Distributed Ridge Regression with Feature Partitioning

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 the ridge regression problem with feature partitioning of the observation matrix. The proposed algorithm, named D-Ridge, is based on the alternating direction method of multipliers (ADMM) and estimates the parameters when the observation matrix is distributed among different agents with feature (or vertical) partitioning. We formulate the associated ridge regression problem as a distributed convex optimization problem and utilize the ADMM to obtain an iterative solution. Numerical results demonstrate that D-Ridge converges faster than its diffusion-based contender does.**

I. Introduction

With the recent advances in technology, ever-growing amounts of data are constantly collected and stored on electronic devices, which are often geographically dispersed. Transporting the entire data to a central processing unit is often unfeasible due to energy constraints or privacy concerns. In addition, concentrating the data in a central hub can create a single point of failure. Hence, we need algorithms that are capable of processing data spread across multiple agents. They ought to operate in a distributed fashion relying only on the available local information [1]–[10].

Distributed solutions for learning, inference, or prediction using sensor data are highly demanded in many of today's data analysis tasks pertaining to statistics, signal processing and machine learning. In this context, an important data analysis tool is the distributed multivariate linear regression.

In recent years, there have been several works describing algorithms to distribute regression problems, i.e., [5]–[19]. In particular, shrinkage methods such as ridge regression and lasso have attracted a lot of attention since they play an important role in preventing the problem from being ill-posed due to possible rank deficiency of the observation matrix. Moreover, such methods regularize the regression parameters by imposing a penalty on their size or density to avoid overfitting [6], [8], [19]–[21]. Example applications are in wireless sensor networks operating under strict power budget constraints where agents collecting and processing data are distributed over a large geographical area [8].

A central issue in distributed regression is how the data are distributed among agents. Horizontal partitioning of data refers to the case when the data samples containing all features are distributed over the network. On the other hand, when subsets of features of all data samples are distributed over the agents, we have feature (vertical) partitioning of data [22]. Regression problems with horizontally partitioned data have been considered for example in [7], [8], [23]. In the framework of vertically partitioned data, some applications related to clustering and classification have been considered in [24], [25]. The regression problem with feature partitioning has also previously been considered in [6], [19]–[21]. However, works of [20], [21] assume a proper coloring scheme of the network and cannot be extended to a general graph labeling. The algorithm proposed in [19] is not truly distributed since its consensus constraints involve the entire network instead of each agent's local neighborhood. The algorithm in [6] is fully distributed and based on the diffusion strategy [26]. However, as we will show later on, it converges relatively slowly.

In this paper, we solve the ridge regression problem with feature partitioning of the observation matrix in a distributed fashion using the alternating direction method of multipliers (ADMM). The proposed algorithm, called D-Ridge, is fully distributed and requires communications only among neighboring agents. It also converges faster than the diffusion-based algorithm of [6] and has a per-iteration per-agent computational complexity order that is linear in the sample size. In addition, D-Ridge does not require the agents to share their local data or dual variables with the other agents but only the primal variables, which are the estimate solutions of the corresponding local optimization subproblems. Hence, D-Ridge respects the possible data privacy of the agents. We verify the convergence of D-Ridge to the centralized solution at all agents through both theoretical analysis and simulations. Our experiments with a verity of network topologies show that D-Ridge outperforms its diffusion-based contender in terms of convergence rate.

II. System Model

We consider a network with $K \in \mathbb{N}$ agents 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 edge set $\mathcal{E}$ represents the bidirectional communication links between the pairs of agents. Agent $k \in \mathcal{K}$ can communicate with the agents in its neighborhood set $\mathcal{N}_k$ whose cardinality is denoted by $|\mathcal{N}_k|$. The set $\mathcal{N}_k$ includes the agent k as well.

Let us denote the network-wide observations as an observation matrix $\mathbf{X} \in \mathbb{R}^{N \times P}$ and a response vector $\mathbf{y} \in \mathbb{R}^{N \times 1}$

This work was carried out during the tenure of an ERCIM 'Alain Bensoussan' Fellowship Programme and was partly supported by the Research Council of Noway.

where N is the number of data samples and P is the number of features in each sample. The data collected at each agent k are stored in the matrix $\mathbf{X}_k \in \mathbb{R}^{N\times P_k}$ where $\sum_{k=1}^{K} P_k = P$. Due to feature partitioning, the observation matrix $\mathbf{X} \in \mathbb{R}^{N\times P}$ consists of K submatrices $\mathbf{X}_k$, i.e., $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K]$. Accordingly, the parameter vector $\boldsymbol{\beta} \in \mathbb{R}^{P\times 1}$ that establishes a linear regression between $\mathbf{X}$ and $\mathbf{y}$ is a stack of K subvectors $\boldsymbol{\beta}_k \in \mathbb{R}^{P_k\times 1}$, i.e., $\boldsymbol{\beta} = \left[\boldsymbol{\beta}_1^{\mathrm{T}}, \boldsymbol{\beta}_2^{\mathrm{T}}, \dots, \boldsymbol{\beta}_K^{\mathrm{T}}\right]^{\mathrm{T}}$.

In the centralized approach, a ridge regression estimator of $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}}^o = \arg\min_{\boldsymbol{\beta}} \{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \eta\,\|\boldsymbol{\beta}\|_2^2\} \tag{1}$$

where $\eta > 0$ is the regularization parameter. From the normal equation associated with (1), the centralized estimate is given by

$$\hat{\boldsymbol{\beta}}^o = \mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \eta\mathbf{I}_N)^{-1}\mathbf{y} \tag{2}$$

where $\mathbf{I}_N$ indicates the $N \times N$ identity matrix.

Since computing a centralized solution of (1) over a network may be inefficient, we propose a distributed algorithm for this purpose in the following section.

III. Distributed Ridge Regression via ADMM

We first discuss the consensus-based reformulation of the ridge regression problem whose solution allows us to find a distributed solution to (1) via the ADMM. Then, we describe the construction steps and main properties of the proposed algorithm for solving the resulting constrained minimization problem. Finally, we establish the global convergence of D-Ridge theoretically.

A. Consensus-Based Reformulation of Ridge Regression

Let us define a vector $\mathbf{f}^o \in \mathbb{R}^{N\times 1}$ as

$$\mathbf{f}^o = (\mathbf{X}\mathbf{X}^{\mathrm{T}} + \eta\mathbf{I}_N)^{-1}\mathbf{y}.$$

From (2), the part of $\boldsymbol{\beta}^o$ corresponding to agent k can be calculated as

$$\hat{\boldsymbol{\beta}}_k^o = \mathbf{X}_k^{\mathrm{T}}\mathbf{f}^o. \tag{3}$$

For computing $\mathbf{f}^o$ at all agents using only in-network processing of the locally available data, we propose a consensus-based distributed algorithm. Note that $\mathbf{f}^o$ is the unique minimizer of the quadratic global cost function $\mathcal{J}(\mathbf{f})$ defined as

$$\mathcal{J}(\mathbf{f}) = \frac{1}{2}\mathbf{f}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \eta\mathbf{I}_N)\mathbf{f} - \mathbf{f}^{\mathrm{T}}\mathbf{y}. \tag{4}$$

Since $\mathbf{X}\mathbf{X}^{\mathrm{T}} = \sum_{k=1}^{K} \mathbf{X}_k\mathbf{X}_k^{\mathrm{T}}$, $\mathbf{f}^o$ is given by

$$\mathbf{f}^o = \arg\min_{\mathbf{f}} \sum_{k=1}^{K} \mathcal{J}_k(\mathbf{f}) \tag{5}$$

where

$$\mathcal{J}_k(\mathbf{f}) = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\Big(\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}} + \frac{\eta}{K}\mathbf{I}_N\Big)\mathbf{f} - \frac{\delta_k}{B}\mathbf{f}^{\mathrm{T}}\mathbf{y}, \tag{6}$$

$B \in \mathbb{N}$ is the number of agents having access to $\mathbf{y}$, and $\delta_k = 1$ if $\mathbf{y}$ is available at agent k and $\delta_k = 0$ otherwise.

We introduce the local variables $\mathcal{F} := \{\mathbf{f}_k\}_{k=1}^{K}$ representing the local copies of $\mathbf{f}^o$ at the agents. Then, we reformulate the unconstrained optimization problem (5) as the following convex *constrained* minimization problem:

$$\{\mathbf{f}_k^o\}_{k=1}^{K} = \arg\min_{\{\mathbf{f}_k\}} \sum_{k=1}^{K} \frac{1}{2}\mathbf{f}_k^{\mathrm{T}}\Big(\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}} + \frac{\eta}{K}\mathbf{I}_N\Big)\mathbf{f}_k - \frac{\delta_k}{B}\mathbf{f}_k^{\mathrm{T}}\mathbf{y}$$
$$\text{s.t. } \mathbf{f}_k = \mathbf{f}_l, \quad l \in \mathcal{N}_k, \quad k \in \mathcal{K}. \tag{7}$$

The equality constraints enforce local consensus over $\{\mathbf{f}_k\}$ across each agent's neighborhood.

To solve (7) in a distributed fashion, we use the ADMM [5]. Hence, we introduce the auxiliary local variables $\mathcal{A} := \{\mathbf{g}_k^l\}_{l\in\mathcal{N}_k}$ and rewrite the problem (7) as

$$\arg\min_{\{\mathbf{f}_k\}} \sum_{k=1}^{K} \frac{1}{2}\mathbf{f}_k^{\mathrm{T}}\Big(\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}} + \frac{\eta}{K}\mathbf{I}_N\Big)\mathbf{f}_k - \frac{\delta_k}{B}\mathbf{f}_k^{\mathrm{T}}\mathbf{y}$$
$$\text{s.t. } \mathbf{f}_k = \mathbf{g}_k^l, \mathbf{f}_l = \mathbf{g}_k^l, \quad l \in \mathcal{N}_k, \quad k \in \mathcal{K}, \quad k \neq l. \tag{8}$$

Using the auxiliary variables $\mathcal{A}$ yields an equivalent alternative representation of the constraints in (7). These variables are only used to derive the local recursions and are eventually eliminated. Associating the Lagrange multipliers $\mathcal{V} := \{\{\boldsymbol{\mu}_k^l\}_{l\in\mathcal{N}_k}, \{\boldsymbol{\lambda}_k^l\}_{l\in\mathcal{N}_k}\}_{k=1}^{K}$ with the constraints in (8), we have the following augmented Lagrangian function:

$$\begin{aligned}\mathcal{L}_\rho(\mathcal{F}, \mathcal{A}, \mathcal{V}) = &\sum_{k=1}^{K}\Big(\frac{1}{2}\mathbf{f}_k^{\mathrm{T}}\Big(\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}} + \frac{\eta}{K}\mathbf{I}_N\Big)\mathbf{f}_k - \frac{\delta_k}{B}\mathbf{f}_k^{\mathrm{T}}\mathbf{y}\Big)\\ &+ \sum_{k=1}^{K}\sum_{l\in\mathcal{N}_k}\Big((\boldsymbol{\mu}_k^l)^{\mathrm{T}}(\mathbf{f}_k - \mathbf{g}_k^l) + (\boldsymbol{\lambda}_k^l)^{\mathrm{T}}(\mathbf{f}_l - \mathbf{g}_k^l)\Big)\\ &+ \frac{\rho}{2}\sum_{k=1}^{K}\sum_{l\in\mathcal{N}_k}\Big(\big\|\mathbf{f}_k - \mathbf{g}_k^l\big\|_2^2 + \big\|\mathbf{f}_l - \mathbf{g}_k^l\big\|_2^2\Big)\end{aligned} \tag{9}$$

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

Minimizing (7) through ADMM entails an iterative process that is described in the next section.

B. Algorithm Description

The D-Ridge algorithm consists of three steps at each iteration. First, the augmented Lagrangian function $\mathcal{L}_\rho$ is minimized with respect to $\mathcal{F}$. Second, $\mathcal{L}_\rho$ is minimized with respect to $\mathcal{A}$. Finally, the Lagrange multipliers in $\mathcal{V}$ are updated through gradient-ascent [27].

Thanks to the reformulation of the original problem (5) as (8), the augmented Lagrangian in (9) is decomposable both with respect to variables in $\mathcal{F}$, $\mathcal{A}$ and across agents.

Setting $\boldsymbol{\mu}_k(m) = 2\sum_{l\in\mathcal{N}_k}\boldsymbol{\mu}_k^l(m)$, and using the Karush-Kuhn-Tucker conditions of optimality [28] for (8), the auxiliary local variables $\mathcal{A}$ and multipliers $\mathcal{V}$ can be eliminated.

Algorithm 1 D-Ridge

```
At all agents $k \in \mathcal{K}$, initialize $\mathbf{f}_k(0)$, $\boldsymbol{\mu}_k(0)$ to zero vectors, and run locally
for $m = 0, 1, \ldots, M$ do
    Receive $\mathbf{f}_k(m)$ from neighbors in $\mathcal{N}_k$.
    Update $\boldsymbol{\mu}_k(m)$ as in (10).
    Update $\mathbf{f}_k(m+1)$ as in (11).
end for
Estimate $\hat{\boldsymbol{\beta}}_k = \mathbf{X}_k^{\mathrm{T}}\mathbf{f}_k(M+1)$.
```

Hence, the D-Ridge algorithm reduces to the following iterative updates that are carried out locally at every agent:

$$\boldsymbol{\mu}_k(m) = \boldsymbol{\mu}_k(m-1) + \rho \sum_{l\in\mathcal{N}_k} [\mathbf{f}_k(m) - \mathbf{f}_l(m)] \tag{10}$$

$$\begin{aligned}\mathbf{f}_k(m+1) &= \arg\min_{\{\mathbf{f}_k\}} \Big\{ \frac{1}{2}\mathbf{f}_k^{\mathrm{T}}\Big(\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}} + \frac{\eta}{K}\mathbf{I}_N\Big)\mathbf{f}_k - \frac{\delta_k}{B}\mathbf{f}_k^{\mathrm{T}}\mathbf{y} \\ &\quad + \boldsymbol{\mu}_k^{\mathrm{T}}(m)\mathbf{f}_k + \rho \sum_{l\in\mathcal{N}_k} \Big\| \mathbf{f}_k - \frac{\mathbf{f}_k(m) + \mathbf{f}_l(m)}{2} \Big\|_2^2 \Big\} \\ &= \Big[\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}} + \Big(\frac{\eta}{K} + 2\rho|\mathcal{N}_k|\Big)\mathbf{I}_N\Big]^{-1} \\ &\quad \Big(\frac{\delta_k}{B}\mathbf{y} - \boldsymbol{\mu}_k(m) + \rho|\mathcal{N}_k|\mathbf{f}_k(m) + \rho\sum_{l\in\mathcal{N}_k}\mathbf{f}_l(m)\Big)\end{aligned} \tag{11}$$

where m is the iteration index and all initial values $\{\mathbf{f}_k(0)\}_{k\in\mathcal{K}}$, $\{\boldsymbol{\mu}_k(0)\}_{k\in\mathcal{K}}$ are set to zero. The proposed approach is summarized in Algorithm 1.

Note that $\mathbf{f}_k(m)$ is the only vector that is shared between the agents at every iteration. The computation of (11) has a per-iteration per-agent complexity of $\mathcal{O}(N^3 + N^2 P_k)$. It involves the inversion of the $N \times N$ matrix $\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}} + \Big(\frac{\eta}{K} + 2\rho|\mathcal{N}_k|\Big)\mathbf{I}_N$ that may be computationally demanding for $N \gg P_k$. However, this operation can be carried out off-line before running the algorithm. We can also use the matrix inversion lemma to obtain $(\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}} + c\mathbf{I}_N)^{-1} = c^{-1}[\mathbf{I}_N - \mathbf{X}_k(c\mathbf{I}_{P_k} + \mathbf{X}_k^{\mathrm{T}}\mathbf{X}_k)^{-1}\mathbf{X}_k^{\mathrm{T}}]$ where $c = \frac{\eta}{K} + 2\rho|\mathcal{N}_k|$. Hence, the dimensions of the matrix to be inverted become $P_k \times P_k$ entailing a per-iteration per-agent computational complexity of $\mathcal{O}(NP_k^2 + P_k^3)$.

In the next subsection, we show that D-Ridge generates sequences of local iterates $\mathbf{f}_k(m)$, $k = 1, \ldots, K$, that, at each agent k, converge to the global centralized solution $\mathbf{f}^o$ as $m \to \infty$.

C. Convergence Analysis

Convergence of the proposed algorithm is established by verifying that both conditions for the ADMM to converge are fulfilled, namely, for each agent $k \in \mathcal{K}$, the cost function $\mathcal{J}_k(\mathbf{f})$ is strongly convex and its gradient $\nabla_{\mathbf{f}}\mathcal{J}_k(\mathbf{f})$ is Lipschitz continuous [29].

The function $\mathcal{J}_k(\mathbf{f})$ is strongly convex since it is twice continuously differentiable and has a positive-definite Hessian matrix:

$$\nabla_{\mathbf{f}}^2\mathcal{J}_k(\mathbf{f}) = \mathbf{X}_k\mathbf{X}_k^{\mathrm{T}} + \frac{\eta}{K}\mathbf{I}_N \succ 0.$$

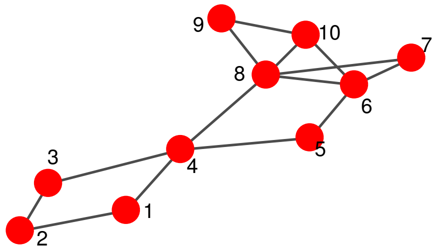

Fig. 1. Topology of the considered multi-agent network.

Moreover, $\nabla_{\mathbf{f}}\mathcal{J}_k(\mathbf{f})$ is a linear function of $\mathbf{f}$. Therefore, it is Lipschitz continuous [30] with a Lipschitz constant being the operator norm of $\nabla_{\mathbf{f}}^2\mathcal{J}_k(\mathbf{f})$.

IV. Simulations

The D-Ridge algorithm is tested here on a network of $K = 10$ agents with the topology as shown in Fig. 1. Each agent holds the data for two features. Therefore, $P_k = 2$, $k = 1, ..., K$, and $P = 20$. The observation data matrix $\mathbf{X}$ has $N = 50$ regressor vectors with independent zero-mean multivariate Gaussian distribution as its rows. The relationship between the entries of $\mathbf{y}$, denoted by $y_n \in \mathbb{R}$, and the rows of $\mathbf{X}$, denoted by $\mathbf{x}_n \in \mathbb{R}^{1\times P}$, with $n = 1, \ldots, N$, is governed by

$$y_n = \sum_{k=1}^{K} \mathbf{x}_{n,k}\boldsymbol{\beta}_k + \epsilon_n$$

where $\mathbf{x}_{n,k} \in \mathbb{R}^{1\times P_k}$ is the part of $\mathbf{x}_n$ that is available at agent k and $\epsilon_n \in \mathbb{R}$ is the zero-mean Gaussian noise with variance $\sigma_\epsilon^2 = 0.1$. The penalty parameter is set to $\rho = 4$ and, as in [6], the regularization parameter is set to $\eta = 10^{-3}$.

In Figs. 2-4, we plot the normalized mean squared error (MSE) versus the iteration index for D-Ridge and the diffusion-based algorithm of [6] with different values of the step-size μ.

The normalized MSE is defined as

$$n_{\mathrm{MSE}}(m) = \frac{\sum_{k=1}^{K} \|\boldsymbol{\beta}_k(m) - \boldsymbol{\beta}_k\|_2^2}{\|\boldsymbol{\beta}\|_2^2}$$

where $\boldsymbol{\beta}_k$ is given by (3) and $\boldsymbol{\beta}_k(m) = \mathbf{X}_k^{\mathrm{T}}\mathbf{f}_k(m)$.

The results in Figs. 2-4 are obtained by averaging over 100 independent trials. The number of agents having access to $\mathbf{y}$, i.e., B affects the convergence speed of D-Ridge, while it does not have any significant effect on the performance of the diffusion-based algorithm [6]. In Figs. 2-4, the regression vector is placed in the agent k with the greatest $|\mathcal{N}_k|$ if $B = 1$, while it is randomly placed over the network if $B > 1$.

Fig. 2 shows that D-Ridge converges significantly faster than the diffusion-based algorithm, especially when all agents have access to $\mathbf{y}$, i.e., $B = 10$. Fig. 3 shows that the D-Ridge algorithm converges faster as the number of agents that have access to $\mathbf{y}$ increases. Fig. 4 shows that D-Ridge converges faster than the diffusion-based algorithm irrespective of the

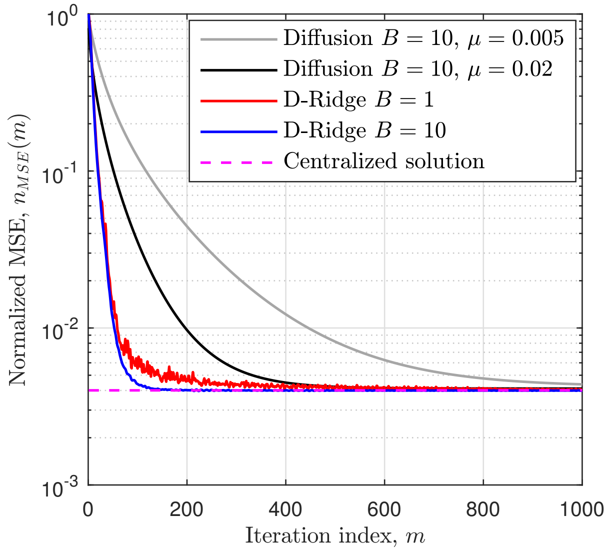

Fig. 2. Normalized MSE of D-Ridge and the diffusion-based algorithm with different values of the step-size $\boldsymbol{\mu}$ when one or all agents have access to $\mathbf{y}$.

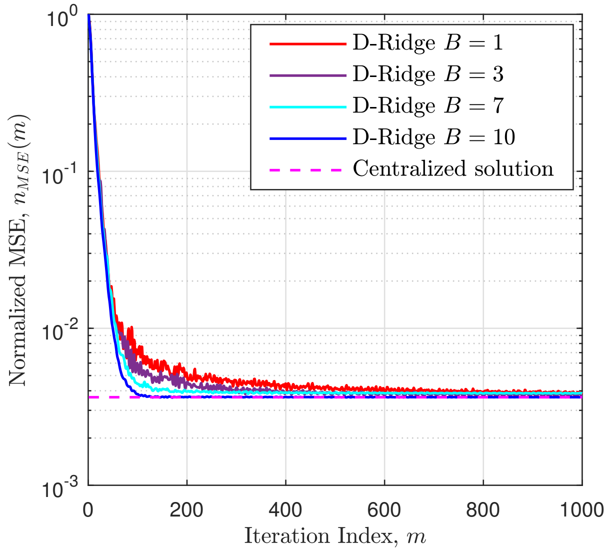

Fig. 3. Normalized MSE of D-Ridge for different values of B.

network topology. The performance of the algorithm with the topology shown in Fig. 1 is compared to a linear topology with the same number of agents where the agents are connected one after the other, hence $|\mathcal{N}_k| = 3$ for $1 < k < K$ and $|\mathcal{N}_k| = 2$ for $k = 1$ and $k = K$.

V. Conclusion

In this paper, we developed a new consensus-based algorithm for distributed solution of the ridge regression problem with feature partitioning of the observation matrix. To this end, we recast the ridge regression problem into an equivalent constrained separable form, whose structure is suitable for distributed implementation through ADMM. In the proposed algorithm, D-Ridge, the agents exchange messages only within their neighborhoods. Simulation results showed that the sequences of local iterates generated by D-Ridge converge to the centralized solution faster than the diffusion-based algorithm does.

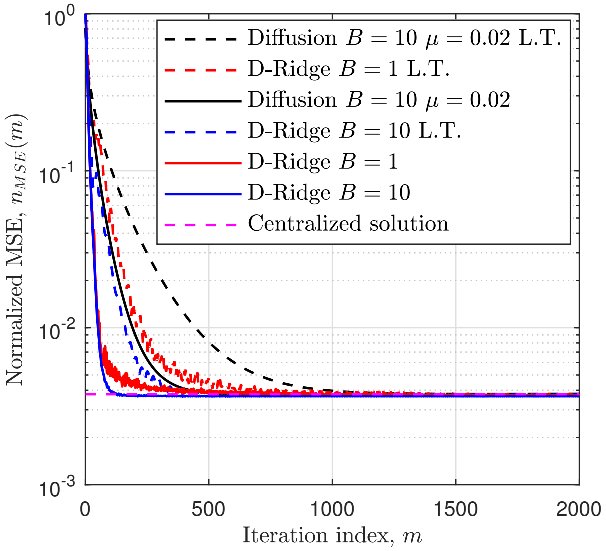

Fig. 4. Normalized MSE of D-Ridge and the diffusion-based algorithm for the considered network topology and for the linear topology (L.T.).

References

[1] 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.

[2] C. Li, S. Huang, Y. Liu, and Z. Zhang, “Distributed jointly sparse multitask learning over networks,” *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 151–164, Jan. 2018.

[3] 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.

[4] S. P. Talebi, S. Werner, and D. P. Mandic, “Distributed adaptive filtering of α-stable signals,” *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1450–1454, Oct. 2018.

[5] 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.

[6] 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.

[7] R. Arablouei, S. Werner, and K. Doğançay, “Diffusion-based distributed adaptive estimation utilizing gradient-descent total least-squares,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 5308–5312.

[8] 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.

[9] I. Schizas, G. Mateos, and G. Giannakis, “Distributed LMS for consensus-based in-network adaptive processing,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, Jun. 2009.

[10] G. Mateos, I. Schizas, and G. Giannakis, “Distributed recursive least-squares for consensus-based in-network adaptive estimation,” *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4583–4588, Nov. 2009.

[11] 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.

[12] ——, "Low-complexity distributed total least squares estimation in ad hoc sensor networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4321–4333, Aug. 2012.

[13] R. Abdolee and B. Champagne, "Diffusion LMS strategies in sensor networks with noisy input data," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 3–14, Feb. 2016.

[14] L. Lu, H. Zhao, and B. Champagne, "Diffusion total least-squares algorithm with multi-node feedback," *Signal Processing*, Jul. 2018.

[15] R. Arablouei, S. Werner, Y.-F. Huang, and K. Doğançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–484, Jan. 2014.

[16] R. Arablouei, K. Doğançay, S. Werner, and Y.-F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3510–3522, Jul. 2014.

[17] R. Arablouei, S. Werner, and K. Doğançay, "Partial-diffusion recursive least-squares estimation over adaptive networks," in *Proc. 2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Dec. 2013, pp. 89–92.

[18] R. Arablouei, K. Doğançay, and S. Werner, "Reduced-complexity distributed least-squares estimation over adaptive networks," in *Proc. 2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications*, Jun. 2013, pp. 150–154.

[19] N. Kashyap, S. Werner, Y.-F. Huang, and R. Arablouei, "Privacy preserving decentralized power system state estimation with phasor measurement units," in *Proc. 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop*, Jul. 2016, pp. 1–5.

[20] 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.

[21] ——, "D-admm: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, May 2013.

[22] 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.

[23] J. Predd, S. Kulkarni, and H. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, Jul. 2006.

[24] J. Vaidya and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," in *Proc. 9th ACM International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 206–215.

[25] O. L. Mangasarian, E. W. Wild, and G. M. Fung, "Privacy-preserving classification of vertically partitioned data via random kernels," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 3, Oct. 2008.

[26] A. H. Sayed, "Adaptive Networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[27] D. P. Bertsekas, *Parallel and distributed computation : numerical methods*. Englewood Cliffs, N.J: Prentice-Hall, 1989.

[28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[29] T. Lin, S. Ma, and S. Zhang, "On the global linear convergence of the admm with multiblock variables," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1478–1497, Jan. 2015.

[30] K. Eriksson, *Applied Mathematics: Body and Soul : Volume 1: Derivatives and Geometry in IR3*, 2004.