2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Distributed Networked Learning with Correlated Data

Lingzhou Hong, Alfredo Garcia, and Ceyhun Eksin

Abstract— **This paper considers a learning problem with heteroscedastic and correlated data that is distributed across nodes. We propose a distributed learning scheme where each node asynchronously implements stochastic gradient descent updates and exchanges their current models with neighbors. We ensure the similarity among the local models and the ensemble average by having a network regularization penalty to the least squares problem. This penalty is associated with weights that are proportional to the relative accuracy of local models. We further provide finite time characterization of the disparity between local models and the ensemble average model based on the penalty constants and network connectivity. We compare the proposed method with generalized least squares and logistic regression in the prediction of activities of individuals based on head movement data.**

I. Introduction

In many applications, like cell phones, sensors, or other computing devices, data is inherently collected from spatially distributed sources. Given the volume of data, communication limitations, and security and privacy issues, a distributed architecture may be preferred over centralized storage and processing. However, an isolated architecture, where processing is solely based on available (local) data, may create vast disparities in performance across computing nodes. Moreover, local datasets can be correlated and heteroscedastic. In such a scenario, a simple averaging of local models can perform significantly worse than a centralized model built at a fusion center. If a centralized architecture is not desirable and a fully distributed architecture may be lacking in performance, it is of interest to study alternatives that make use of possible network structure in data and communication capabilities among nodes.

In addressing the shortcomings of centralized and isolated settings mentioned above, we consider a network of local learners. Each learner accesses to a local dataset and solves a distributed estimation problem with a network regularization penalty, which enforces the model to be similar to its neighbors. This penalization method that reduces performance disparity among local models is similar to methods known as Network Lasso [1]–[3] and graph Laplacian regularization [4]. Both existing methods aim to improve local learners' estimates by making use of neighboring models. However, the underpinning modeling assumption in these studies is that local datasets are independent and identically distributed.

Unlike these approaches, here we consider local datasets that suffer from both global and local noise. In such a setting, one node may have "better" data than the other, we cannot assume all neighbors to be equal. Two relevant approaches are model averaging [5], [6] and ensemble learning [7], [8], e.g., "bagging" in statistics that aim to find a weighted averaging of models to reduce variance and increase forecast robustness against measurement errors. In some settings, local computing nodes weight the neighboring models by their data fidelity. However, often in the ensemble learning methods, e.g., "divide and conquer" [9], averaging is done at a single step after all local models are identified in isolation. This synchronous updating scheme does not provide a good working model for local nodes and assumes aggregation at a fusion center. Instead in this paper, we propose an updating scheme where nodes implement an asynchronous distributed stochastic gradient descent algorithm [10]–[13]. The proposed approach can not be interpreted as consensus-based optimization (see e.g. [11] and [14]). We are not aiming to find a common linear estimate for all the nodes but to maintain sufficiently cohesion among *diverse* local models that the ensemble solution is (eventually) arbitrarily close to that of the generalized least squares problem.

In stochastic gradient with network regularization (SGN), the network regularization penalty requires nodes to exchange their current models with each other after each update. We show that SGN updates converge, and provide a finite time bound for the disparity between local models (Theorem 1). In the analysis, we assume the regression model is linear. Thus, given heteroscedastic and correlated data, the centralized regression problem is generalized least squares (GLS). Our method approximates the centralized GLS by assuming a network structure in data and imposing the smoothness of models across nodes. We characterize finite-time bounds for the optimality of the ensemble (weighted average) model (Theorem 2). Similar to Federated Learning approaches [15]–[17], we locally process data, iteratively average local models, and our analysis focuses on the ensemble average. Unlike Federated Learning, this ensemble

Authors are with the Industrial and Systems Engineering Department, Texas A&M University, College Station, TX 77843. E-mail:{hlz, alfredo.garcia, eksinc}@tamu.edu

This work was supported by NSF ECCS-1933878, NSF CCF-2008855, and Grant AFOSR-15RT0767.

978-1-7281-7447-1/20/$31.00 ©2020 IEEE

model is not kept at any single location (fusion center) and is only a measure of the method's robustness.

We compare the performance of SGN to the centralized GLS solution on a head movement dataset collected via Google Glass [18]. In the numerical implementation, local data fidelity is not known. We compound our method with mini-batching for stochastic gradient computation, and a fading memory update rule to compute local data accuracy. Our distributed approach compares well against both GLS and logistic regression (LR).

II. Network of Local Learners

There are $N > 1$ nodes each with access to its local dataset $(\mathbf{X}_i, \mathbf{y}_i)$, where $\mathbf{X}_i \in \mathbb{R}^{m\times d}$ is the input matrix with d features, $\mathbf{y}_i \in \mathbb{R}^m$ is the associated output vector. Node i would like to find the best linear model with coefficients $\mathbf{w}_i \in \mathbb{R}^d$ by minimizing the following function,

$$f_i(\mathbf{w}_i) = \frac{1}{2}(\mathbf{y}_i - \mathbf{X}_i\mathbf{w}_i)^T\Omega_i^{-1}(\mathbf{y}_i - \mathbf{X}_i\mathbf{w}_i), \qquad (1)$$

where Ω_i is the covariance matrix of the error term in a linear model for $\mathbf{y}_i$.

The set of nodes $\mathcal{V} := \{1, \ldots, N\}$ is connected via a communication network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with edge set $\mathcal{E}$. We use $\alpha_{i,j}$ to denote the ij-th element of the adjacency matrix. Nodes i and j can exchange information, if there is an edge between them, i.e., if $\alpha_{i,j} = 1$. In particular, we assume neighboring agents exchange their models with each other. Each node solves the problem

$$\min_{\mathbf{w}_i} \big(f_i(\mathbf{w}_i) + \lambda\rho_i(\mathbf{w}_i)\big), \qquad (2)$$

where

$$\rho_i(\mathbf{w}_i) = \frac{1}{2}\sum_{j=1,j\neq i}^{N} \frac{\alpha_{i,j}}{\mathrm{tr}(\Omega_j)} \|\mathbf{w}_i - \mathbf{w}_j\|^2 \qquad (3)$$

is the network regularization penalty with parameter $\lambda \geq 0$. In (3), each neighboring node's model is weighted by the trace of the covariance matrix. The larger the trace of the covariance matrix of j is, the smaller is the weight i has on j's model. That is, each node prioritizes neighboring nodes with better data fidelity. As λ gets larger, node i's model gets closer to a weighted sum of its neighbors' models.

In the following we specify the assumptions on the dataset. There exists a ground truth coefficient vector $\mathbf{w}^* \in \mathbb{R}^d$. The output vector model for node i is given as follows,

$$\mathbf{y}_i = \mathbf{X}_i\mathbf{w}^* + \varepsilon_i + \Lambda_i\xi, \quad i \in \mathcal{V} := \{1, \ldots, N\}, \quad (4)$$

where $\varepsilon_i \in \mathbb{R}^{m\times 1}$ is an *individual* noise vector specific to data subset i, and $\xi \in \mathbb{R}^{m\times 1}$ is a *common* noise which affects different subsets differently according to the matrices $\Lambda_i \in \mathbb{R}^{m\times m}$. We assume Λ_i is a diagonal matrix with possibly different diagonal entries.

We assume the individual noise vector is zero-mean and independent across different nodes, i.e., $\mathrm{E}[\varepsilon_i\varepsilon_j^\intercal] = \mathbf{0}_{m\times m}$ for all i and $j \neq i$, and $\mathrm{E}\,\|\varepsilon_i\varepsilon_i^\intercal\| = \sigma_i^2\mathbf{I}_m$. Also, $\mathrm{E}[\xi] = \mathbf{0}_m$ and $\mathrm{E}[\|\xi\|^2] = \mathbf{I}_m$. From the model in (4), it follows the covariance matrix of the error term in the model for $\mathbf{y}_i$ as

$$\Omega_i := \mathrm{E}\,\|\varepsilon_i + \Lambda_i\xi\|^2 = \sigma_i^2\mathbf{I} + \Lambda_i^2 \in \mathbb{R}^{m\times m}. \qquad (5)$$

Throughout the analysis, we assume $\{\Omega_1, \Omega_2, \ldots, \Omega_N\}$ is known by all nodes. In Section IV, we consider an update rule to estimate (5).

Given the model in (4), the centralized problem at a fusion center node that have access to all the data $\{(\mathbf{X}, \mathbf{y}) : \mathbf{X} = [\mathbf{X}_1^\intercal, \ldots, \mathbf{X}_N^\intercal]^\intercal, \mathbf{y} = [\mathbf{y}_1^\intercal, \ldots, \mathbf{y}_N^\intercal]^\intercal\}$ and solves a GLS problem:

$$\min_{\mathbf{w}} \Big(\frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T\Omega^{-1}(\mathbf{y} - \mathbf{X}\mathbf{w})\Big), \qquad (6)$$

where $\Omega = \mathrm{E}(\varepsilon + \Lambda\xi)(\varepsilon + \Lambda\xi)^\intercal = \Sigma + \Lambda\Lambda^T \in \mathbb{R}^{p\times p}$, with $\Lambda = [\Lambda_1, \ldots, \Lambda_N]^\intercal$, $\varepsilon = [\varepsilon_i^\intercal, \ldots, \varepsilon_N^\intercal]^\intercal$, and Σ as a block-diagonal matrix with the i-th block as $\sigma_i^2\mathbf{I}_m$. While the problem afforded by minimizing (1) is a weighted least squares problem, the local minimization problem in (2) approximates the GLS problem (6) by assuming a network-structure among local datasets.

III. Distributed Stochastic Gradient Descent

Each local node is implementing a stochastic gradient descent algorithm to solve the least squares problem with network regularization (2). For $k = 1, 2, \ldots,$

$$\mathbf{w}_{i,k+1} = \mathbf{w}_{i,k} - \Gamma(\nabla f_{i,k} + \lambda\nabla\rho_{i,k}), \quad k \in \mathbb{N}^+ \quad (7)$$

where $\Gamma > 0$ is the step size, and $\nabla f_{i,k}$ and $r_{i,k}$ are the gradient of $f_{i,k}$ and $\rho_{i,k}$ written respectively as

$$\begin{aligned} \nabla f_{i,k} &= \mathbf{X}_i^\intercal\Omega_i^{-1}(\mathbf{X}_i\mathbf{w}_{i,k} - \mathbf{y}_i), \text{ and} \\ r_{i,k} &= \sum_{j\neq i} \alpha_{i,j}(\mathbf{w}_{i,k} - \mathbf{w}_{j,k})/\mathrm{tr}(\Omega_j). \end{aligned} \qquad (8)$$

The update in (7) assumes nodes exchange their current estimates with their neighbors, i.e., node i receives $\{\mathbf{w}_{j,k} : \alpha_{i,j} = 1, j \in \mathcal{V}\}$, as per the gradient in (8).

In the following, we analyze the convergence properties by focusing on the continuous time stochastic approximation of SGN.

A. Continuous time approximation

We embed the discrete-time process in (7) into a continuous-time domain. Let $\Delta t_{(i,k)}$ be the random time needed by node i to calculate $\nabla f_{i,k}$ and $\nabla\rho_{i,k}$ and to complete the update from $\mathbf{w}_{i,k}$ to $\mathbf{w}_{i,k+1}$. We assume that $\Delta t_{(i,k)}$'s are i.i.d. with $\mathrm{E}[\Delta t_{(i,k)}] = \Delta t$ and $\mathbf{w}_{i,k}$

is obtained at time $t_{(i,k)} = \sum_{l<k} \Delta t_{(i,l)}$. The process $\{\mathbf{w}_{i,t} : t > 0\}$ is defined as follows: $\mathbf{w}_{i,t} \triangleq \mathbf{w}_{i,k}$, if $t \in [t_{(i,k)}, t_{(i,k+1)})$. Then the corresponding continuous expression of (7) is as follows:

$$\mathbf{w}_{i,t_{(i,k+1)}} = \mathbf{w}_{i,t_{(i,k)}} - \Gamma(\nabla f_{i,t_{(i,k+1)}} + \lambda \nabla \rho_{i,t_{(i,k+1)}}) \quad (9)$$

To simplify the notation we set $w_{i,t} := w_{i,t}/\Gamma$. We rewrite the scheme (9) in the form of the summation of previous steps, and approximate the noise terms by standard m-dimensional Brownian motions and the rest by integrals. Then $d\mathbf{w}_{i,t}$ can be approximated by the differential form of a stochastic Ito integral—see Section 5.2 in [19] for a detailed derivation. The continuous time dynamics of $\mathbf{w}_{i,t}$ is as follows:

$$\begin{aligned} d\mathbf{w}_{i,t} = &- \gamma(g_{i,t} + \lambda r_{i,t})dt \\ &+ \gamma \mathbf{X}_i^\intercal \Omega_i^{-1} \Big(\tau_i dB_{i,t} + \varsigma \Lambda_i dB_t \Big), \end{aligned} \quad (10)$$

where $g_{i,t} := \mathbf{X}_i^\intercal \Omega_i^{-1} \mathbf{X}_i (\mathbf{w}_{i,t} - \mathbf{w}^*)$ and $r_{i,t}$ is defined as in (8) with k replaced by t; $\gamma = 1/\Delta t$, $\tau_i = \sigma_i \sqrt{\Gamma \Delta t}$, and $\varsigma = \sqrt{\Gamma \Delta t}$. Here $B_{i,t}$ and B_t are the standard m dimensional Brownian Motion approximating the local noise associated with node i and the common noise, respectively. We note that in deriving (10), we do not assume a specific distribution, e.g., Gaussian. Our results follow by central limit theorem, which means the approximation holds for general error distributions.

We make use of the following definitions and relations to characterize the convergence of (10).

B. Preliminaries

We define the Laplacian matrix of $\mathcal{G}$ as $L = \hat{\Delta} - A$, where $\hat{\Delta}$ is a diagonal matrix whose ith diagonal entry is equal to the degree of ith node and $A = [\alpha_{i,j}/\text{tr}(\Omega_i)\text{tr}(\Omega_j)]_{i,j}$ is the corresponding adjacency matrix. Let a_2 be the second smallest eigenvalue of L.

The continuous-time gradient $g_{i,t}$ defined above is a function of $\mathbf{w}_{i,t}$. In our analyses, we denote $g_{i,t}(\hat{\mathbf{w}}_t) = \mathbf{X}_i^\intercal \Omega_i^{-1} \mathbf{X}_i (\hat{\mathbf{w}}_t - \mathbf{w}^*)$. Note that $g_{i,t}(\mathbf{w}^*) = 0$ for all $i \in \mathcal{V}$ and t, to simplify notation we will write $g(\mathbf{w}^*)$ instead. Similarly, when a property holds for all t, we drop t and write $g_{i,t}$ as g_i.

We note that g_i's are μ-Lipschitz continuous and the corresponding loss function is strongly convex with κ. To see this, we note that Ω_i^{-1} is positive definite, and can be expressed as $\Omega_i^{-1} = P^\intercal P$, where P is the matrix resulting from the eigendecomposition. Let $\mathbf{w}_{i,1}$ and $\mathbf{w}_{i,2}$ be two input vectors taken from the function domain, then

$$\begin{aligned} \|g_i(\mathbf{w}_{i,1}) - g_i(\mathbf{w}_{i,2})\| &= \left\| \mathbf{X}_i^\intercal \Omega_i^{-1} \mathbf{X}_i (\mathbf{w}_{i,1} - \mathbf{w}_{i,2}) \right\| \\ &\leq \mu \left\| \mathbf{w}_{i,1} - \mathbf{w}_{i,2} \right\|, \end{aligned}$$

where $\mu = \|P\mathbf{X}_i\|_F$ and $\|\cdot\|_F$ is the Frobenius norm. Furthermore,

$$\begin{aligned} &(g_i(\mathbf{w}_{i,1}) - g_i(\mathbf{w}_{i,2}))^\intercal (\mathbf{w}_{i,1} - \mathbf{w}_{i,2}) \\ &\quad = \|P\mathbf{X}_i(\mathbf{w}_{i,1} - \mathbf{w}_{i,2})\|^2 \geq \kappa \|\mathbf{w}_{i,1} - \mathbf{w}_{i,2}\|^2, \end{aligned} \quad (11)$$

for some $0 < \kappa < \|P\mathbf{X}_i\|_F^2$. Note that g_i is strongly convex with κ and the corresponding loss function is Lipschitz continuous with constant μ.

C. Regularity and Consistency

To characterize convergence we define measures for *regularity* and *consistency*. Let $\hat{\mathbf{w}}_t$ denote the weighted average solution at time t, i.e.,

$$\hat{\mathbf{w}}_t = \frac{1}{v} \sum_{i=1}^N \frac{\mathbf{w}_{i,t}}{\text{tr}(\Omega_i)}, \quad (12)$$

where $v = \sum_{i=1}^N 1/\text{tr}(\Omega_i)$ is a normalization constant. We also refer to the above solution as the *ensemble model*. Let $V_{i,t} = \|\mathbf{e}_{i,t}\|^2/2$, where $\mathbf{e}_{i,t} := \mathbf{w}_{i,t} - \hat{\mathbf{w}}_t$. Regularity is defined as the weighted sum of the differences between the ensemble model and the solution at each node:

$$\bar{V}_t = \frac{1}{v} \sum_{i=1}^N \frac{\|\mathbf{w}_{i,t} - \hat{\mathbf{w}}\|^2}{2\text{tr}(\Omega_i)} = \frac{1}{v} \sum_{i=1}^N \frac{V_{i,t}}{\text{tr}(\Omega_i)}.$$

Consistency is the distance between the ensemble model and the ground truth,

$$U_t = \frac{1}{2} \|\hat{\mathbf{w}}_t - \mathbf{w}^*\|^2.$$

Via standard algebra, we have

$$\frac{1}{2v} \sum_{i=1}^N \frac{\|\mathbf{w}_{i,t} - \mathbf{w}^*\|^2}{\text{tr}(\Omega_i)} = \bar{V}_t + U_t. \quad (13)$$

In what follows, we obtain upper bounds on the expectations of regularity $\bar{V}_t$ and consistency U_t processes in Theorems 1 and 2, respectively. Given (13), these bounds provide a bound on the average error of individual estimates generated by the SGN algorithm with respect to the ground truth $\mathbf{w}^*$.

D. Convergence: Regularity

The following result provides an upper bound on the expected regularity of the estimates at a given time.

Theorem 1 *Let* $\mathbf{w}_{i,t}$ *evolve according to* (10)*. Then*

$$\mathrm{E}[\bar{V}_t] \leq \frac{\gamma C_1}{2(\kappa + \lambda a_2)} + (\bar{V}_0 - \frac{\gamma C_1}{2(\kappa + \lambda a_2)}) e^{-2(\kappa + \lambda a_2)\gamma t}, \quad (14)$$

where C_1 *is the summation of constant terms,*

$$C_1 = \frac{1}{2v} \sum_{i=1}^N \frac{1}{tr(\Omega_i)} C_{1,i}, \quad (15)$$

with $C_{1,i}$ for $i \in \mathcal{V}$ defined as,

$$C_{1,i} = \tau_i^2\Big(1 - \frac{2}{v tr(\Omega_i)}\Big)\left\|\mathbf{X}_i^\intercal \Omega_i^{-1}\right\|_F^2 + \frac{\varsigma^2}{v^2}\sum_{k=1}^{N}\sum_{j=1}^{N}\frac{1}{tr(\Omega_k)tr(\Omega_j)}\mathbf{1}^\intercal(\mathbf{X}_k^\intercal\Omega_k^{-1}\Lambda_k \circ \mathbf{X}_j^\intercal\Omega_j^{-1}\Lambda_j)\mathbf{1} - \frac{2\varsigma^2}{v}\sum_{k=1}^{N}\frac{1}{tr(\Omega_k)}\mathbf{1}^\intercal(\mathbf{X}_i^\intercal\Omega_i^{-1}\Lambda_i \circ \mathbf{X}_k^\intercal\Omega_k^{-1}\Lambda_k)\mathbf{1} + \varsigma^2\left\|\mathbf{X}_i\Omega_i^{-1}\Lambda_i\right\|_F^2 + \frac{1}{v^2}\sum_{k=1}^{N}\frac{\tau_k^2}{tr(\Omega_k)^2}\left\|\mathbf{X}_k^\intercal\Omega_k^{-1}\right\|_F^2. \tag{16}$$

In the long run, $\lim_{t\to\infty}\mathrm{E}[\bar{V}_t] \leq \frac{\gamma C_1}{2(\kappa+\lambda a_2)}$.

Proof: See Appendix V. ∎

It is not surprising that the expected difference in estimates in (14) decreases with growing λ which penalizes disagreement with neighbors. Similarly, the larger the algebraic connectivity of the network a_2 or the strong convexity constant κ is, the smaller is the expected $\bar{V}_t$.

Finally, the constant term C_1 is determined by data $\mathbf{X}$ and the matrices Λ_i. In particular, C_1 is small when we have nodes that are less affected by the noise. Intuitively, with increasing network size, nodes with less exposure to noise are given increasing weight which then increases regularity across estimates.

E. Convergence: Consistency

The consistency measure $\{U_t, t \geq 0\}$ captures the performance of the average solution $\hat{\mathbf{w}}$. The following theorem provides a characterization of the performance of the collective effort.

Theorem 2 *Let $\mathbf{w}_{i,t}$ evolve according to* (10)*. Then*

$$\mathrm{E}[U_t] \leq e^{-2\kappa\gamma t}U_0 + \frac{\gamma}{2\kappa}\Big(\frac{\mu-\kappa}{\lambda a_2}C_1 + C_2\Big)\Big(1 - e^{-2\kappa\gamma t}\Big),$$

where C_1 is defined in (15)-(16) *and*

$$C_2 = \frac{1}{2v^2}\left(\sum_{k=1}^{N}\frac{\tau_k^2}{tr(\Omega_k)^2}\left\|\mathbf{X}_k\Omega_k^{-1}\right\|_F^2 + \varsigma^2\sum_{k=1}^{N}\sum_{j=1}^{N}\frac{1}{tr(\Omega_k)tr(\Omega_j)}\mathbf{1}^\intercal(\mathbf{X}_k^\intercal\Omega_k^{-1}\Lambda_k \circ \mathbf{X}_j^\intercal\Omega_j^{-1}\Lambda_j)\mathbf{1}\right)$$

with "$\circ$" denoting the Hadamard product. In the long run,

$$\lim_{t\to\infty}\mathrm{E}[U_t] \leq \frac{\gamma}{2\kappa}\left(\frac{\mu-\kappa}{\lambda a_2}C_1 + C_2\right).$$

The proof (see [19] for details) follows a similar outline as Theorem 1. We apply Ito's Lemma to get the stochastic dynamics form of dU_t and then introduce an auxiliary variable $W_t = U_t + \frac{\mu-\kappa}{\lambda a_2}\bar{V}_t$. We obtain dW_t in a similar fashion as that of dU_t. Then we use the μ-Lipschitz continuity of the gradient g_i and the properties of the Laplacian matrix to obtain an upper bound for dW_t. By integrating and taking the expectation of the bound, we obtain the desired upper bound for $\mathrm{E}(U_t)$ since $\mathrm{E}[U_t] \leq \mathrm{E}[W_t]$.

Similar to the regularity measure bound, the penalty constant λ and the algebraic connectivity a_2 reduce the bound on the expected consistency. However, the long-run expected difference between the collective estimate and the ground truth does not reduce to zero as $\lambda a_2 \to \infty$. Indeed, we cannot expect the collective performance to improve above a given level by increasing connectivity or increasing regularity among different models. The constant C_2, determined by the data $\mathbf{X}$ and matrices Λ_i, captures the performance gap in the long run due to available data. According to C_2, we can only improve performance by the addition of new nodes that have access to more reliable data.

IV. Numerical Implementation

We consider a real-world problem of predicting the activity of individuals from head movement data GLEAM [18]. The data contains 2-hour head motions of 38 participants' activities recorded by Google glass. We use the records of 37 participants as the training dataset and the remaining one as the test. There are $96,829$ data points in the training set and $2,617$ in the test set. We are only interested in two types of activities: eating and working (including study and operating electrical devices, e.g., iPad, computer, and phone). We denote "eating" as 0 and "working" as 1, and the response variable activity (act) is binary. The activity "eating" constitutes 20% of the entries in the training set and 10% of that in the test set. We use 18 predictors to represent the readings (gyroscope, accelerometer, magnetic field, rotation, linear acceleration, and gravity) from the 3 axis of the glass sensors. We use 37 computing nodes and construct a N-node complete network (lower connectivity may reduce the prediction accuracy slightly) in the following experiment.

A. SGN with Mini-Batch and Unknown Covariance

We use the mini-batch process to approximate the gradient at each step and set the mini-batch size as 100.

The covariance matrix is unknown and need to be estimated by each node. In this example, it is computed as the diagonals of the empirical covariance matrix of 50 mini-batch samples. We use a fading memory update rule to compute the trace of the covariance matrix $\mathrm{tr}(\Omega_i)$ ([20]):

$$\mathrm{tr}(\Omega_{i,k+1}) = \varphi\mathrm{tr}(\Omega_{i,k}) + (1-\varphi)\mathrm{tr}(\hat{\Omega}_{i,k+1}),$$

where $\text{tr}(\hat{\Omega}_{i,k+1})$ is the i-th covariance matrix trace computed at the $(k+1)$-th iteration, and $\varphi \in (0,1)$ is the fading parameter that controls the memory of the past covariance values.

B. Numerical results

We set the fading parameter $\varphi = 0.9$, the step size $\Gamma = 300$, and the regularization penalty $\lambda = 100$.

We denote node i's estimate with $\mathbf{w}_i = [w_0, \ldots, w_{18}]$, and the activity estimate of the i-th node is given by

$$\widehat{act}_i = \mathbf{X}_i \mathbf{w}_i.$$

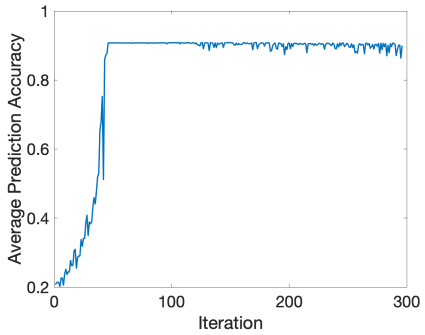
At step k, we predict $\widehat{act}_j = 0$ if $\mathbf{x}_j^\intercal \hat{\mathbf{w}}_k < 0.5$ and $\widehat{act}_j = 1$ otherwise. The prediction accuracy of SGN is given by

$$\text{accuracy}_k = 1 - \frac{\left\| \mathbf{y}_{\text{test}} - \mathbf{1}_{\{\mathbf{X}_{\text{test}} \hat{\mathbf{w}}_k > 0.5\}} \right\|^2}{m},$$

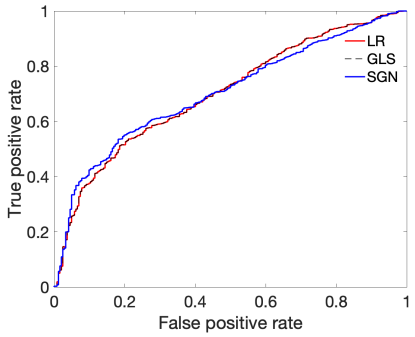
where $(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$ is the test set, $\hat{\mathbf{w}}_k$ is the weighted estimation (12) from SGN, and $\mathbf{1}(\cdot)$ is the indicator function. We define the average prediction accuracy as the mean prediction accuracy at step k over 10 runs. Figure 1(a) shows that the average prediction accuracy of SGN converges after 50 iterations. At the final step $T = 300$, the average prediction accuracy of SGN (0.8972) is close to that of LR (0.9079) and GLS (0.8991). Figure 1(b) presents the Receiver operating characteristic (ROC) curve of the three approaches. The Area under the curve (AUC) of SGN is 0.7037, and GLS and LR have the same AUC at 0.7014. Though this example is a binary classification problem and potentially violate the assumption in (4), our approach has comparable classification accuracy to LR, which suggests that SGN is robust.

V. Conclusion

The ever-increasing dimension and the size of data have introduced new challenges to centralized estimation. For example, limited bandwidth in current networking infrastructure may not satisfy the demands for transmitting high-volume datasets to a central location. Hence, it is of interest to study alternatives to centralized estimation. In this paper, we considered a distributed architecture for learning a linear model via generalized least squares by relying on a network of interconnected "local" learners. In the proposed distributed scheme, each local learner is assigned a dataset, and *asynchronously* implements stochastic gradient updates based upon a sample (or a mini-batch sample). To ensure robust estimation, a network regularization term that penalizes models with high *local* variability is used. Unlike other model averaging schemes based upon a synchronized step, the proposed scheme implements local model averaging continuously and asynchronously.



(a)



(b)

Figure 1. (a) SGN average prediction accuracy at each iterations. (b) The ROC curve of LR, GLS, and SGN.

We provided finite-time performance guarantees on the consistency of the ensemble model. We illustrated the robustness of the proposed method in the detection of activities from head movement data.

Appendix

In the following, we first provide the differential form of the regularity measure (Lemma 1), and then obtain an upper bound of $d\bar{V}_t$. By integrating and taking the expectation of the upper bound we obtain desired result.

Lemma 1 *The regularity measure $\bar{V}_t$ satisfies*

$$d\bar{V}_t = -\frac{\gamma}{v} \sum_{i=1}^{N} \frac{1}{tr(\Omega_i)} g_{i,t}^\intercal \mathbf{e}_{i,t} dt + \gamma K_1 d\tilde{B}_t + \gamma^2 C_1 dt - \frac{\lambda\gamma}{v} \sum_{i=1}^{N} \frac{1}{tr(\Omega_i)} r_{i,t}^\intercal \mathbf{e}_{i,t} dt, \tag{17}$$

where $K_1 d\tilde{B}_t$ is the summation of Ito terms,

$$K_1 \tilde{B}_t = \frac{1}{v} \sum_{i=1}^{N} \frac{1}{tr(\Omega_i)} K_{1,i} d\tilde{B}_t,$$

with $K_{1,i} d\tilde{B}_t$ for $i \in \mathcal{V}$ defined as,

$$K_{1,i} d\tilde{B}_t = \varsigma \mathbf{X}_i^\intercal \Omega_i^{-1} \Lambda_i dB_t^\intercal \mathbf{e}_{i,t} + \tau_i \left(\mathbf{X}_i^\intercal \Omega_i^{-1} \right) dB_{i,t}^\intercal \mathbf{e}_{i,t}.$$

Proof: See [19] for the proof. ∎

A. Proof of Theorem 1

Consider the first term of (17), let $h_t = \min_{i\in\mathcal{V}} g_{i,t}(\hat{\mathbf{w}}_t)$. We can add a zero-valued term $(h_t^{\intercal}/v)\sum_{i=1}^{N}\mathbf{e}_{i,t}/\mathrm{tr}(\Omega_i)$ to the equation, where we define $\mathbf{e}_{i,t} := \mathbf{w}_{i,t} - \hat{\mathbf{w}}_t$. By the strong convexity of g_i in (11), we can obtain the following inequality,

$$-\frac{1}{v}\sum_{i=1}^{N}\frac{1}{\mathrm{tr}(\Omega_i)}g_{i,t}^{\intercal}\mathbf{e}_{i,t} = -\frac{1}{v}\sum_{i=1}^{N}\frac{1}{\mathrm{tr}(\Omega_i)}(g_{i,t}-h_t)^{\intercal}\mathbf{e}_{i,t} \leq -\kappa\frac{1}{v}\sum_{i=1}^{N}\frac{\|\mathbf{e}_{i,t}\|^2}{\mathrm{tr}(\Omega_i)} = -2\kappa\bar{V}_t.$$

Now we consider the last term in (17). Define the vector $\mathbf{e}_t = [\mathbf{e}_{1,t}^T,\ldots,\mathbf{e}_{N,t}^T]^T$ and the matrix $\hat{L} = L\otimes I_m$, where $\otimes$ is the Kronecker product. Using these definitions, we can express it as follows,

$$-\sum_{i=1}^{N}\frac{1}{\mathrm{tr}(\Omega_i)}\sum_{j=1,j\neq i}^{N}\frac{\alpha_{ij}}{\mathrm{tr}(\Omega_j)}(\mathbf{w}_{i,t}-\mathbf{w}_{j,t})^T\mathbf{e}_{i,t} = \sum_{i=1}^{N}\sum_{j\neq i}\frac{-\alpha_{ij}}{\mathrm{tr}(\Omega_i)\mathrm{tr}(\Omega_j)}(\mathbf{e}_{i,t}-\mathbf{e}_{j,t})^T\mathbf{e}_{i,t} = -\mathbf{e}_t^T\hat{L}\mathbf{e}_t, \tag{18}$$

where the first equality follows by adding and subtracting $\hat{\mathbf{w}}_t$ and the second equality is by the definition of $\hat{L}$. Note that the second largest eigenvalue a_2 satisfies $\min_{x\neq 0,\,1^Tx=0}(x^TLx)/\|x\|^2 = a_2$ [21]. Thus, we have

$$-\mathbf{e}_t^T\hat{L}\mathbf{e}_t \leq -a_2\sum_{i=1}^{N}\|\mathbf{e}_{i,t}\|^2. \tag{19}$$

Combining (18) and (19), an upper bound for $d\bar{V}_t$ follows :

$$d\bar{V}_t \leq -2\gamma(\kappa+\lambda a_2)\bar{V}_t dt + \gamma^2 C_1 dt + \gamma K_1 d\tilde{B}_t \tag{20}$$

Next we consider the derivative of $e^{2(\kappa+\lambda a_2)\gamma t}\bar{V}_t$ by applying chain rule and substituting in the inequality for (20), we obtain an upper bound. By integrating both sides of the obtained inequality, we obtain

$$\bar{V}_t \leq e^{-2(\kappa+\lambda a_2)\gamma t}\bar{V}_0 + \frac{\gamma C_1}{2(\kappa+\lambda a_2)}(1-e^{-2(\kappa+\lambda a_2)\gamma t}) + e^{-2(\kappa+\lambda a_2)\gamma t}\int_0^t e^{2(\kappa+\lambda a_2)\gamma s}K_1 d\tilde{B}_s. \tag{21}$$

Since the stochastic integral is a martingale,

$$E\Big[\int_0^t e^{2(\kappa+\lambda a_2)\gamma s}K_1 d\tilde{B}_s\Big] = 0.$$

We obtain the desired upper bound by taking the expectation on both sides of (21). In the long run, as $t\to\infty$, the exponential terms will vanish, and the upper bound of the regularity measure follows.

REFERENCES

[1] D. Hallac, J. Leskovec, and S. Boyd, "Network lasso: Clustering and optimization in large graphs," *Proceedings SIGKDD*, pp. 387–396, 2015.

[2] A. Jung, T. Nguyen, and A. Mara, "When is network lasso accurate?" *Frontiers in Applied Mathematics and Statistics*, vol. 3, pp. 1–11, 2018.

[3] M. Yamada, T. Koh, T. Iwata, S.-T. J., and S. Kaski, "Localized lasso for high- dimensional regression," *Conference on Artificial Intelligence and Statistics, AISTATS*, pp. 325–333, 2017.

[4] R. Nassif, S. Vlaski, and A. H. Sayed, "Learning over multitask graphs-part i: Stability analysis," *arXiv preprint arXiv:1805.08535*, 2018.

[5] B. Hansen, "Least squares model averaging," *Econometrica*, vol. 75, no. 4, pp. 1175–1189, 2007.

[6] Q. Liu, R. Okui, and A. Yoshimura, "Generalized least squares model averaging," *Econometric Reviews*, vol. 35, no. 8, pp. 1692–1752, 2016.

[7] J. Mendes-Moreira, C. Soares, A. Jorge, and J. Freire de Sousa, "Ensemble approaches for regression: A survey," *ACM Computing Surveys*, vol. 45, no. 1, pp. 10–40, 2012.

[8] J. M. Bates and C. M. W. Granger, "The combination of forecasts," *Operations Research Quaterly*, vol. 20, pp. 451–468, 1969.

[9] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates," *Journal of Machine Intelligence Research*, vol. 16, pp. 3299–3340, 2015.

[10] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.

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

[12] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2011.

[13] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.

[14] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, pp. 944–966, 2015.

[15] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[16] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[17] C. He, T. Xie, Y. Zhengyu, Z. Hu, and S. Xia, "Federated multi-task learning with decentralized periodic averaging sgd," 2019. [Online]. Available: https://fl.chaoyanghe.com/

[18] S. A. Rahman, C. Merck, Y. Huang, and S. Kleinberg, "Unintrusive eating recognition using google glass," in *2015 9th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. IEEE, 2015, pp. 108–111.

[19] L. Hong, A. Garcia, and C. Eksin, "Distributed estimation via network regularization," *arXiv preprint arXiv:1910.12783*, 2019.

[20] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[21] C. Godsil and G. Royle, "Algebraic graph theory," *Springer, New York*, 2001.