2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

On Distributed Exact Sparse Linear Regression over Networks

Tu Anh-Nguyen and César A. Uribe

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9993262

Abstract— **In this work, we propose an algorithm for solving exact sparse linear regression problems over a network in a distributed manner. Particularly, we consider the problem where data is stored among different computers or agents that seek to collaboratively find a common regressor with a specified sparsity k, i.e., the L_0-norm is less than or equal to k. Contrary to existing literature that uses L_1 regularization to approximate sparseness, we solve the problem with exact sparsity k. The main novelty in our proposal lies in showing a problem formulation with zero duality gap for which we adopt a dual approach to solve the problem in a decentralized way. This sets a foundational approach for the study of distributed optimization with explicit sparsity constraints. We show theoretically and empirically that, under appropriate assumptions, where each agent solves smaller and local integer programming problems, all agents will eventually reach a consensus on the same sparse optimal regressor.**

I. Introduction

Data regression analysis is a fundamental task in many modern research fields, ranging from natural sciences and engineering to management and social sciences [1]. Linear regression is one of the most popular and well-studied methods to efficiently capture the relations between variables of interest and their predictors [2]. Analyzing the linear regressor is a common practice that yields meaningful interpretations of the data [3]. However, due to the high dimensionality of real-world data, such as RNA sequencing [4], it is a common practice to assume the linear regressor is sparse [5]. A sparse regressor is not only computationally more efficient but also more interpretable compared to a dense solution [6]–[8].

Although sparse linear regression is a well-studied problem on a single machine [6], [9], it remains a challenge when dealing with modern distributed data storage. Distributed data stored and data transfer between agents can be costly or access-controlled due to privacy policies. Thus, it becomes non-trivial to solve the sparse linear regression problem subject to strict communication and information constraints.

In this work, we propose a distributed sparse linear regression algorithm. Formally, given N agents, where each agent $i \in [\![N]\!]$ only has access to its local data $X^i := (x^i_1, \ldots, x^i_{n_i})^T \in \mathbb{R}^{n_i \times p}$ and local observations $Y^i := (y^i_1, \ldots, y^i_{n_i}) \in \mathbb{R}^{n_i}$. Ultimately, we want the group of agents to jointly solve the following optimization problem:

$$\begin{aligned} &\min_{w \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{N} \|Y^i - X^i w\|_2^2 + \frac{1}{\gamma} \|w\|_2^2 \\ &\text{such that } \|w\|_0 \leq k, \end{aligned} \tag{1}$$

where p is the dimension of the regressor. Here, $\gamma > 0$ is a fixed parameter that controls the effect of the Tikhonov regularization term, and $k > 0$ is a predefined number which is interpreted as the number of non-zero coefficient of w needed to model the data $\{(X^i, Y^i)\}_{i \in [\![N]\!]}$. Since data transfer is prohibited in this setting, our proposed algorithm allows agents to exchange their intermediate parameters w^i with their neighbors. We show that the algorithm not only returns a sparse regressor but also guarantees a consensus across all agents.

A good amount of effort has been devoted to studying distributed linear regression over the last decades. In [10], [11], the authors study the problem where data are distributed vertically among units. In the setting where data is distributed among agents instead of features, Dobriban and Sheng [12] study the scheme where each machine solves a linear regression problem locally and then sends the result to a central processing unit for averaging. Mateos et al. [13] developed techniques for obtaining sparsity in distributed linear regression using Lasso. However, Mateos's approach does not guarantee to return the optimal solution of (1). In this work, instead of using Lasso to attain a desired sparsity approximately, we focus on solving (1) to optimality. To the best of our knowledge, we are the first to consider such a problem in a distributed manner.

Most of the work done on sparse linear regression solves (1) heuristically by replacing the combinatorial condition $\|w\|_0 \leq k$ by a L_1-norm constraint [7]. Elastic Net or Lasso is usually favored over solving (1) exactly because of its computational feasibility and scalability. However, they possess innate drawbacks as the L_1-constraint penalizes both large and small coefficients while, in contrast, the L_0-constraint does not, and thus the sparsity pattern is not well recovered [14]. Despite the NP-hardness of (1), Bertsimas et al. [6] have recently developed a cutting plane algorithm for solving (1) in a matter of minutes where the number of data n and the number of features p is in order of $100,000$s. With the ability to solve such a large combinatorial problem, we can compare the performance between Elastic Net and sparse regression. As shown in [6], the solution of (1) is superior in both accuracy and true support recovery. Moreover, it has been shown both empirically and theoretically that the new cutting plane method requires much less data than Elastic

TAN is with the Department of Computational and Applied Mathematics, Rice University, Texas, USA, tan5@rice.edu. CAU is with the Department of Electrical and Computer Engineering, Rice University, Texas, USA, cauribe@rice.edu. This material is based upon work supported by the National Science Foundation under Grant No. 2211815.

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on September 12,2023 at 19:20:05 UTC from IEEE Xplore. Restrictions apply.

Net to attain phase transition - the phenomenon in which the true support of regressors is recovered with enough data with high probability [15]–[17]. Interestingly, in contradiction to the common intuition for the complexity of (1), solving times for (1) drops significantly as the number of data increases [6].

The result in [6] enables the computation of solutions of (1) in high-dimensional regimes. Moreover, sparse linear regression is a more promising candidate in the distributed setup than methods relying on L_1 regularization. Specifically: 1) Sparse linear regression tends to attain higher accuracy than L_1-based methods given the same amount of data, and the performance gap between these is larger when the number of samples is small. This is the case for distributed problems because each agent is not allowed to share data and thus can only process a limited number of data. 2) The phase transition of sparse linear regression occurs sooner than L_1-based methods. Thus sparse linear regression has a good chance to recover the support of the true regressor.

We employ a dual approach to solve exact sparse linear regression in a distributed manner. We first show that, even though the problem we want to solve involves binary variables, we can still achieve zero gap between the primal problem and the Lagrangian dual. A simple gradient ascent algorithm can converge to a sparse regressor. However, each agent must solve a local quadratic integer program in the proposed dual framework at every iteration. As we shall see later, this problem is the sparse linear regression for the local data and observations at each agent plus an additional linear function. To this end, we extend the outer approximation used in [6] for solving sparse regression to solve the local quadratic integer programming problem at every agent. This reformulation of the approximation method proposed in [6] effectively makes the sparse linear regression problem *dual-friendly* in the sense of [18, Definition 2]

The rest of the paper is organized as follows: In Section II, we show that by using the distributed dual framework, we will converge to a sparse regressor. Section III provides an alternative transformation of the local quadratic integer programming problem within an agent so that we can solve it efficiently. In Section IV, we evaluate our distributed algorithm on a synthetic dataset and observe the convergence behavior with a different number of features, various network structures, and different network sizes.

Notation: We define $[\![n]\!] := \{1, 2, \dots, n\}$. Given a graph $\mathcal{G}$, we denote $V(\mathcal{G})$ and $E(\mathcal{G})$ as its vertices and edges set respectively. For a node $i \in V(\mathcal{G})$, we let $N(i) := \{j \in V(\mathcal{G}) | (i,j) \in E(\mathcal{G})\}$ be its neighbor set. For any set S, we use $\|S\|$ to represent the cardinality of S, and $\text{conv}(S)$ as its convex hull. We denote $\mathbb{R}_+ = \{x \in \mathbb{R} | x \geq 0\}$.

II. Algorithms and Results

We construct the distributed algorithm for exact sparse linear regression using the dual approach from [18]. First, we transform (1) to a quadratic mixed-integer program using a big-M formulation [19], which is a traditional technique to model L_0-constraints as linear inequalities with additional integer variables. Without the sparsity constraint $\|w\|_0 \leq k$, the optimization (1) is a minimization of a strictly convex function, which has an unique minimizer w^*. Hence, there exists a real number $M \in \mathbb{R}_+$ such that $\|w^*\|_2 \leq M$. In practice, we do not need to compute the value of M, and we only use M for the argument of the big-M formulation. In addition, we assume that there exists an underlying undirected graph $\mathcal{G}$ that represents which pair of agents can communicate with each other. The graph $\mathcal{G}$ is assumed to be connected, and we denote its Laplacian by $L \in \mathbb{R}^{n \times n}$. With these assumptions, (1) can be rewritten as a quadratic integer programming (QIP) problem:

$$z_{QIP} = \min_{w^i \in \mathbb{R}^p, \forall i \in [\![p]\!]} \sum_{i=1}^{N} \left(\frac{1}{2} \|Y^i - X^i w^i\|_2^2 + \frac{1}{\gamma N} \|w^i\|_2^2 \right)$$

$$\begin{aligned} \text{such that } & Lv^j = 0 && \forall j \in [\![p]\!] && \text{(2a)} \\ & -Ms^i \leq w^i \leq Ms^i && \forall i \in [\![N]\!] && \text{(2b)} \\ & s^i \in S_k^p && \forall i \in [\![N]\!], && \text{(2c)} \end{aligned}$$

where $v^j = (w_j^1, \dots, w_j^N)^T$ denotes the vector consisting of the j-th entries of $w^1, \dots, w^N$, and $S_k^p = \{s \in \{0,1\}^p | \mathbb{1}^T s \leq \ k\}$. In (2), the coupling constraints (2a) assures that every agent has the same regressor. Since the elements of the vector s^i for every $i \in [\![N]\!]$ can only be 0 or 1, w_j^i must be 0 when $s_j^i = 0$ for some $j \in [\![p]\!]$, and takes an arbitrary value otherwise. Thus, the two constraints (2b) and (2c) enforces the sparsity on w^i. The next result shows that we can apply Lagrangian multiplier theory for the coupling constraints and derive strong duality.

Lemma 1 Let $\gamma > 0$, and the Lagrangian function of the mixed-integer optimization Problem (2) be given by

$$\phi(y) = \min_{\substack{s^i \in S_k^p, \\ -Ms_i \leq w^i \leq Ms_i \\ \forall i \in [\![N]\!]}} f(w, y), \quad (3)$$

where

$$f(w,y) := \sum_{i=1}^{N} \left(\frac{1}{2} \|Y^i - X^i w^i\|_2^2 + \frac{1}{\bar{\gamma}} \|w^i\|_2^2 \right) + \sum_{j=1}^{p} \langle y^j, Lv^j \rangle,$$

and the constant $\bar{\gamma} = \gamma N$. Then, $\max_y \phi(y) = z_{QIP}$, where z_{QIP} is the optimal value of (2) [1].

In the big-M formulation (2), we impose the consensus constraint by $Lv^j = 0, \forall j \in [\![p]\!]$. Therefore, we do not need to impose the sparsity condition on every agent. Indeed, we can still derive the same result from Lemma 1 when we only require a subset of agents to solve the exact sparse linear regression. Lemma 1 implies that even though (2) is a quadratic integer programming problem, we still have zero gap between the primal objective and its Lagrangian dual. Hence, instead of minimizing (2) with the hard coupling constraints, we can maximize $\phi(y)$.

[1]Please refer to [20] for the extended version of this paper with detail on the proof

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on September 12,2023 at 19:20:05 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Distributed Exact Sparse Linear Regression

```
1: procedure INITIALIZATION
2:     Each agent initializes its own local multipliers:
           $\psi_L^{i,1} \leftarrow (y_1^i, \ldots, y_p^i)\ \forall i \in [\![N]\!]$
3: procedure DISTRIBUTED SPARSE LINEAR REGRESSION($\{(X^i, Y^i)\}_{i\in[\![N]\!]}$, $G$, $k$, $\gamma$, $T$)
4:     for $t = 1, 2, \ldots, T$ do
5:         Agents receive multipliers from their neighbor
               $\psi_N^{i,t} \leftarrow [\psi_L^{j,t}]_{j\in N(i)}\ \forall i \in [\![N]\!]$
6:         Agents compute their dual multiplier
               $D^{i,t} \leftarrow L_{i,i}\psi_L^{i,t} + \sum_{j\in N(i)} L_{i,j}\psi_{N,j}^{i,t}$
7:         Agents solve their respective local problem:
               $\min_{\|w^i\|_0\le k} \frac{1}{2}\|Y^i - X^i w^i\|_2^2 + \frac{1}{\gamma}\|w^i\|_2^2 + \langle D^{i,t}, w^i\rangle$
           and obtaining local regressor $w^{i,t}$
8:         Agents send and receive new regressor from their neighbor, then update their local multiplier
               $\psi_L^{i,t+1} \leftarrow \psi_L^{i,t} + \alpha_t(L_{i,i}w^{i,t} + \sum_{j\in N(i)} L_{i,j}w^{j,t})$
```

Next, we state some useful properties of the function $\phi(y)$ that will enable us to propose a gradient ascent method for our dual problem.

Proposition 2 The function $\phi(y)$ in (3) is a concave and continuous function, whose gradient is given by

$$\nabla\phi(y) = \left[L\hat{v}^1, L\hat{v}^2, \ldots, L\hat{v}^p\right]^T, \tag{4}$$

where

$$(\hat{v}^1, \ldots, \hat{v}^p, s) = \underset{\substack{s^i\in S_k^p,\\ -Ms^i\le w^i\le Ms^i,\\ \forall i\in[\![N]\!]}}{\arg\min} f(w, y)$$

Proposition 2 follows directly from Danskin's theorem [21]. Given that the function $\phi(y)$ is concave and has an explicit formulation for computing its gradient, we can find its maximum using the classical gradient ascent.

We can derive the general framework for solving (2) as described in Algorithm 1. In Algorithm 1, we use $\psi_L^{i,t}$ and $\psi_N^{i,t}$ to denote the local multiplier and neighbor multiplier of agent i at iteration t respectively. We should keep in mind that the variable $\psi_L^{i,t}$ is just a rearrangement of the Lagrangian multiplier y in the function ϕ. Based on the strong duality from Lemma 1 and the distributed dual framework [18], we derive the following result. We note that in step 7 of Algorithm 1, we solve a mixed-integer programming problem, which makes the problem more difficult then the continuous version of linear regression.

Theorem 3 Assume that Problem (2) admits a unique solution $(\hat{s}^1, \ldots, \hat{s}^N)$ and $(\hat{w}^1, \ldots, \hat{w}^N)$. Furthermore, if at every iteration t in Algorithm 1, the step size α_t is chosen to be square summable but not summable, i.e.,

$$\sum_{t=1}^{\infty}\alpha_t = +\infty, \quad \text{and} \quad \sum_{t=1}^{\infty}\alpha_t^2 < \infty,$$

then for every $\epsilon > 0$, there exist T such that

$$\frac{1}{\|E(\mathcal{G})\|}\sum_{(i,j)\in E(\mathcal{G})}\|w_T^i - w_T^j\| \le \epsilon. \tag{5}$$

Furthermore, we have that $\lim_{t\to\infty} w_T^i = \hat{w}$, where $\hat{w}$ is the optimal solution of (2).

Proof. By [22, Theorem 7], we have $\lim_{t\to\infty} y_t = \hat{y}$, where $\hat{y} = \arg\max_y \phi(y)$. Hence we only need to show that the sequence of optimal solutions $s_t^1, \ldots, s_t^N$ and $w_t^1, \ldots, w_t^N$ of $\min_{\substack{-Ms_t^i\le w^i\le Ms_t^i,\\ \forall i\in[\![N]\!]}} f(w, y_t)$ at each iteration will converge to the optimal sparse regressor. To prove the desired result, we first show that the sequence of optimal solutions of the sparsity variables converges. Let

$$g(s^1, \ldots, s^N, y) = \min_{\substack{-Ms^i\le w^i\le Ms^i\\ \forall i\in[\![N]\!]}} f(w, y).$$

Since $(S_k^p)^N$ is a compact set, there exist a subsequence $\{s_{t_j}^1, \ldots, s_{t_j}^N\}_{j=1}^{\infty}$ of the sequence $\{s_t^1, \ldots, s_t^N\}_{t=1}^{\infty}$ such that

$$\lim_{j\to\infty}(s_{t_j}^1, \ldots, s_{t_j}^N) = (\bar{s}^1, \ldots, \bar{s}^N) \in (S_k^p)^N.$$

Moreover, because $(S_k^p)^N$ is finite, there exist a positive number $K_1 > 0$, such that for every $j \ge K_1$, we have $(s_{t_j}^1, \ldots, s_{t_j}^N) = (\bar{s}^1, \ldots, \bar{s}^N)$. Furthermore, for every $j \ge K_1$, it holds that

$$g(s_{t_j}^1, \ldots, s_{t_j}^N, y_t) = g(\bar{s}^1, \ldots, \bar{s}^N, y_{t_j}) \le g((\hat{s}^1, \ldots, \hat{s}^N, y_{t_j}),$$

by the optimality of $\bar{s}^1, \ldots, \bar{s}^N$. Nevertheless, when $(\bar{s}^1, \ldots, \bar{s}^N)$ is fixed, the function $g(\bar{s}^1, \ldots, \bar{s}^N, y)$ is continuous with respect to the variables y, thus

$$\begin{aligned}\lim_{j\to\infty} g(\bar{s}^1, \ldots, \bar{s}^N, y_{t_j}) &= g(\bar{s}^1, \ldots, \bar{s}^N, \hat{y})\\ &\le g(\hat{s}^1, \ldots, \hat{s}^N, \hat{y}).\end{aligned}$$

However, by definition of $\hat{y}$, we also have $g(\bar{s}^1, \ldots, \bar{s}^N, \hat{y}) \ge g(\hat{s}^1, \ldots, \hat{s}^N, \hat{y})$. Thus, by the uniqueness of optimal solution of (2), we must have $(\bar{s}^1, \ldots, \bar{s}^N) = (\hat{s}^1, \ldots, \hat{s}^N)$. Therefore, there exist $K_2 > 0$ such that for every $t > K_2$, we have

$$(s_t^1, \ldots, s_t^N) = (\hat{s}^1, \ldots, \hat{s}^N),$$

and

$$(w_t^1, \ldots, w_t^N) \in \underset{\substack{-M\hat{s}^i\le w^i\le M\hat{s}^i,\\ \forall i\in[\![N]\!]}}{\arg\min} f(w, y_t),$$

which is a strictly convex quadratic optimization problem. Hence, $w_t^1, \ldots, w_t^N$ is unique. Therefore,

$$\lim_{t\to\infty}(w_t^1, \ldots, w_t^N) = (\hat{w}^1, \ldots, \hat{w}^N).$$

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on September 12,2023 at 19:20:05 UTC from IEEE Xplore. Restrictions apply.

The conclusion of the theorem follows, because $\hat{w}^1, \dots, \hat{w}^N$ satisfies the coupling constraints, i.e., $\hat{w}^1 = \dots = \hat{w}^N$. $\square$

The consensus error in (5) can be interpreted as the average difference between two adjacent agents. When this error goes to zero, the agents reach a consensus on a sparse regressor. In the next section, we present an outer approximation algorithm [23] for solving the inner problem in Step 7 of Algorithm 1.

III. Quadratic Integer Programming Local Solver

In this section, we provide an algorithm for solving the local problem in step 7 of Algorithm 1. In particular, at each iteration, we need to solve a quadratic integer programming problem, which is given as

$$c^* = \min_{\|w\| \le k} \frac{1}{2}\|Y - Xw\|_2^2 + \frac{1}{\gamma}\|w\|_2^2 + \langle D, w\rangle. \tag{6}$$

In (6), for simplicity of notations, we drop the superscript denoting agents and the number of iterations. The case where $D = 0$ is, in fact, a sparse linear regression problem, which can be solved effectively using an outer approximation algorithm [6]. Motivated by the success of solving sparse linear regression in a very high-dimensional regime, we provide an alternative transformation of the objective function of (6), which is favorable for an outer approximation algorithm. Initially, we have

$$\begin{aligned}\frac{1}{2}\|Y - Xw\|_2^2 + \frac{1}{2\gamma}\|w\|_2^2 = \frac{1}{2}w^T(\frac{I}{\gamma} + X^TX)w \\ + Y^TXw + \frac{1}{2}Y^TY.\end{aligned}$$

For $\gamma > 0$, we have $(\frac{I}{\gamma} + X^TX)$ is a positive-definite matrix. Hence, there exists an invertible matrix $\bar{X} \in \mathbb{R}^{p\times p}$ such that $(\frac{I}{\gamma} + X^TX) = \bar{X}^T\bar{X}$. Since $\bar{X}$ is invertible[2], there exist $\bar{Y} \in \mathbb{R}^p$ such that $\bar{Y}^T\bar{X} = Y^TX$. Thus, we can rewrite the optimization problem (6) as

$$c^* = \min_{\|w\|_0 \le k} \frac{1}{2}(\|\bar{Y} - \bar{X}w\|_2^2 + \frac{1}{\gamma}\|w\|_2^2) + d^T\bar{X}w + \text{const}, \tag{7}$$

where $d = (\bar{X}^{-1})^TD$. The constant term in (7) equals to $\frac{1}{2}(Y^TY - \bar{Y}^T\bar{Y})$ and is dropped for simplicity. For a fixed $s \in S_k^p$, we define $c(s)$ as the optimal value of (7) with additional constraints $-Ms \le w \le Ms$, i.e.,

$$c(s) := \min_{-Ms\le w\le Mw} \frac{1}{2}||\bar{Y} - \bar{X}_sw||_2^2 + \frac{1}{2\gamma}||w||_2^2 + d^T\bar{X}_sw, \tag{8}$$

where $\bar{X}_s = \bar{X}I_s$ and $I_s \in \mathbb{R}^{p\times p}$ is the diagonal matrix whose diagonal is s. Moreover, (8) becomes a (continuous) quadratic programming problem, $c(s)$ can be explicitly computed as:

$$\begin{aligned}2c(s) = &-(\bar{Y}^T\bar{X}_s - d^T\bar{X}_s) \times (\frac{I}{\gamma} + X_s^TX_s)^{-1} \\ &\times (\bar{X}_s^T\bar{Y} - \bar{X}_s^Td) + \bar{Y}^T\bar{Y}.\end{aligned} \tag{9}$$

[2]Indeed, we do not need the L_2 regularization term as long as the matrix X^TX is invertible

Algorithm 2 Outer Approximation for Solving Local Problem

```
1: procedure OUTER APPROXIMATION($(\bar{X}, \bar{Y})$, $\gamma$, $d$)
2:     $s_1 \leftarrow$ warm start
3:     $\eta_1 \leftarrow 0$
4:     $t \leftarrow 1$
5:     while $\eta_t < c(s_t)$ do
6:         Compute $c(s_t)$ using Proposition 4
7:         Compute $\nabla c(s_t)$ using Proposition 5
8:         $s_{t+1}, \eta_{t+1} \leftarrow \arg\min_{s,\eta}$
           $\{\eta | s \in S_k^p,\ \eta \ge c(s_i) + \nabla c(s_i)(s - s_i)\ \forall i \in [\![t]\!]\}$
9:         $t \leftarrow t + 1$
10:    $\hat{s} \leftarrow s_t$
11:    $\hat{d} \leftarrow (\bar{Y} - d)^T\bar{X}_s$
12:    $\hat{w} \leftarrow \left(\frac{I_p}{\gamma} + \bar{X}_s^T\bar{X}_s)\right)^{-1}\hat{d}$
```

In the next proposition, we provide a simpler equivalent transformation for $c(s)$, which enables a simple computation of the value of $c(s)$ and its gradient.

Proposition 4 For a fixed $s \in S_k^p$, we have

$$c(s) = \frac{1}{2}(\bar{Y}^T - d^T)(I + \gamma\sum_{i=1}^p s_iK_i)^{-1}(\bar{Y} - d) - \frac{1}{2}d^Td + \bar{Y}^Td,$$

where $K_i = \bar{X}_i\bar{X}_i^T$ and $\bar{X}_i$ is the i^{th} column of $\bar{X}$.

By Proposition 4, the optimization problem (6) can now be reformulated as

$$\min_{s\in S_k^p} \frac{1}{2}(Y^T - d^T)(I + \gamma\sum_{i=1}^p s_iK_i)^{-1}(Y - d) - \frac{1}{2}d^Td + Y^Td. \tag{10}$$

The next proposition allows us to take the derivative of $c(s)$ for $s \in \text{conv}(S_k^p)$.

Proposition 5 The function

$$c(s) = \frac{1}{2}(Y^T - d^T)(I + \gamma\sum_{i=1}^p s_iK_i)^{-1}(Y - d) - \frac{1}{2}d^Td + Y^Td$$

is convex and continuous on $\text{conv}(S_k^p)$. Furthermore, its gradient is given by

$$\nabla_s c(s) = -\frac{1}{2}\alpha(s)^TK_i\alpha(s),$$

where $\alpha(s) = (I + \gamma\sum_{i=1}^p s_iK_i)^{-1}(Y - d)$, for $i \in [\![p]\!]$.

Proposition 4 and Proposition 5 derive a simple representation of $c(s)$ and its derivative. These results help us to attain an outer approximation algorithm for solving (7), see Algorithm (2).

According to [24, Theorem 2], Algorithm 2 will stop after a finite number of iterations and return the optimal solution of (7). This implies that Problem (1) is dual-friendly [18], [25]–[27]. Certainly, to the best of our knowledge, a closed form or a polynomial algorithm does not exist for solving a

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on September 12,2023 at 19:20:05 UTC from IEEE Xplore. Restrictions apply.

quadratic integer programming problem. However, the ability to yield the exact optimal solution of (7) can help us derive a convergence analysis of Algorithm (1) in terms of the number of iterations T.

IV. Numerical Experiments

To evaluate the convergence behavior of Algorithm 1, we perform a series of experiments on different datasets and different network structures. Before presenting our numerical experiments, we first describe how we generate our synthetic dataset. The input data X and its corresponding observations Y are generated following a linear relationship, i.e.,

$$Y = Xw^* + W,$$

where w^* is considered to be the true regressor with $\|w\|_0 = k$. To generate a true regressor w^*, we first pick k indices from $[\![p]\!]$ as non-zero entries. Afterwards, the chosen k non-zero entries of w^* are drawn from an uniform distribution on $[-1, 1]$. The white noise W is a random vector whose components are independently drawn from a normal distribution $\mathcal{N}(0, \sigma^2)$ for $\sigma > 0$. The input data $X = (x_1, \ldots, x_n)$ is sampled from an independent identically distributed (i.i.d.) Gaussian distribution $\mathcal{N}(0, \Sigma)$, where $\Sigma_{i,j} = \rho^{|i-j|}$. Finally, in all of the following computational experiments, we set $\delta = \rho = 0.1$. It is well-known that the square summable but not summable step size, albeit guarantees convergence, does not provide the best numerical performance. Hence, in all of the following experiments, initially, every agent starts with the same step-size α and a learning rate κ. At each iterations t, after receiving regressor weights from their neighbors, each agent i computes the average difference between its own local regressor and its neighbors', i.e.,

$$\epsilon_i^t := \frac{1}{\|N(i)\|} \sum_{j \in N(i)} \|w_t^i - w_t^j\|_2^2. \tag{11}$$

If at a certain iteration t, every agent has their current local error larger than the previous step's $\epsilon_i^t \geq \epsilon_i^{t-1}\ \forall i \in [\![N]\!]$, we update the step size for all agent by damping it down using the pre-defined learning rate, $\alpha_{new} = \kappa\alpha$.

In Figure 1, we show the simulation results of Algorithm 1 on a small-world network of size $N = 50$ while we vary the number of features p and the size of true support k. The small-world property of our network is generated by using the Watts-Strogatz Algorithm [28] using a mean degree $K = 12$ and $\beta = 0.25$. For each different choice of p, we generate 5 different datasets and plot the mean as a bold line and the 95% confidence range of the error of (5) across $T = 100$ iterations as a colored shadow.

In Figure 2, we analyze Algorithm 1 convergence behavior on different network classes. We pick four common network structures: clique, star, cycle, and small-world. Similarly, as in Figure 1, for each network structure, we run Algorithm 1 five times on 5 different randomly generated dataset for 100 iterations or until the error (5) is below 10^{-5}, and record the mean and the 95% confidence interval of the error (5).

In Figure 3, we observe the error (5) after $T = 100$ iterations on a path graph with various sizes. We first

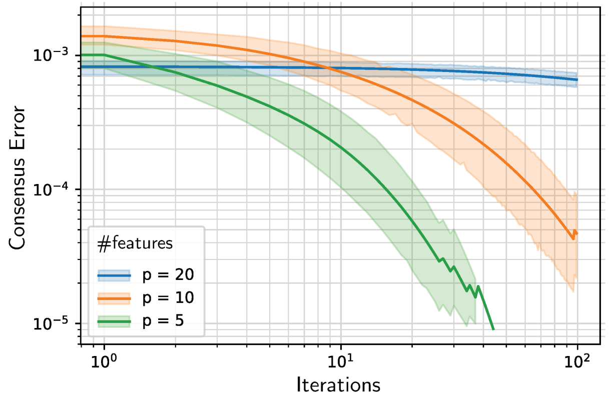

Fig. 1: **Convergence with respect to number of features p.** The value for k is chosen to be approximately 15% the value of p. In particular, in the case where $p = 20$, $k = 3$, when $p = 10$, $k = 2$ and for $p = 5$, $k = 1$. The number of data-observations stored within each agent is set to scale linearly with p and k, i.e., every agents has the same number of data points $n_i = 10pk$ for every $i \in [\![N]\!]$.

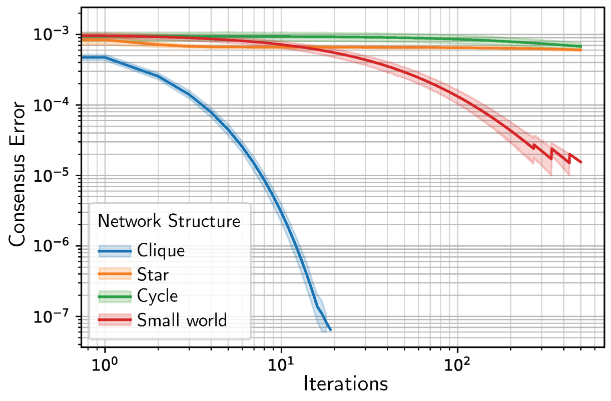

Fig. 2: **Influence of network structure on convergence rate.** The number of feature p is set to be 18, k is set to be 3, and the number of agents N is chosen to be 50 for every network structure.

generate a dataset with 2000 data points. Then, we distribute the generated dataset evenly among all agents for different network sizes. We then run Algorithm 1 on each of these settings. Figure 3 shows that as the number of agents increases, the number of iterations needed to reach the same error increases.

Remark 1 Note that the optimal solution $\hat{w}$ of (2) is not the same the true regressor w^*. Their support is the same, however, due to the L_2-regularization term, the value $\|\hat{w} - w^*\| > 0$. Moreover, we have shown in Theorem 3 that, at each agent i, we have a Cauchy sequence $\{w_t^i\}_{t=1}^{\infty}$ that converges to the optimal solution $\hat{w}$. Therefore, the sub-optimal gap $\|w_t^i - \hat{w}\|$ behaves similarly to the consensus error shown in Figures 1, 2, and 3.

For our current computational implementation, we are limited to solving the sparse linear regression problem with

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on September 12,2023 at 19:20:05 UTC from IEEE Xplore. Restrictions apply.

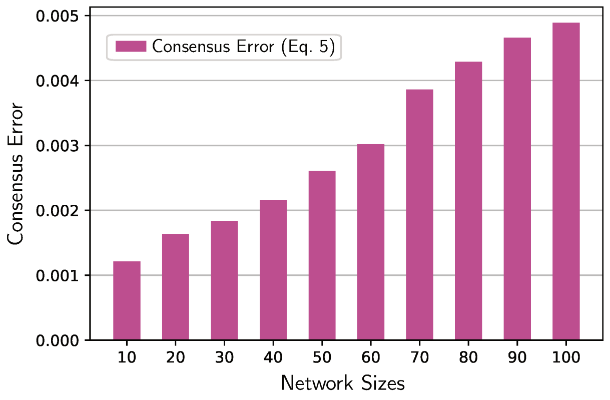

Fig. 3: **Influence of network's size on convergence rate.** The dataset is generated using a regressor with $p = 18$ features where there are 3 non-zero coefficients, and the underlying graphs for the networks are paths.

small p. This is because, at each iteration, we need to solve N integer programming problems. This requires extensive computing resources. Our software implementation for solving the inner QIP problems is not faster than a commercial solver like Gurobi [29].

V. Conclusion and Future Research

We have presented a decentralized scheme for solving exact sparse linear regression problems in the current work. We prove the convergence of the proposed method to the desired sparse solution. Our main contribution sets a foundational approach for the study of distributed optimization methods for larger problem classes with explicit sparsity constraints. The benefits of distributed sparse regression can be summarized as: interpretable regressors, solutions for distributed storage, and sparse communication between machines. Future work should investigate the theoretical convergence rate of Algorithm 1 knowing that the problem has a dual-friendly structure. Numerical experiments on larger problem scales should be studied. As possible extensions, we consider the case where the graph is directed or time-varying as described in [30]. Another direction is to study the convergence behavior of Algorithm 1 when the sparse regressor at each agent is not optimal. As we can see in Algorithm 2, the gap $\eta_t - c(s_t)$ is non-decreasing, and the moment this quantity attains a non-negative value, we are at an optimal solution. Thus, $\eta_t - c(s_t)$ is a surrogate sub-optimality gap.

References

[1] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2021.
[2] S. Weisberg, *Applied linear regression*. John Wiley & Sons, 2005, vol. 528.
[3] G. A. Seber and A. J. Lee, *Linear regression analysis*. John Wiley & Sons, 2012.
[4] K. R. Moon, D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, A. v. d. Elzen, M. J. Hirn, R. R. Coifman *et al.*, "Visualizing structure and transitions in high-dimensional biological data," *Nature biotechnology*, vol. 37, no. 12, pp. 1482–1492, 2019.
[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
[6] D. Bertsimas and B. Van Parys, "Sparse high-dimensional regression: Exact scalable algorithms and phase transitions," *The Annals of Statistics*, vol. 48, no. 1, pp. 300–323, 2020.
[7] 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.
[8] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, "A modified principal component technique based on the lasso," *Journal of computational and Graphical Statistics*, vol. 12, no. 3, pp. 531–547, 2003.
[9] D. Foster, S. Kale, and H. Karloff, "Online sparse linear regression," in *Conference on Learning Theory*. PMLR, 2016, pp. 960–970.
[10] M. Hellkvist, A. Özçelikkale, and A. Ahlén, "Linear regression with distributed learning: A generalization error perspective," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5479–5495, 2021.
[11] A. Gascón, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur, and D. Evans, "Privacy-preserving distributed linear regression on high-dimensional data." *Proc. Priv. Enhancing Technol.*, vol. 2017, no. 4, pp. 345–364, 2017.
[12] E. Dobriban and Y. Sheng, "Distributed linear regression by averaging," *The Annals of Statistics*, vol. 49, no. 2, pp. 918–943, 2021.
[13] 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, 2010.
[14] D. Bertsimas, A. King, and R. Mazumder, "Best subset selection via a modern optimization lens," *The annals of statistics*, vol. 44, no. 2, pp. 813–852, 2016.
[15] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1906, pp. 4273–4293, 2009.
[16] P. Bühlmann and S. Van De Geer, *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.
[17] G. David and Z. Ilias, "High dimensional regression with binary coefficients. estimating squared error and a phase transtition," in *Conference on Learning Theory*. PMLR, 2017, pp. 948–953.
[18] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić, "A dual approach for optimal algorithms in distributed optimization over networks," *Optimization Methods and Software*, vol. 36, no. 1, pp. 171–210, 2021.
[19] J. P. Vielma, "Mixed integer linear programming formulation techniques," *Siam Review*, vol. 57, no. 1, pp. 3–57, 2015.
[20] T. Anh-Nguyen and C. A. Uribe, "On distributed exact sparse linear regression over networks," *arXiv preprint arXiv:2204.00529*, 2022.
[21] J. M. Danskin, "The theory of max-min, with applications," *SIAM Journal on Applied Mathematics*, vol. 14, no. 4, pp. 641–664, 1966.
[22] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.
[23] M. A. Duran and I. E. Grossmann, "An outer-approximation algorithm for a class of mixed-integer nonlinear programs," *Mathematical programming*, vol. 36, no. 3, pp. 307–339, 1986.
[24] R. Fletcher and S. Leyffer, "Solving mixed integer nonlinear programs by outer approximation," *Mathematical programming*, vol. 66, no. 1, pp. 327–349, 1994.
[25] C. Dünner, S. Forte, M. Takác, and M. Jaggi, "Primal-dual rates and certificates," in *International Conference on Machine Learning*. PMLR, 2016, pp. 783–792.
[26] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.
[27] M. Raginsky and J. Bouvrie, "Continuous-time stochastic mirror descent on a network: Variance reduction, consensus, convergence," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 6793–6800.
[28] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[29] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2022. [Online]. Available: https://www.gurobi.com
[30] A. Nedić, A. Olshevsky, and C. A. Uribe, "Nonasymptotic convergence rates for cooperative learning over time-varying directed graphs," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 5884–5889.

Authorized licensed use limited to: Fondren Library Rice University. Downloaded on September 12,2023 at 19:20:05 UTC from IEEE Xplore. Restrictions apply.