Distributed Learning over Networks with Non-Smooth Regularizers and Feature Partitioning

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

***Abstract*—We develop a new algorithm for distributed learning with non-smooth regularizers and feature partitioning. To this end, we transform the underlying optimization problem into a suitable dual form and solve it using the alternating direction method of multipliers. The proposed algorithm is fully-distributed and does not require the conjugate function of any non-smooth regularizer function, which may be unfeasible or computationally inefficient to acquire. Numerical experiments demonstrate the effectiveness of the proposed algorithm.**

I. Introduction

An important issue associated with distributed learning is how the data is distributed among the agents. Horizontal partitioning of data refers to when subsets of data samples with a common set of features are distributed over the network. Examples of learning with horizontal partitioning of data can be found in [1]–[4]. However, many regression or classification problems encountered in machine learning deal with heterogeneous data that do not contain common features. These problems lead to the so-called feature (column) partitioning of the data where subsets of features of all data samples are distributed over the network agents. Distributed learning problems with feature partitioning also arise in several signal processing applications, e.g., bioinformatics, multi-view learning, and dictionary learning, as mentioned in [5], [6].

There have been several attempts to solve learning problems with feature partitioning of data, e.g., in [5]–[19]. However, the algorithms in [8], [9] can only be used to solve the basis pursuit and lasso problems, respectively, while the work in [10] is based on assuming an appropriate coloring scheme of the network and cannot be extended to a general graph labeling. The algorithms developed in [6], [11], [12] are based on the diffusion strategy. In contrast, the approaches in [5], [13] are based on the consensus strategy. However, [5] is not fully distributed since the consensus constraints are imposed globally across the entire network rather than being applied locally within each agent's neighborhood. Although the algorithm in [13] is fully distributed, it assumes a specific structure for the objective function and is only suitable for ridge regression. The works of [14]–[17] consider distributed agent-specific estimation. However, the objective functions considered in these works are smooth. The authors of [18] propose a coordinate-descent-based algorithm with an inexact update to reduce communication costs for feature-partitioned distributed learning. In [19], an asynchronous stochastic gradient-descent algorithm was developed for distributed learning with feature partitioning of data. However, none of the above-mentioned algorithms consider distributed problems with general non-smooth regularization and arbitrary graphs.

In this paper, we develop a new fully-distributed algorithm for distributed learning with non-smooth regularizers and feature partitioning of data. We consider a general regularized learning problem whose cost function cannot be written as the sum of the local agent-specific cost functions, i.e., it is not separable. To achieve separability, we formulate the dual problem associated with the underlying convex optimization problem and exploit its favorable structure that, unlike the original problem, allows us to solve it by utilizing the alternating direction method of multipliers (ADMM). By utilizing the dual of the optimization problem associated with the ADMM primal variable update step, we devise a new strategy that does not require any conjugate function of the non-smooth regularizers, which may be infeasible or hard to obtain in some scenarios. The proposed algorithm is fully-distributed as every agent communicates only with its neighboring agents and no central coordinator is needed. Our simulation results show that the proposed algorithm converges in various scenarios.

Notations: The operators $(\cdot)^\mathsf{T}$ and $\mathrm{tr}(\cdot)$ denote transpose and trace of a matrix, respectively. $\|\cdot\|$ represents the Euclidean norm of its vector argument. $\mathbf{I}_n$ is an identity matrix of size n, $\mathbf{0}_n$ is an $n \times 1$ vector with all zeros, $\mathbf{0}_{n\times p} = \mathbf{0}_n\mathbf{0}_p^\mathsf{T}$, and $|\cdot|$ denotes the cardinality if its argument is a set. For a function f, f^* denotes the conjugate function of f.

II. System Model

We consider a network with $N \in \mathbb{N}$ agents and $E \in \mathbb{N}$ edges that is modeled as an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with the set of vertices $\mathcal{V} = \{1, \ldots, N\}$ corresponding to the agents and the set of edges $\mathcal{E}$ representing the bidirectional communication links between the pairs of agents. Agent $i \in \mathcal{V}$ communicates only with its neighbors specified by the set $\mathcal{V}_i$.

Due to feature partitioning, the data of each agent i resides in the matrix $\mathbf{A}_i \in \mathbb{R}^{M\times P_i}$ and the response vector $\mathbf{b} \in \mathbb{R}^{M\times 1}$ where M is the number of data samples and P_i the number of features in each sample at agent i. The feature vector at agent i that relates $\mathbf{A}_i$ and $\mathbf{b}$ is denoted by $\mathbf{x}_i \in \mathbb{R}^{P_i\times 1}$.

We consider a regularized learning problem of form

$$\min_{\{\mathbf{x}_i\}} \quad f\left(\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i - \mathbf{b}\right) + \sum_{i=1}^N r_i(\mathbf{x}_i) \tag{1}$$

where $f(\cdot)$ is the global cost function and $r_i(\cdot)$, $i = 1, \ldots, N$, are the agent-specific regularizer functions. The learning problem (1) pertains to several applications in machine learning, e.g., regression over distributed features [5], clustering in graphs [20], smart grid control [21], dictionary learning [22], and network utility maximization [23]. In this work, we consider learning problems where functions $r_i(\cdot)$, $i = 1, \ldots, N$, are convex, proper, and lower semi-continuous but not necessarily smooth and $f(\cdot) = \|\cdot\|^2$. In the next section, we solve (1) in a distributed manner, where each agent communicates only with its neighbors.

III. Distributed Algorithm for Learning with Feature Partitioning

First, we present the reformulation of the considered non-separable problem into a dual form that is separable and can be solved in a fully-distributed fashion via the ADMM. Then, we describe the new strategy that allows us to employ the ADMM without computing any conjugate function of the non-smooth regularizers explicitly.

A. Distributed ADMM for the Dual Problem

To develop a distributed solution, we introduce the auxiliary variables $\{\mathbf{z}_i\}_{i=1}^N$ and recast (1) as

$$\begin{aligned} \min_{\{\mathbf{x}_i, \mathbf{z}_i\}} \quad & f\left(\sum_{i=1}^N \mathbf{z}_i - \mathbf{b}\right) + \sum_{i=1}^N r_i(\mathbf{x}_i) \\ \text{s. t.} \quad & \mathbf{A}_i\mathbf{x}_i = \mathbf{z}_i, \quad i = 1, \ldots, N. \end{aligned} \tag{2}$$

The objective function in (2) is not separable among the agents. Therefore, we consider the dual problem of (2). For this purpose, we associate the Lagrange multipliers $\{\boldsymbol{\mu}_i\}_{i=1}^N$ with the equality constraints in (2) and form the Lagrangian function $\mathcal{L}(\{\mathbf{x}_i\}, \{\mathbf{z}_i\}, \{\boldsymbol{\mu}_i\})$. The dual function for problem (2) is given by

$$\begin{aligned} d(\{\boldsymbol{\mu}_i\}) &= \inf_{\{\mathbf{x}_i, \mathbf{z}_i\}} \mathcal{L}(\{\mathbf{x}_i\}, \{\mathbf{z}_i\}, \{\boldsymbol{\mu}_i\}) \\ &= -\sum_{i=1}^N r_i^*(-\mathbf{A}_i^\mathsf{T}\boldsymbol{\mu}_i) + \inf_{\mathbf{z}_i}\Big\{ f(\sum_{i=1}^N \mathbf{z}_i - \mathbf{b}) - \sum_{i=1}^N \boldsymbol{\mu}_i^\mathsf{T}\mathbf{z}_i \Big\} \end{aligned} \tag{3}$$

where r_i^* is the conjugate function of r defined as

$$r_i^*(\mathbf{y}) = \sup_{\mathbf{x}} \ \mathbf{y}^T\mathbf{x} - r_i(\mathbf{x}).$$

Next, for the second infimum in (3), introducing

$$\mathbf{z} = \sum_{i=1}^N \mathbf{z}_i$$

and its corresponding dual variable $\boldsymbol{\lambda}$, and using the duality theory, an alternate form of the dual function (3) is obtained as

$$\tilde{d}(\{\boldsymbol{\mu}_i\}, \boldsymbol{\lambda}) = \begin{cases} -\tilde{f}^*(\boldsymbol{\lambda}) - \sum_{i=1}^N r_i^*(-\mathbf{A}_i^\mathsf{T}\boldsymbol{\mu}_i), & \boldsymbol{\lambda} = \boldsymbol{\mu}_i, \forall i \in \mathcal{V} \\ -\infty, & \text{otherwise} \end{cases} \tag{4}$$

where

$$\tilde{f}^*(\boldsymbol{\lambda}) = f^*(\boldsymbol{\lambda}) + \boldsymbol{\lambda}^\mathsf{T}\mathbf{b}.$$

Eliminating the redundant variable $\boldsymbol{\lambda}$, the dual problem for (2) can be expressed as

$$\begin{aligned} \max_{\{\boldsymbol{\mu}_i\}} \quad & -\frac{1}{N}\sum_{i=1}^N \tilde{f}^*(\boldsymbol{\mu}_i) - \sum_{i=1}^N r_i^*(-\mathbf{A}_i^\mathsf{T}\boldsymbol{\mu}_i) \\ \text{s. t.} \quad & \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \cdots = \boldsymbol{\mu}_N. \end{aligned} \tag{5}$$

To solve (5) in a distributed fashion, we employ the ADMM [24]. First, we recast (5) as a constrained minimization problem by imposing consensus constraints across each agent's neighborhood $\mathcal{V}_i$ as follows

$$\begin{aligned} \min_{\{\boldsymbol{\mu}_i\}, \{\mathbf{u}_i^j\}} \quad & \frac{1}{N}\sum_{i=1}^N \tilde{f}^*(\boldsymbol{\mu}_i) + \sum_{i=1}^N r_i^*(-\mathbf{A}_i^\mathsf{T}\boldsymbol{\mu}_i) \\ \text{s. t.} \quad & \boldsymbol{\mu}_i = \mathbf{u}_i^j, \quad \boldsymbol{\mu}_j = \mathbf{u}_i^j, \ j \in \mathcal{V}_i, \ i = 1, \ldots, N. \end{aligned} \tag{6}$$

To facilitate a fully-distributed solution, we decouple the constraints in (5) by introducing the auxiliary variables $\{\mathbf{u}_i^j\}_{j\in\mathcal{V}_i}$. Then, we generate the relevant augmented Lagrangian function by associating the Lagrange multipliers $\{\bar{\mathbf{v}}_i^j\}_{j\in\mathcal{V}_i}$, $\{\tilde{\mathbf{v}}_i^j\}_{j\in\mathcal{V}_i}$ with the consensus constraints. In [24], it is shown that, by setting

$$\mathbf{v}_i^{(k)} = 2\sum_{j\in\mathcal{V}_i} (\bar{\mathbf{v}}_i^j)^{(k)},$$

the Lagrange multipliers $\{\tilde{\mathbf{v}}_i^j\}_{j\in\mathcal{V}_i}$ and the auxiliary variables $\{\mathbf{u}_i^j\}_{j\in\mathcal{V}_i}$ are eliminated and the ADMM reduces to an iterative procedure with two steps at each iteration as

$$\begin{aligned} \boldsymbol{\mu}_i^{(k)} =& \arg\min_{\boldsymbol{\mu}_i} \Big\{ \frac{1}{N} f^*(\boldsymbol{\mu}_i) + \frac{1}{N}\boldsymbol{\mu}_i^\mathsf{T}\mathbf{b} + r_i^*(-\mathbf{A}_i^\mathsf{T}\boldsymbol{\mu}_i) \\ & + \boldsymbol{\mu}_i^\mathsf{T}\mathbf{v}_i^{(k-1)} + \rho\sum_{j\in\mathcal{V}_i} \Big\| \boldsymbol{\mu}_i - \frac{\boldsymbol{\mu}_i^{(k-1)} + \boldsymbol{\mu}_j^{(k-1)}}{2} \Big\|^2 \Big\}, \end{aligned} \tag{7}$$

$$\mathbf{v}_i^{(k)} = \mathbf{v}_i^{(k-1)} + \rho\sum_{j\in\mathcal{V}_i} (\boldsymbol{\mu}_i^{(k)} - \boldsymbol{\mu}_j^{(k)}). \tag{8}$$

where $\rho > 0$ is the penalty parameter.

Since $r_i(\cdot)$, $i = 1, \ldots, N$, are non-smooth, the minimization problem in (7) can be solved by employing appropriate subgradients or proximal operators [25], [26]. However, computing the conjugate function of the regularizers in (7) may be hard. To overcome this challenge, in the next subsection, we describe a new procedure that does not require the explicit calculation of any conjugate function.

B. ADMM with no Conjugate Function

In order to solve the problem in (7), we need to calculate the conjugate function of r_i^*. This can be difficult, especially for non-smooth functions. We exploit the Fenchel-Moreau theorem to eliminate the computation of conjugate function.

To that end, the problem in (7) can be restated as

$$\begin{aligned} \min_{\{\boldsymbol{\mu}_i,\boldsymbol{\nu}_i\}} \quad & \frac{f^*(\boldsymbol{\mu}_i)+\boldsymbol{\mu}_i^\mathsf{T}\mathbf{b}}{N} + r_i^*(\boldsymbol{\nu}_i) + \boldsymbol{\mu}_i^\mathsf{T}\mathbf{c}_i^{(k-1)} + \bar{\rho}_i \left\|\boldsymbol{\mu}_i\right\|_2^2 \\ \text{s. t.} \quad & \mathbf{A}_i^\mathsf{T}\boldsymbol{\mu}_i + \boldsymbol{\nu}_i = \mathbf{0} \end{aligned} \tag{9}$$

where

$$\mathbf{c}_i^{(k-1)} = \mathbf{v}_i^{(k-1)} - \rho|\mathcal{V}_i|\boldsymbol{\mu}_i^{(k-1)} - \rho\sum_{j\in\mathcal{V}_i}\boldsymbol{\mu}_j^{(k-1)}$$

and $\bar{\rho}_i = \rho|\mathcal{V}_i|$. The Lagrangian function for (9) is

$$\begin{aligned} \mathcal{L}(\boldsymbol{\mu}_i,\boldsymbol{\nu}_i,\boldsymbol{\theta}_i) =& \frac{f^*(\boldsymbol{\mu}_i)+\boldsymbol{\mu}_i^\mathsf{T}\mathbf{b}}{N} + r_i^*(\boldsymbol{\nu}_i) + \boldsymbol{\mu}_i^\mathsf{T}\mathbf{c}_i^{(k-1)} \\ & + \bar{\rho}_i \left\|\boldsymbol{\mu}_i\right\|_2^2 + \boldsymbol{\theta}_i^\mathsf{T}(\mathbf{A}_i^\mathsf{T}\boldsymbol{\mu}_i + \boldsymbol{\nu}_i) \end{aligned} \tag{10}$$

where $\boldsymbol{\theta}_i$ is the Lagrange multiplier vector associated with the constraint in (9). Hence, the dual function for the objective in (9) can be expressed as

$$\begin{aligned} &\delta(\boldsymbol{\theta}_i) = \inf_{\{\boldsymbol{\mu}_i,\boldsymbol{\nu}_i\}} \mathcal{L}(\boldsymbol{\mu}_i,\boldsymbol{\nu}_i,\boldsymbol{\theta}_i) \\ &= \inf_{\boldsymbol{\nu}_i}\{r_i^*(\boldsymbol{\nu}_i) + \boldsymbol{\theta}_i^\mathsf{T}\boldsymbol{\nu}_i\} \\ &+ \inf_{\boldsymbol{\mu}_i}\Big\{\frac{f^*(\boldsymbol{\mu}_i)+\boldsymbol{\mu}_i^\mathsf{T}\mathbf{b}}{N} + (\mathbf{c}_i^{(k-1)} + \mathbf{A}_i\boldsymbol{\theta}_i)^\mathsf{T}\boldsymbol{\mu}_i + \bar{\rho}_i\left\|\boldsymbol{\mu}_i\right\|^2\Big\} \\ &= -r_i^{**}(-\boldsymbol{\theta}_i) \\ &+ \inf_{\boldsymbol{\mu}_i}\Big\{\frac{f^*(\boldsymbol{\mu}_i)}{N} + \left(\mathbf{c}_i^{(k-1)} + \mathbf{A}_i\boldsymbol{\theta}_i + \frac{\mathbf{b}}{N}\right)^\mathsf{T}\boldsymbol{\mu}_i + \bar{\rho}_i\left\|\boldsymbol{\mu}_i\right\|^2\Big\} \end{aligned} \tag{11}$$

where the last equality follows from the definition of conjugate function.

For $f(\cdot) = \|\cdot\|^2$, the conjugate function is given by $f^*(\boldsymbol{\mu}_i) = \|\boldsymbol{\mu}_i\|^2/4$. Thus, the optimal value of second infimum of the dual function in (11) is

$$\frac{-1}{4\rho|\mathcal{V}_i| + \frac{1}{N}}\left\|\mathbf{A}_i\boldsymbol{\theta}_i + \mathbf{c}_i^{(k-1)} + \frac{\mathbf{b}}{N}\right\|^2$$

and the infimum is attained at the optimal point

$$\boldsymbol{\mu}_i^o = \frac{-2}{4\rho|\mathcal{V}_i| + \frac{1}{N}}\left(\mathbf{A}_i\boldsymbol{\theta}_i^o + \mathbf{c}_i^{(k-1)} + \frac{\mathbf{b}}{N}\right) \tag{12}$$

where $\boldsymbol{\theta}_i^o = \arg\max_{\boldsymbol{\theta}_i}\delta(\boldsymbol{\theta}_i)$. Since $r_i(\cdot)$ is convex, proper, and lower semi-continuous, we have $r_i^{**} = r_i$ due to the Fenchel-Moreau theorem [27]. Therefore, the dual function is given by

$$\delta(\boldsymbol{\theta}_i) = -r_i(-\boldsymbol{\theta}_i) - \frac{1}{4\rho|\mathcal{V}_i| + \frac{1}{N}}\left\|\mathbf{A}_i\boldsymbol{\theta}_i + \mathbf{c}_i^{(k-1)} + \frac{\mathbf{b}}{N}\right\|^2. \tag{13}$$

Algorithm 1 Proposed algorithm for feature-partitioned distributed learning

At all agents $i \in \mathcal{V}$, initialize $\boldsymbol{\mu}_i^{(0)} = \mathbf{0}$, $\mathbf{v}_i^{(0)} = \mathbf{0}$, and locally run:
for $k = 1, 2, \ldots, K$ **do**
  Update $\boldsymbol{\theta}_i^{(k)}$ via (14).
  Update the dual variables $\boldsymbol{\mu}_i^{(k)}$ via (15).
  Share $\boldsymbol{\mu}_i^{(k)}$ with the neighbors in $\mathcal{V}_i$.
  Update the Lagrange multipliers $\mathbf{v}_i^{(k)}$ via (16).
  Update the auxiliary variables $\mathbf{c}_i^{(k)}$ via (17).
end for

Using (12) and (13), the ADMM steps in (7) and (8) can be equivalently expressed as

$$\begin{aligned} \boldsymbol{\theta}_i^{(k)} =& \arg\min_{\boldsymbol{\theta}_i}\Big\{r_i(-\boldsymbol{\theta}_i) \\ & + \frac{1}{4\rho|\mathcal{V}_i| + \frac{1}{N}}\left\|\mathbf{A}_i\boldsymbol{\theta}_i + \mathbf{c}_i^{(k-1)} + \frac{\mathbf{b}}{N}\right\|^2\Big\} \end{aligned} \tag{14}$$

$$\boldsymbol{\mu}_i^{(k)} = \frac{-2}{4\rho|\mathcal{V}_i| + \frac{1}{N}}\left(\mathbf{A}_i\boldsymbol{\theta}_i^{(k)} + \mathbf{c}_i^{(k-1)} + \frac{\mathbf{b}}{N}\right) \tag{15}$$

$$\mathbf{v}_i^{(k)} = \mathbf{v}_i^{(k-1)} + \rho\sum_{j\in\mathcal{V}_i}(\boldsymbol{\mu}_i^{(k)} - \boldsymbol{\mu}_j^{(k)}) \tag{16}$$

$$\mathbf{c}_i^{(k)} = \mathbf{v}_i^{(k)} - \rho|\mathcal{V}_i|\boldsymbol{\mu}_i^{(k)} - \rho\sum_{j\in\mathcal{V}_i}\boldsymbol{\mu}_j^{(k)}. \tag{17}$$

The proposed algorithm is summarized in Algorithm 1. Note that the minimization problem in (14) can be solved using standard optimization techniques, or alternatively, subgradient-based algorithms [28]. Regardless of the technique used to solve (14), the proposed algorithm converges according to [28, Section 3.6.2]. Convergence of Algorithm 1 follows from [1, Proposition 2] and [29]. Moreover, due to the strong duality theorem, we have $\boldsymbol{\theta}_i^o = \mathbf{x}_i^o$, i.e., the optimal dual variable $\boldsymbol{\theta}_i^o$ at agent i is the optimal estimate $\mathbf{x}_i^o$ [30].

IV. Simulations

To illustrate the performance of the proposed algorithm, we consider the elastic-net regression problem [31] and benchmark the proposed algorithm against a broadcast-based algorithm for learning with distributed features [5]. The only existing work considering non-smooth distributed learning with feature partitioning over general graphs is [5]. Therefore, we compared our algorithm only with this algorithm to provide a comparison that is as fair as possible. In a centralized setting, the optimal solution $\mathbf{x}^c$ is obtained as

$$\mathbf{x}^c = \arg\min_{\mathbf{x}} \left\|\mathbf{A}\mathbf{x} - \mathbf{b}\right\|_2^2 + \eta_1\left\|\mathbf{x}\right\|_1 + \eta_2\left\|\mathbf{x}\right\|_2^2 \tag{18}$$

where

$$\begin{aligned} \mathbf{A} &= [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N] \\ \mathbf{x} &= \left[\mathbf{x}_1^\mathsf{T}, \mathrm{x}_2^\mathsf{T}, \ldots, \mathbf{x}_N^\mathsf{T}\right]^\mathsf{T}, \end{aligned}$$

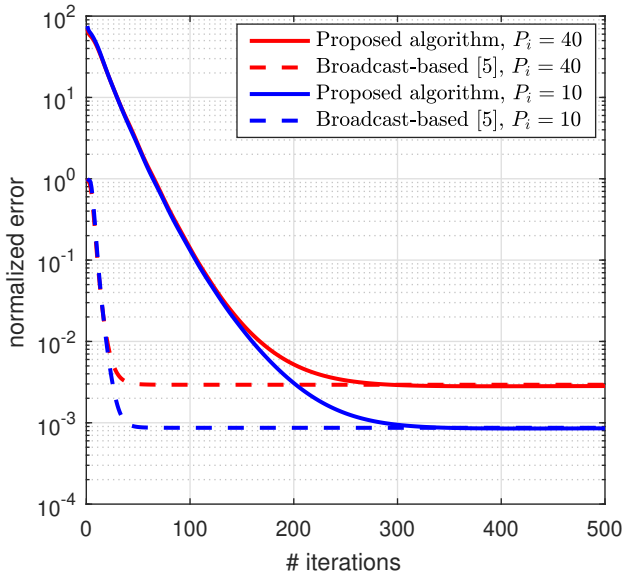

Fig. 1. Normalized error of the proposed algorithm and the broadcast-based algorithm of [5] with $N = 20$ agents and different values of P_i.

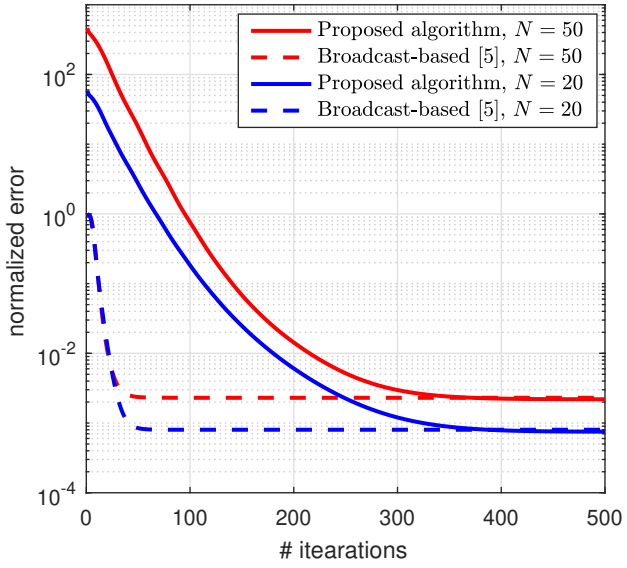

Fig. 2. Normalized error of the proposed algorithm and the broadcast-based algorithm of [5] with $P_i = 10$ and different values of N.

and $\eta_1 \in \mathbb{R}^+$ and $\eta_2 \in \mathbb{R}^+$ are the regularization parameters. In the distributed setting, we solve the problem (1) with

$$f(\mathbf{x}_i) = \left\| \sum_{i=1}^{N} \mathbf{A}_i \mathbf{x}_i - \mathbf{b} \right\|^2,$$
$$r_i(\mathbf{x}_i) = \eta_1 \|\mathbf{x}_i\|_1 + \eta_2 \|\mathbf{x}_i\|^2.$$

We test the proposed algorithm on a multi-agent network with a random topology, where each agent links to three other agents on average. For each agent $i \in \mathcal{V}$, we create a $2P_i \times P_i$ local observation matrix $\mathbf{A}_i$ whose entries are independent identically distributed Gaussian random variables with zero mean and unit variance. The response vector $\mathbf{b}$ is obtained as

$$\mathbf{b} = \mathbf{A}\boldsymbol{\omega} + \boldsymbol{\psi}$$

where $\boldsymbol{\omega} \in \mathbb{R}^P$, $P = \sum_{i=1}^{N} P_i$, and $\boldsymbol{\psi} \in \mathbb{R}^M$ are drawn from the distributions $\mathcal{N}(\mathbf{0}, \mathbf{I}_P)$ and $\mathcal{N}(\mathbf{0}, 0.1\mathbf{I}_M)$, respectively. The regularization parameters are set to $\eta_1 = \eta_2 = 1$ and penalty parameter to $\rho = 1$. The performance of the proposed algorithm is evaluated using the normalized error $\epsilon(k)$ between the centralized solution $\mathbf{x}^c$ as per (18) and the solution from Algorithm 1 at iteration k denoted by

$$\mathbf{x}^d(k) = \left[(\mathbf{x}_1^{(k)})^\mathsf{T}, \ldots, (\mathbf{x}_N^{(k)})^\mathsf{T} \right]^\mathsf{T}.$$

The normalized error is defined as

$$\epsilon(k) = \frac{\left\| \mathbf{x}^d(k) - \mathbf{x}^c \right\|^2}{\left\| \mathbf{x}^c \right\|^2}.$$

The centralized solution $\mathbf{x}^c$ is computed using the optimization toolbox CVX [32]. Results are obtained by averaging over 100 independent trials.

Fig. 1 shows that, for $N = 20$ agents, the proposed algorithm converges when the number of parameters at the ith agent is $P_i = 10$ and $P_i = 40$, $\forall i \in \mathcal{V}$. Fig. 2 shows that the proposed algorithm converges when $P_i = 10$ and the network consists of 20 or 50 agents. The faster convergence of the broadcast-based algorithm of [5] is due to its centralized processing.

V. Conclusion

We developed a fully-distributed algorithm for learning with non-smooth regularization functions under distributed features. We reformulated the underlying problem into an equivalent dual form and used the ADMM to solve it in a distributed fashion without using any conjugate function. To the best of our knowledge, the proposed algorithm is the first of its kind that solves the feature-distributed learning problems with non-smooth regularizer functions over arbitrary graphs while not relying on any conjugate function. We verified the convergence of the proposed algorithm at all agents via simulation results.

References

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

[2] 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.

[3] ——, "Distributed learning with non-smooth objective functions," in *Proc. 28th European Signal Processing Conference*, Jan. 2021, pp. 2180–2184.

[4] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2320–2330, May 2011.

[5] 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.

[6] B. Ying, K. Yuan, and A. H. Sayed, "Supervised learning under distributed features," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 977–992, Feb. 2019.

[7] H. Zheng, S. R. Kulkarni, and H. V. Poor, "Attribute-distributed learning: Models, limits, and algorithms," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 386–398, Jan. 2011.

[8] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1942–1956, Apr. 2012.

[9] C. Manss, D. Shutin, and G. Leus, "Distributed splitting-over-features sparse bayesian learning with alternating direction method of multipliers," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 3654–3658.

[10] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, May 2013.

[11] R. Arablouei, K. Doğançay, S. Werner, and Y.-F. Huang, "Model-distributed solution of regularized least-squares problem over sensor networks," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2015, pp. 3821–3825.

[12] S. A. Alghunaim, M. Yan, and A. H. Sayed, "A multi-agent primal-dual strategy for composite optimization over distributed features," in *Proc. 28th European Signal Processing Conference*, Jan. 2021, pp. 2095–2099.

[13] 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.

[14] J. Szurley, A. Bertrand, and M. Moonen, "Topology-independent distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 1, pp. 130–144, 2017.

[15] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS for clustered multitask networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 5487–5491.

[16] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5382–5397, 2014.

[17] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based LMS for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3448–3460, 2015.

[18] B. Zhang, J. Geng, W. Xu, and L. Lai, "Communication efficient distributed learning with feature partitioned data," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2018, pp. 1–6.

[19] Y. Hu, D. Niu, J. Yang, and S. Zhou, "FDML: A collaborative machine learning framework for distributed features," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2232–2240.

[20] D. Hallac, J. Leskovec, and S. Boyd, "Network lasso: Clustering and optimization in large graphs," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, p. 387–396.

[21] T. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.

[22] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, 2015.

[23] D. P. Palomar and Mung Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[24] 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.

[25] D. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[26] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, p. 127–239, Jan. 2014.

[27] J. M. Borwein and A. S. Lewis, *Convex analysis and nonlinear optimization: theory and examples*. Springer, 2006.

[28] Z. Han, M. Hong, and D. Wang, *Signal processing and networking for big data applications*. Cambridge University Press, 2017.

[29] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, pp. 475–494, Jan. 2001.

[30] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[31] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

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