. The operators $(\cdot)^\mathsf{T}$ and $\mathrm{tr}(\cdot)$ denote transpose and trace of a matrix, respectively. $\mathbf{I}_p$ denotes an identity matrix of size p, $\mathbf{0}_{q\times l}$ defines a matrix with all zero entries, and $\otimes$ stands for the Kronecker product. The statistical expectation and covariance operators are represented by $\mathbb{E}[\cdot]$ and $\mathrm{cov}[\cdot]$, respectively. For a vector $\mathbf{y}$ and a matrix $\mathbf{Y} \in \mathbb{R}^{r\times s}$, $\|\mathbf{y}\|_\mathbf{Y}$ denotes the quadratic form $\mathbf{y}^\mathsf{T}\mathbf{Y}\mathbf{y}$. The

This work was partly supported by the Research Council of Norway.

nuclear norm of $\mathbf{Y}$ is denoted by $\|\mathbf{Y}\|_*$ and is defined as

$$\|\mathbf{Y}\|_* = \sum_{i=1}^{\min\{r,s\}} \sigma_i(\mathbf{Y})$$

where $\sigma_i(\mathbf{Y})$ denotes the ith singular value of $\mathbf{Y}$. $\|\cdot\|$ and $\|\cdot\|_F$ represent the Euclidean norm and the Frobenius norm, respectively. The operators vec$(\mathbf{Y})$ forms a column vector from the matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_s]$ by stacking the column vectors $\mathbf{y}_i$. For a positive semidefinite matrix $\mathbf{X}$, $\lambda_{\min}(\mathbf{X})$ and $\lambda_{\max}(\mathbf{X})$ denote the nonzero smallest and largest eigenvalues of $\mathbf{X}$, respectively.

II. System Model

We consider a network with $K \in \mathbb{N}$ agents and $E \in \mathbb{N}$ edges that is modeled as an undirected graph $\mathcal{G}(\mathcal{K}, \mathcal{E})$, where the set of vertices $\mathcal{K} = \{1, \ldots, K\}$ corresponds to the agents and the set $\mathcal{E}$ represents the bidirectional communication links between the pairs of agents. Agent $k \in \mathcal{K}$ can communicate only with the agents in its neighborhood $\mathcal{N}_k$ whose cardinality is denoted by $|\mathcal{N}_k|$. By convention, the set $\mathcal{N}_k$ includes the agent k as well.

We consider the problem when the K agents of the network solve the following minimization problem collaboratively

$$\min_{\mathbf{x}} \sum_{k=1}^{K} f_k(\mathbf{x}; \mathcal{X}_k) \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^P$ is the unknown model parameter, $\mathcal{X}_k$ represents the local information at agent k, and $f_k : \mathbb{R}^P \to \mathbb{R}$ is the local cost function that is convex but *non-smooth*. Let us denote the solution to (1) by $\mathbf{x}^c$.

III. Non-Smooth Distributed Learning

We first discuss the consensus-based reformulation of the problem that allows its distributed solution through an iterative process consisting of two nested loops. Then, we describe the ADMM procedure that forms the outer loop and the zeroth-order two-point stochastic gradient algorithm that constitutes the inner loop solving the ADMM primal update step. Finally, we establish the convergence of D-ZOA theoretically.

A. Consensus-Based Reformulation

To solve (1) in a distributed fashion, we introduce the primal variables $\mathcal{V} \coloneqq \{\mathbf{x}_k\}_{k=1}^K$ that represent local copies of $\mathbf{x}$ at the agents. Then, we reformulate (1) as the following constrained minimization problem:

$$\begin{aligned} \min_{\{\mathbf{x}_k\}} \quad & \sum_{k=1}^{K} f_k(\mathbf{x}_k; \mathcal{X}_k) \\ \text{s.t.} \quad & \mathbf{x}_k = \mathbf{x}_l, \quad l \in \mathcal{N}_k, \quad \forall k \in \mathcal{K}. \end{aligned} \tag{2}$$

Since the network is connected, the equality constraints in (2) enforce consensus over $\{\mathbf{x}_k\}_{k=1}^K$ by imposing consensus across each agent's neighborhood $\mathcal{N}_k$. To solve (2) in a distributed fashion, we employ the ADMM [8]. Therefore, we introduce the auxiliary variables $\mathcal{Z} \coloneqq \{\mathbf{z}_k^l\}_{l \in \mathcal{N}_k}$ and rewrite (2) as

$$\begin{aligned} \min_{\{\mathbf{x}_k\}} \quad & \sum_{k=1}^{K} f_k(\mathbf{x}_k; \mathcal{X}_k) \\ \text{s.t.} \quad & \mathbf{x}_k = \mathbf{z}_k^l, \ \mathbf{x}_l = \mathbf{z}_k^l, \quad l \in \mathcal{N}_k, \quad \forall k \in \mathcal{K}. \end{aligned} \tag{3}$$

The use of auxiliary variables $\mathcal{Z}$ renders an equivalent representation of the constraints in (2). These variables are only used to derive the local recursions and are eventually eliminated. The augmented Lagrangian function is given by

$$\begin{aligned} \mathcal{L}_\rho(\mathcal{V}, \mathcal{Z}, \mathcal{M}) &= \sum_{k=1}^{K} f_k(\mathbf{x}_k; \mathcal{X}_k) \\ &+ \sum_{k=1}^{K} \sum_{l \in \mathcal{N}_k} \left[\boldsymbol{\mu}_k^{l\mathsf{T}} \left(\mathbf{x}_k - \mathbf{z}_k^l \right) + \boldsymbol{\lambda}_k^{l\mathsf{T}} \left(\mathbf{x}_l - \mathbf{z}_k^l \right) \right] \\ &+ \frac{\rho}{2} \sum_{k=1}^{K} \sum_{l \in \mathcal{N}_k} \left(\left\| \mathbf{x}_k - \mathbf{z}_k^l \right\|^2 + \left\| \mathbf{x}_l - \mathbf{z}_k^l \right\|^2 \right) \end{aligned} \tag{4}$$

where $\mathcal{M} \coloneqq \{\{\boldsymbol{\mu}_k^l\}_{l \in \mathcal{N}_k}, \{\boldsymbol{\lambda}_k^l\}_{l \in \mathcal{N}_k}\}_{k=1}^K$ are the Lagrange multipliers associated with (3), and $\rho > 0$ is a penalty parameter.

Solving (3) via the ADMM requires an iterative process that is described in the next subsection.

B. Distributed ADMM Algorithm

To solve the minimization problem (3) in a distributed fashion, the ADMM entails an iterative procedure consisting of three steps at each iteration. In the first step, $\mathcal{L}_\rho$ is minimized with respect to the primal variables $\mathcal{V}$. Then, $\mathcal{L}_\rho$ is minimized with respect to the auxiliary variables $\mathcal{Z}$. In the end, the Lagrange multipliers in $\mathcal{M}$ are updated via dual gradient-ascent iterations [8]. By using the Karush-Kuhn-Tucker conditions of optimality for (3) and setting $\boldsymbol{\lambda}_k(m) = 2 \sum_{l \in \mathcal{N}_k} \boldsymbol{\lambda}_k^l(m)$, it can be shown that the Lagrange multipliers $\{\boldsymbol{\mu}_k^l\}_{l \in \mathcal{N}_k}$ and the auxiliary variables $\mathcal{Z}$ are eliminated [8]. Therefore, the distributed ADMM algorithm reduces to the following iterative updates at the kth agent

$$\mathbf{x}_k(m+1) = \arg\min_{\mathbf{x}_k} \mathcal{L}_\rho(\mathbf{x}_k, \boldsymbol{\lambda}_k(m)) \tag{5}$$

$$\boldsymbol{\lambda}_k(m+1) = \boldsymbol{\lambda}_k(m) + \rho \sum_{l \in \mathcal{N}_k} [\mathbf{x}_k(m+1) - \mathbf{x}_l(m+1)] \tag{6}$$

where m is the iteration index and all initial values $\{\mathbf{x}_k(0)\}_{k \in \mathcal{K}}$, $\{\boldsymbol{\lambda}_k(0)\}_{k \in \mathcal{K}}$ are set to zero. The iterations (5) and (6) can be implemented in a fully distributed manner as they only involve the parameters available within each node's neighborhood.

The objective function of the minimization problem in (5) is non-smooth, which makes it hard to obtain a solution using first-order information. To solve this problem, we employ a zeroth-order method described in the next subsection.

Algorithm 1 D-ZOA

At all agents $k \in \mathcal{K}$, initialize $\mathbf{x}_k(0) = \mathbf{0}$, $\boldsymbol{\lambda}_k(0) = \mathbf{0}$, and locally run
for $m = 1, 2, \ldots$ **do**
  Receive $\mathbf{x}_k(m)$ from neighbors in $\mathcal{N}_k$
  Update $\boldsymbol{\lambda}_k(m+1)$ as in (6)
  Initialize $\mathbf{x}_k^0 = \mathbf{0}$
  for $t = 1, 2, \ldots, T$ **do**
    Draw independent $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_P)$
    Set $u_1^t = u_1/t$, $u_2^t = u_1/t^2$ and compute $\mathbf{g}^t$ as in (8)
    Update $\mathbf{x}_k^{t+1}$ as in (9)
  end for
  Update $\mathbf{x}_k(m+1) = \mathbf{x}_k^{T+1}$
end for

C. Zeroth-Order Method

In order to solve (5) utilizing a zeroth-order method, we assume that $\mathcal{L}_\rho(\cdot)$ is closed and Lipschitz-continuous with the Lipschitz constant G. These assumptions are common for zeroth-order optimization, see, e.g., [1], [20].

Subsequently, we employ the two-point stochastic gradient algorithm for general non-smooth functions proposed in [20]. More specifically, we use the stochastic mirror descent method with the proximal function $\|\cdot\|/2$ and the gradient estimator at point $\mathbf{x}_k$ given by

$$G_{\text{ns}}(\mathbf{x}_k; u_1, u_2, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\lambda}_k(m)) = u_2^{-1}[\mathcal{L}_\rho(\mathbf{x}_k + u_1\boldsymbol{\nu}_1 + u_2\boldsymbol{\nu}_2, \boldsymbol{\lambda}_k(m)) - \mathcal{L}_\rho(\mathbf{x}_k + u_1\boldsymbol{\nu}_1, \boldsymbol{\lambda}_k(m))]\boldsymbol{\nu}_2 \tag{7}$$

where $u_1 > 0$ and $u_2 > 0$ are smoothing constants and $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2$ are zero-mean Gaussian random vectors independent of each other with covariance matrix $\mathbf{I}_P$, i.e., $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_P)$.

The two-point stochastic gradient algorithm entails an iterative procedure that consists of three steps at each iteration t. First, independent random vectors $\boldsymbol{\nu}_1^t$ and $\boldsymbol{\nu}_2^t$ are sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I}_P)$. Second, a stochastic gradient $\mathbf{g}^t$ is formed as

$$\mathbf{g}^t = G_{\text{ns}}(\mathbf{x}_k^t; \boldsymbol{\lambda}_k(m), u_1^t, u_2^t, \boldsymbol{\nu}_1^t, \boldsymbol{\nu}_2^t) \tag{8}$$

where $\mathbf{x}_k^t$ is the tth iterate of the two-point stochastic gradient algorithm with the initial point $\mathbf{x}_k = \mathbf{0}$, $\{u_1^t\}_{t=1}^\infty$ and $\{u_2^t\}_{t=1}^\infty$ are two non-increasing sequences of positive parameters such that $u_2^t \leq u_1^t/2$. Finally, $\mathbf{x}_k^{t+1}$ is updated as

$$\mathbf{x}_k^{t+1} = \mathbf{x}_k^t - \alpha(t)\mathbf{g}^t \tag{9}$$

where the time-dependent step-size $\alpha(t)$ is set as $\alpha(t) = (G\sqrt{tP\log(2P)})^{-1}\alpha_0 R$, α_0 is an appropriate initial step-size and R is an upper bound for the distance between a minimizer $\mathbf{x}_k^*$ to (5) and the first iterate $\mathbf{x}_k^1$ as per [20].

Note that no communication among agents is involved throughout the inner loop.

The proposed algorithm, D-ZOA, is summarized in Algorithm 1.

In the next subsection, we show that the D-ZOA produces sequences of local iterates $\mathbf{x}_k(m)$, $k \in \mathcal{K}$, that converge to the global centralized solution $\mathbf{x}^c$ as $m \to \infty$.

D. Convergence Analysis

The convergence of D-ZOA to the centralized solution is established by corroborating that both inner and outer loops of the algorithm converge.

The convergence of the inner loop can be proven following [20, Theorem 2], i.e., it can be shown that there exists a numerical constant c such that, for each T representing a fixed number of iterations of the inner loop, the following inequality holds:

$$\begin{aligned}&\mathbb{E}[\mathcal{L}_\rho(\hat{\mathbf{x}}_k(T)) - \mathcal{L}_\rho(\mathbf{x}_k^*)] \\ &\leq c\frac{RG\sqrt{P}}{\sqrt{T}}\Big[\max\{\alpha_0, \alpha_0^{-1}\}\sqrt{\log(2P)} + \frac{u_1\log(2T)}{\sqrt{T}}\Big]\end{aligned} \tag{10}$$

where $\hat{\mathbf{x}}_k(T) = T^{-1}\sum_{t=1}^T \mathbf{x}_k^t$. In [20], it is shown that $c = 0.5$ whenever $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ are sampled from a normal distribution.

The convergence of the outer loop can be verified by proving the convergence of a fully distributed ADMM with inexact primal updates. For this purpose, the primal variable can be assumed to be a perturbed version of the exact primal update as per [21]. Therefore, $\mathbf{x}_k(m+1)$ can be written as

$$\mathbf{x}_k(m+1) = \bar{\mathbf{x}}_k(m+1) + \boldsymbol{\gamma}_k(m+1) \tag{11}$$

where $\bar{\mathbf{x}}_k(m+1)$ is the exact ADMM primal update and $\boldsymbol{\gamma}_k(m+1)$ is a random variable representing the perturbation of $\bar{\mathbf{x}}_k(m+1)$. Similar to [21], we assume the perturbation to have zero expectation, i.e., $\mathbb{E}[\boldsymbol{\gamma}_k(m+1)] = \mathbf{0}$, $\forall k \in \mathcal{K}$ and for all the ADMM iterations m, and have finite covariance matrix, i.e., $\text{cov}[\boldsymbol{\gamma}_k(m+1)]_{i,j} < \infty$, $\forall k \in \mathcal{K}$, $\forall i, j = 1, \ldots, P$ and for all the ADMM iterations m.

For a clear presentation of the convergence results, we rewrite (3) in the matrix form. By defining $\tilde{\mathbf{x}} \in \mathbb{R}^{KP}$ as a vector concatenating all $\mathbf{x}_k$ and $\tilde{\mathbf{z}} \in \mathbb{R}^{2EP}$ concatenating all $\mathbf{z}_k^l$, (3) can be written as

$$\begin{aligned}\min_{\tilde{\mathbf{x}}, \tilde{\mathbf{z}}} \quad & f(\tilde{\mathbf{x}}) + g(\tilde{\mathbf{z}}) \\ \text{s.t.} \quad & \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{z}} = \mathbf{0}\end{aligned} \tag{12}$$

where $f(\tilde{\mathbf{x}}) = \sum_{k=1}^K f_k(\mathbf{x}_k; \mathcal{X}_k)$, $g(\tilde{\mathbf{z}}) = 0$, $\mathbf{A} = [\mathbf{A}_1; \mathbf{A}_2]$, and $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{2EP \times KP}$ are both composed of $2E \times K$ blocks of $P \times P$ matrices. If $(k,l) \in \mathcal{E}$ and $\mathbf{z}_k^l$ is the qth block of $\tilde{\mathbf{z}}$, then the (q,k)th block of $\mathbf{A}_1$ and the (q,l)th block of $\mathbf{A}_2$ are identity matrices $\mathbf{I}_P$. Otherwise, the corresponding blocks are $P \times P$ zero matrices $\mathbf{0}_P$. Furthermore, we have $\mathbf{B} = [-\mathbf{I}_{2EP}; -\mathbf{I}_{2EP}]$. We define the matrices $\mathbf{M}_+ = \mathbf{A}_1^\mathsf{T} + \mathbf{A}_2^\mathsf{T}$ and $\mathbf{M}_- = \mathbf{A}_1^\mathsf{T} - \mathbf{A}_2^\mathsf{T}$, $\mathbf{L}_+ = 0.5\mathbf{M}_+\mathbf{M}_+^\mathsf{T}$, $\mathbf{L}_- = 0.5\mathbf{M}_-\mathbf{M}_-^\mathsf{T}$, $\mathbf{Q} = \sqrt{0.5\mathbf{L}_-}$ and $\boldsymbol{\gamma}(m+1) \in \mathbb{R}^{KP}$ as the vector concatenating all $\boldsymbol{\gamma}_k(m+1)$.

We construct the auxiliary sequence $\mathbf{r}(m) = \sum_{s=0}^m \mathbf{Q}\tilde{\mathbf{x}}(s)$ and define the auxiliary vector $\mathbf{q}(m)$ and the auxiliary matrix $\mathbf{G}$ as

$$\mathbf{q}(m) = \begin{bmatrix}\mathbf{r}(m) \\ \tilde{\mathbf{x}}(m)\end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix}\rho\mathbf{I}_P & \mathbf{0}_{P\times P} \\ \mathbf{0}_{P\times P} & \rho\frac{\mathbf{L}_+}{2}\end{bmatrix}. \tag{13}$$

The convergence results of [21] can now be adapted to D-ZOA as per the following theorem.

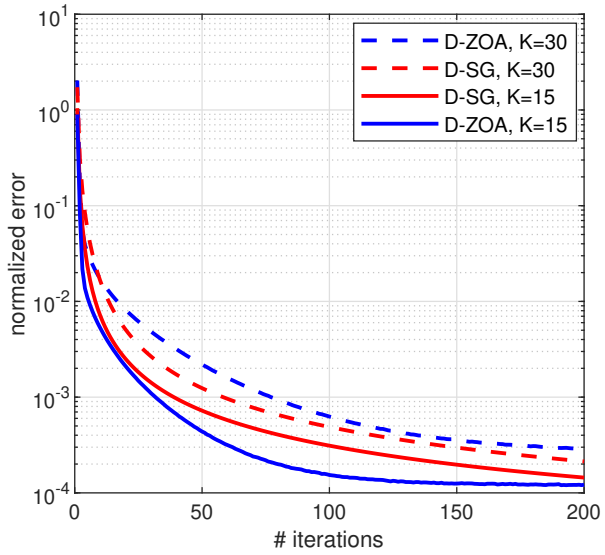

Fig. 1. Normalized error of D-ZOA and D-SG for generalized lasso with $P = 10$, $\rho = 3$ and two different values of K.

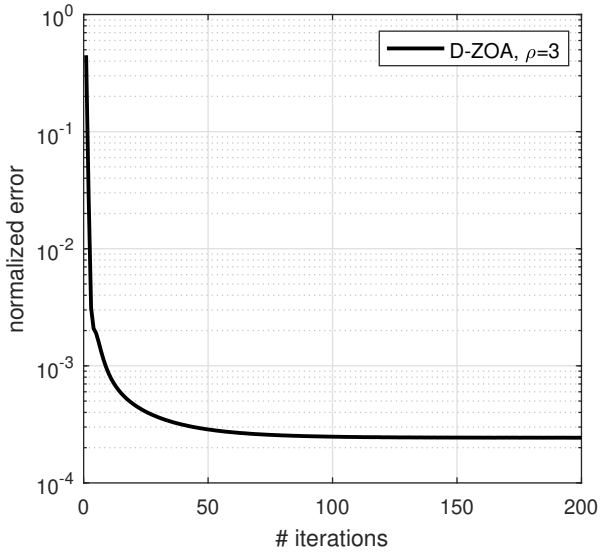

Fig. 2. Normalized error of D-ZOA for RRR with $P = 5$, $S = 4$, $\rho = 3$ and $K = 10$.

Theorem 1. *If $f(\cdot)$ is convex, then, for any fixed number of iterations N of the outer loop, we have*

$$\begin{aligned}&\mathbb{E}[f(\hat{\mathbf{x}}^N) - f(\tilde{\mathbf{x}}^*)]\\ \leq&\frac{\|\mathbf{q}(0)-\mathbf{q}\|^2_{\mathbf{G}}}{N} + \frac{\rho\lambda^2_{max}(\mathbf{L}_+)\sum_{m=0}^{N-1} tr\,(cov[\boldsymbol{\gamma}(m)])}{2N\lambda_{min}(\mathbf{L}_-)}\end{aligned} \tag{14}$$

where the expectation is taken with respect to the perturbation, $\tilde{\mathbf{x}}^*$ is the optimal solution of (12) *and $\hat{\mathbf{x}}^N = \frac{1}{N}\sum_{m=0}^{N-1}\tilde{\mathbf{x}}(m+1)$.*

Proof. Since $\mathbb{E}[\boldsymbol{\gamma}_k(m)] = \mathbf{0}$ and $\text{cov}[\boldsymbol{\gamma}_k(m)]_{i,j} < \infty$, $\forall k \in \mathcal{K}$, $\forall i, j = 1, \ldots, P$ and for all the ADMM iterations m, proof follows from [21, Lemma 6] and [21, Theorem 5]. □

IV. Simulations

The D-ZOA algorithm is tested on a multi-agent network with a random topology, where each agent is linked to three other agents on average. To benchmark D-ZOA with existing solutions, we consider a distributed version of the generalized lasso [15] that can be solved with subgradient methods [2]. Furthermore, we consider a distributed version of the reduced-rank regression (RRR) problem where the objective function is least squares with nuclear norm regularization [8]. Nuclear norm is a non-smooth function that is used as a convex surrogate for the rank. Calculating any subgradient of the nuclear norm function is impractical. RRR has applications in robust PCA [22], low-rank matrix decomposition [23], matrix completion [24], etc.

The network-wide observations are represented as an observation matrix $\mathbf{D} \in \mathbb{R}^{M\times P}$ and a response matrix $\mathbf{H} \in \mathbb{R}^{M\times S}$, where M is the number of data samples and P is the number of features in each sample. The matrix $\mathbf{D}$ consists of K submatrices $\mathbf{D}_k$, i.e., $\mathbf{D} = [\mathbf{D}_1^\mathsf{T}, \mathbf{D}_2^\mathsf{T}, \ldots, \mathbf{D}_K^\mathsf{T}]^\mathsf{T}$, and the matrix $\mathbf{H}$ of K submatrices $\mathbf{H}_k$, i.e., $\mathbf{H} = \left[\mathbf{H}_1^\mathsf{T}, \mathbf{H}_2^\mathsf{T}, \ldots, \mathbf{H}_K^\mathsf{T}\right]^\mathsf{T}$, as the data are distributed among the agents and each agent k holds its respective $\mathbf{D}_k \in \mathbb{R}^{M_k\times P}$ and $\mathbf{H}_k \in \mathbb{R}^{M_k\times S}$ where $\sum_{k=1}^K M_k = M$. The parameter matrix that establishes a linear regression between $\mathbf{D}$ and $\mathbf{H}$ is $\mathbf{X} \in \mathbb{R}^{P\times S}$. In the generalized lasso, $S = 1$ and, hence, $\mathbf{H}$ is the vector $\mathbf{h} \in \mathbb{R}^M$ and $\mathbf{X}$ becomes $\mathbf{x} \in \mathbb{R}^P$. In the centralized approach, a generalized lasso estimate of $\mathbf{x}$ is given by

$$\mathbf{x}^c = \arg\min_{\mathbf{x}}\{\|\mathbf{D}\mathbf{x} - \mathbf{b}\|^2 + \eta\,\|\mathbf{F}\mathbf{x}\|_1\} \tag{15}$$

where $\eta > 0$ is a regularization parameter and $\mathbf{F}$ is an arbitrary matrix. An RRR estimate of $\mathbf{X}$ is also given by

$$\mathbf{X}^c = \arg\min_{\mathbf{X}}\{\|\mathbf{D}\mathbf{X} - \mathbf{H}\|^2 + \eta_*\,\|\mathbf{X}\|_*\} \tag{16}$$

where $\eta_* > 0$ is a rank-controlling parameter. In the distributed setting, we solve problem (2) with

$$f_k(\mathbf{x}_k; \mathcal{X}_k) = \|\mathbf{D}_k\mathbf{x}_k - \mathbf{h}_k\|^2 + \frac{\eta}{K}\,\|\mathbf{F}\mathbf{x}_k\|_1 \tag{17}$$

for the generalized lasso case and with

$$f_k(\mathbf{X}_k; \mathcal{X}_k) = \|\mathbf{D}_k\mathbf{X}_k - \mathbf{H}_k\|^2 + \frac{\eta}{K}\,\|\mathbf{X}_k\|_* \tag{18}$$

for the RRR case. For each agent $k \in \mathcal{K}$, we create a $10P \times P$ local observation matrix $\mathbf{D}_k$ whose entries are independent identically distributed zero-mean unit-variance Gaussian random variables. The response vector $\mathbf{h}$ is obtained as

$$\mathbf{h} = \mathbf{D}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\beta} \in \mathbb{R}^P$ and $\boldsymbol{\epsilon} \in \mathbb{R}^M$ are chosen as random vector with distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_P)$ and $\mathcal{N}(\mathbf{0}, 0.1\mathbf{I}_M)$. The response matrix $\mathbf{H}$ is obtained as

$$\mathbf{H} = \mathbf{D}\boldsymbol{\Phi} + \boldsymbol{\Psi}$$

where $\mathbf{\Phi} \in \mathbb{R}^{P\times S}$ and $\mathbf{\Psi} \in \mathbb{R}^{M\times S}$ are random matrices with matrix normal distributions $\mathcal{MN}(\mathbf{0}_{P\times S}, \mathbf{I}_P, \mathbf{I}_S)$ and $\mathcal{MN}(\mathbf{0}_{M\times S}, 0.1\mathbf{I}_M, 0.1\mathbf{I}_S)$, respectively. The regularization parameter η is set to $0.01\left\|\mathbf{D}^{\mathsf{T}}\mathbf{b}\right\|_{\infty}$ and η_* is set to $0.01\left\|(\mathbf{I}_S \otimes \mathbf{D})^{\mathsf{T}}\text{vec}(\mathbf{H})\right\|_{\infty}$ as in [15]. The number of iterations of the ADMM outer loop is set to 200. For the inner loop, the number of iterations is set to 1000, the smoothing constant u_1 is set to 1 and the convergence in mean is achieved by averaging the outputs of 10 inner loops. Performance of D-ZOA is evaluated using the normalized error between the centralized solutions $\mathbf{x}^c$ as per (15) or $\mathbf{X}^c$ as per (16) and the local estimates. It is defined as $\sum_{k=1}^{K}\|\mathbf{x}_k - \mathbf{x}^c\|^2/\|\mathbf{x}^c\|^2$ for generalized lasso and as $\sum_{k=1}^{K}\|\mathbf{X}_k - \mathbf{X}^c\|_F^2/\|\mathbf{X}^c\|_F^2$ for RRR, where $\mathbf{x}_k$ and $\mathbf{X}_k$ denote the local estimates at agent k. The centralized solutions $\mathbf{x}^c$ and $\mathbf{X}^c$ are computed using the convex optimization toolbox CVX [25]. Results are obtained by averaging over 100 independent trials.

Figs. 1-2 show the performance of D-ZOA for the generalized lasso and the RRR scenarios, respectively. In Fig. 1, we plot the normalized error versus the outer loop iteration index for D-ZOA and a subgradient-based distributed algorithm, called D-SG and proposed in [2]. We observe that, for $P = 10$ and $\rho = 3$, D-ZOA has similar performance to D-SG both when the network consists of 15 and 30 agents. Fig. 2 shows that D-ZOA converges to the centralized solution of the considered RRR problem for $P = 5$, $S = 4$, $K = 10$ and $\rho = 3$.

V. Conclusion

We developed a new consensus-based algorithm for solving a distributed optimization problem with a non-smooth convex objective. We recast the original problem into an equivalent constrained optimization problem whose structure is suitable for distributed implementation via ADMM. We employed a zeroth-order method, known as the two-point stochastic gradient algorithm, to minimize the augmented Lagrangian in the primal update step. Compared to existing algorithms for non-smooth optimization, D-ZOA is fully-distributed and does not require the computation of subgradients, nor proximal operators which may be difficult to derive in some scenarios. D-ZOA only requires the computation of objective function values. The convergence of D-ZOA to the centralized solution was verified through theoretical analysis and simulations.

References

[1] D. Hajinezhad, M. Hong, and A. Garcia, "ZONE: Zeroth-order nonconvex multiagent optimization over networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 3995–4010, Oct. 2019.

[2] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[3] C. Gratton, N. K. D. Venkategowda, R. Arablouei, and S. Werner, "Consensus-based distributed total least-squares estimation using parametric semidefinite programming," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2019, pp. 5227–5231.

[4] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[5] C. Gratton, N. K. D. Venkategowda, R. Arablouei, and S. Werner, "Distributed ridge regression with feature partitioning," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Oct. 2018.

[6] J. Akhtar and K. Rajawat, "Distributed sequential estimation in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 86–100, Jan. 2018.

[7] N. K. D. Venkategowda and S. Werner, "Privacy-preserving distributed precoder design for decentralized estimation," in *Proc. IEEE Global Conference on Signal and Information Processing*, Nov. 2018.

[8] G. B. Giannakis, Q. Ling, G. Mateos, and I. D. Schizas, *Splitting Methods in Communication, Imaging, Science, and Engineering*, ser. Scientific Computation, R. Glowinski, S. J. Osher, and W. Yin, Eds. Cham: Springer International Publishing, 2016.

[9] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.

[10] A. Nedic and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, Dec. 2016.

[11] ——, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.

[12] A. Nedic, A. Ozdaglar, and P. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[13] E. Ghadimi, I. Shames, and M. Johansson, "Multi-step gradient methods for networked optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5417–5429, Nov. 2013.

[14] X. Wu and J. Lu, "Improved convergence rates of P-EXTRA for non-smooth distributed optimization," in *IEEE International Conference on Control and Automation*, Jul. 2019, pp. 55–60.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2010.

[16] D. Yuan, D. W. C. Ho, and S. Xu, "Zeroth-order method for distributed optimization with approximate projections," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 284–294, Feb. 2016.

[17] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," in *Proc. 23rd Annual Conference on Learning Theory*, Jun. 2010, pp. 28–40.

[18] J. C. Spall, *Introduction to Stochastic Search and Optimization*. Wiley, 2003.

[19] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop on Artificial Intelligence and Security*, Nov. 2017, pp. 15–26.

[20] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: the power of two function evaluations," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2788–2806, May 2015.

[21] J. Ding, Y. Gong, M. Pan, and Z. Han, "Optimal differentially private ADMM for distributed machine learning," 2019. [Online]. Available: http://arxiv.org/abs/1901.02094

[22] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, Jun. 2011.

[23] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, pp. 572–596, 2011.

[24] M. Mardani, G. Mateos, and G. B. Giannakis, "Decentralized sparsity-regularized rank minimization: algorithms and applications," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5374–5388, Nov. 2013.

[25] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, 2014.