Finite Sample Guarantees for Distributed Online Parameter Estimation with Communication Costs

Lei Xin, George Chiu, Shreyas Sundaram

Abstract—We study the problem of estimating an unknown parameter in a distributed and online manner. Existing work on distributed online learning typically either focuses on asymptotic analysis, or provides bounds on regret. However, these results may not directly translate into bounds on the error of the learned model after a finite number of time-steps. In this paper, we propose a distributed online estimation algorithm which enables each agent in a network to improve its estimation accuracy by communicating with neighbors. We provide non-asymptotic bounds on the estimation error, leveraging the statistical properties of the underlying model. Our analysis demonstrates a trade-off between estimation error and communication costs. Further, our analysis allows us to determine a time at which the communication can be stopped (due to the costs associated with communications), while meeting a desired estimation accuracy. We also provide a numerical example to validate our results.

I. INTRODUCTION

Learning an accurate model from data is an important problem in many fields [1], including machine learning, economics, and control theory. However, in many cases, the available datasets are usually split among multiple agents/learners and come in a streaming manner, which require online processing. Coordination among the various agents to process their data also comes with a communication cost, and thus algorithms must be designed to balance the amount of communication with the speed and accuracy of learning.

The problem of distributed learning/optimization has been studied extensively over the last few decades [2], [3]. These papers typically provide theoretical guarantees on the convergence of local solutions to the optimizer of the sum of local functions over the network. When it comes to distributed online parameter estimation, the existing literature typically focuses on proving asymptotic convergence of the estimate to the true value [4], [5]. There is another branch of research on distributed online learning that focuses on providing bounds on regret, which is defined as the difference between the costs generated by the sequence of local decisions and the true optimal costs obtained in hindsight [6]–[8]. The bound on regret can be used as an appropriate metric to evaluate a proposed algorithm, as a sublinear regret implies that the algorithm performs as well as its centralized counterpart on average (over time). However, it is unclear how such bounds can be translated into the bounds on the accuracy of the learned model after a finite number of time-steps. The paper [9] studies distributed state estimation problem with finite time convergence guarantee with a fixed observation matrix, and under Byzantine faults. In contrast, we consider the problem where the observation/feature matrix is random, which is often encountered in general machine learning problems.

In this paper, we propose a distributed online parameter estimation algorithm in a networked setting, which enables each agent to improve its estimation accuracy by communicating with neighbors in the network. Our algorithm can be viewed as an extension of the distributed least squares method in [10] to an online setting. In our algorithm, each agent stores two estimates of the true parameter: one computed purely based on local data and one computed after communicating with neighbors in the network. We provide finite time (or sample) upper bounds on the estimation errors of both of these two estimates, which highlight the role of communication. Our results demonstrate a trade-off between estimation error and communication costs. To balance such a trade-off, we discuss how we can leverage our finite time error bounds to determine a time at which the communication can be stopped (due to the costs associated with maintaining communications), while meeting a desired estimation accuracy. We also provide a numerical example to validate our results.

II. NOTATION

Vectors are taken to be column vectors unless indicated otherwise. Let $\mathbb{R}$ and $\mathbb{Z}$ denote the set of real numbers and integers, respectively. Let $I_n$ denote a vector of dimension $n$ with all of its elements equal to 1. Let $\sigma_{\text{min}}(\cdot)$ and $\lambda_{\text{min}}(\cdot)$ be the smallest singular value and eigenvalue in magnitude, respectively, of a given matrix. The eigenvalues of a given matrix are ordered with nonincreasing magnitude, i.e., $|\lambda_1(\cdot)| \geq \cdots \geq |\lambda_{\text{min}}(\cdot)|$. For a given matrix $A$, we use $A(i,j)$ to denote the element in its $i$-th row and $j$-th column. $A^*$ to denote its conjugate transpose, $A^t$ to denote its pseudoinverse, and $\text{vec}(A)$ to denote its vectorization (i.e., the vector obtained by stacking the columns of $A$ starting from the left). We use $\|A\|$, $\|A\|_1$ and $\|A\|_F$ to denote the spectral norm, 1-norm, and Frobenius norm, respectively, of matrix $A$. We use $I_n$ to denote the identity matrix with dimension $n$. An $n$-dimensional Gaussian distributed random vector is denoted as $u \sim \mathcal{N}(\mu, \sigma^2 I_n)$, where $\mu$ is the mean and $\sigma^2 I_n$ is the covariance matrix. The symbol $\cap$ is used to...
denote the intersection of sets. We use the symbol mod to denote the modulo operation.

III. PROBLEM FORMULATION

Consider a group of $m$ agents $\mathcal{V}$ interconnected over an undirected and connected graph $G = (\mathcal{V}, E)$. An edge $(i, j) \in E$ is an unordered pair, which indicates a bidirectional communication link between agents $i$ and $j$. Let $\mathcal{N}_i \triangleq \{j : (i, j) \in E\}$ be the set of neighbors of agent $i$. The goal of these agents is to collaboratively estimate an unknown parameter $\Theta \in \mathbb{R}^{l \times n}$ with finite time guarantees, under a finite number of communication steps. At each time step $t = 1, 2, \ldots$, each agent $i \in \mathcal{V}$ gathers the data pair $(x_{i,t}, y_{i,t})$ generated by the following model

$$y_{i,t} = \Theta x_{i,t} + \eta_{i,t},$$

where $y_{i,t} \in \mathbb{R}$ is the label vector, $x_{i,t} \in \mathbb{R}^n$ is the feature vector, and $\eta_{i,t} \in \mathbb{R}$ is the noise. We make the following assumption.

**Assumption 1:** The feature vector $x_{i,t}$ and noise $\eta_{i,t}$ are Gaussian random vectors that are independent over time and agents, where $x_{i,t} \sim \mathcal{N}(\mu_{i,t}, \sigma_{i,t}^2 I_n)$ and $\eta_{i,t} \sim \mathcal{N}(0, \sigma_\eta^2 I_1)$. The mean $\mu_{i,t} \in \mathbb{R}^n$ is deterministic with $\sum\{||\mu_{i,t}|| : i \in \mathcal{V}, t \in \mathbb{Z}_{\geq 1}\} = \bar{\mu} \in \mathbb{R}_{>0}$.

The above model can be used to capture many problems. For example, it can be used to capture the problem of dynamical system identification via multiple independent trajectories (assuming zero initial condition and without process noise), where $y_{i,t}$ is the output of the system in each trajectory, $x_{i,t}$ is the input applied in each trajectory, and $\Theta$ is the Markov parameter matrix of the system, e.g., [11]. We note that the $x_{i,t}$ considered in our model allows for time- and agent-dependent mean $\mu_{i,t}$, and hence is more general than the analogous system identification problem, which typically considers zero-mean Gaussian inputs. Also we note that our algorithm does not require any parameters of the model to be known in advance. However, we assume that there are known upper bounds on $\sigma_x, \sigma_y, \bar{\mu}, ||\Theta||$, and there is a known non-zero lower bound on $\sigma_x$. These bounds will facilitate the design of certain user-specified parameters in our algorithm, which will become clear when we present our results.

**Remark 1:** One may observe that a trivial solution to the above problem might be to not communicate at all, i.e., each agent only updates based on its local dataset. However, such a solution does not leverage the distributed nature of the problem, which provides each agent with the potential to speed up the learning by communicating with the other agents in the network. On the other hand, communications with the other agents should be carefully designed, as information from others might become less useful when each agent already has a good estimate based on the information it has so far. In the sequel, we study a distributed algorithm that leverages the communication network, which allows all agents to learn the model efficiently (when some upper/lower bounds on $\sigma_x, \sigma_y, \bar{\mu}, ||\Theta||$ are available). More specifically, the algorithm allows every agent to hold an estimate with an estimation error comparable to that of the centralized solution throughout time, while saving communication costs.

IV. A DISTRIBUTED ONLINE ESTIMATION ALGORITHM

In this section, we describe a two-time-scale distributed algorithm. At each time step $t = 1, 2, \ldots$, based on its local dataset, each agent $i \in \mathcal{V}$ wishes to solve the following least squares problem:

$$\min_{\Theta \in \mathbb{R}^{l \times n}} \sum_{j=1}^{t} ||y_{i,j} - \hat{\Theta} x_{i,j}||^2_F.$$  \hspace{1cm} (2)

The least squares local estimate for agent $i$, given its samples collected up to time step $t$, is

$$\hat{\Theta}_{i,t+1} = (\sum_{j=1}^{t} y_{i,j} x_{i,j}^*) (\sum_{j=1}^{t} x_{i,j} x_{i,j}^*)^{-1},$$ \hspace{1cm} (3)

assuming the matrix $\sum_{j=1}^{t} x_{i,j} x_{i,j}^*$ is invertible.

The above estimate can be updated iteratively with the arrival of new data pair $(x_{i,t}, y_{i,t})$, through

$$\alpha_{i,t+1} = \alpha_{i,t} + y_{i,t} x_{i,t}^*, \quad \beta_{i,t+1} = \beta_{i,t} + x_{i,t} x_{i,t}^*, \quad \hat{\Theta}_{i,t+1} = \alpha_{i,t+1} \beta_{i,t+1}^{-1} \hspace{1cm} (4)$$

where $\alpha_{i,1} = 0$, $\beta_{i,1} = 1$. Note that $\beta_{i,t+1}^{-1} = \beta_{i,t+1}^{-1}$ once $\beta_{i,t+1}$ becomes invertible. Also, $\beta_{i,t+1}^{-1}$ can be updated iteratively using the Sherman-Morrison formula [12], which states $\beta_{i,t+1}^{-1} = \beta_{i,t}^{-1} - \frac{\beta_{i,t}^{-1} x_{i,t} x_{i,t}^* \beta_{i,t}^{-1}}{1 + x_{i,t}^* \beta_{i,t}^{-1} x_{i,t}}$.

The algorithm enters the communication phase when the conditions $t \mod \zeta = 0$ and $t \leq S$ are satisfied, where $\zeta \in \mathbb{Z}_{>1}$ and $S \in \mathbb{Z}_{\geq 0}$, i.e., when the current time step $t$ is an integer multiple of the pre-specified communication period $\zeta$ and is less than the pre-specified stopping time $S$. Letting the superscript $k$ denote the current communication time step, each agent $i \in \mathcal{V}$ sends its current $\alpha_{i,t+1}^k$ and $\beta_{i,t+1}^k$ to its neighbors $j \in \mathcal{N}_i$, and receives $\alpha_{j,t+1}^k$ and $\beta_{j,t+1}^k$ from $j \in \mathcal{N}_i$. The update is given by

$$\alpha_{i,t+1} = W(i, i) \alpha_{i,t+1}^k + \sum_{j \in \mathcal{N}_i} W(i, j) \alpha_{j,t+1}^k, \quad \beta_{i,t+1} = W(i, i) \beta_{i,t+1}^k + \sum_{j \in \mathcal{N}_i} W(i, j) \beta_{j,t+1}^k, \hspace{1cm} (5)$$

for $k = 0, 1, \ldots, T-1$, where $T \in \mathbb{Z}_{\geq 1}$ is the number of pre-specified total communication steps whenever the algorithm enters the communication phase, and $W \in \mathbb{R}^{m \times m}$ is the matrix where $W(i, j)$ is the weight agent $i \in \mathcal{V}$ assigns to agent $j \in \mathcal{V}$. We make the following assumption on $W$, which is commonly used, e.g., [13].

**Assumption 2:** The weight matrix $W \in \mathbb{R}^{m \times m}$ associated with the communication graph $G = (\mathcal{V}, E)$ is assumed to satisfy: (1) $W(i, j) \in \mathbb{R}$ and $W(i, j) \geq 0$ for all $i, j \in \mathcal{V}$, and $W(i, j) = 0$ if $j \notin \mathcal{N}_i$ and $i \neq j$; (2) $W1m = 1m$; (3) $W = W^*$ and (4) $\rho(W) \triangleq \max\{\lambda_2(W), -\lambda_m(W)\} < 1$. 

5981
The local estimate after communication is set to be \( \Theta_{i,t+1} = \alpha_{i,t+1}^T (\beta_{i,t+1}^T)^\dagger \). If there is no communication happened at the current time step \( t \), agent \( i \) just keeps its estimate from the previous time-step, i.e., \( \Theta_{i,t+1} = \Theta_{i,t} \).

The above steps are encapsulated in Algorithm 1.

Algorithm 1 Distributed Online Estimation Algorithm

Input: Weight matrix \( W \), stopping time \( S \), communication period \( \zeta \), number of communication steps \( T \)

1. Each \( v_i \in V \) initializes \( \alpha_{i,1} = 0, \beta_{i,1} = 0, \Theta_{i,1} = 0 \)
2. for \( t = 1, 2, 3, \ldots \) do
3. for \( v_i \in V \) do
   > Implement in parallel
4. Gather the data pair \( (x_{i,t}, y_{i,t}) \), where \( x_{i,t} \sim N(\mu_{i,t}, \sigma_i^2 I_n) \)
5. Update \( \Theta_{i,t+1} = \alpha_{i,t+1} (\beta_{i,t+1})^\dagger \) as in (4)
6. if \( t \mod \zeta = 0 \) and \( t \leq S \) then
7. Set \( \Theta_{i,t+1} = \Theta_{i,t+1} \)
8. for \( k = 0, 1, \ldots, T-1 \) do
9. Broadcast \( \Theta_{i,t+1} \) to \( j \in N_i \), and receive \( \Theta_{j,t+1} \) from \( j \in N_i \)
10. Update \( \Theta_{i,t+1} \) as in (5)
11. end for
12. \( \Theta_{i,t+1} = \alpha_{i,t+1}^T (\beta_{i,t+1}^T)^\dagger \)
13. else
14. \( \Theta_{i,t+1} = \Theta_{i,t} \)
15. end if
16. end for
17. end for

Remark 2: Note that Algorithm 1 has two time scales. In practice, this captures the scenario where agents can communicate multiple times between receiving data samples. Further, note that both \( \Theta_{i,t+1} \) (without communication) and \( \Theta_{i,t+1} \) (after communication) are estimates of the true parameter \( \Theta \). In the next section, we will provide bounds on the finite time communication errors \( \|\Theta_{i,t+1} - \Theta\| \) and \( \|\Theta_{i,t+1} - \Theta\| \). In practice, one could choose the estimate with smaller (estimated) error bound as the “true” output of the algorithm. In section VI, we will discuss how to choose the user-specified parameters \( \zeta, S \) and \( T \) to enable efficient learning.

V. ANALYSIS OF THE ERROR

In this section, we provide finite time upper bounds of the local estimation error without communication, the global estimation error, and the local estimation error after communication. Due to space constraints, the proofs of some of the following results are provided in [14].

A. Local Estimation Error Without Communication

We will start with bounding the estimation error using only local samples. Note that for any agent \( i \in V \), we have

\[
\|\Theta_{i,t+1} - \Theta\| = \|\alpha_{i,t+1} \beta_{i,t+1}^{-1} - \Theta\| = \|\sum_{j=1}^t y_{i,j} x_{i,j}^* (\sum_{j=1}^t x_{i,j} x_{i,j}^*)^{-1} - \Theta\|
\]

assuming the the matrix \( \sum_{j=1}^t x_{i,j} x_{i,j}^* \) is invertible. The proof of our error bound follows by upper bounding the above terms separately, as follows. The proofs of these two results, given in [14], rely on [15, Corollary 5.35] and [16, Lemma A.1] respectively.

Lemma 1: Let Assumption 1 hold. Fix \( \delta > 0 \) and let \( t \geq \delta_1 t_c(t_1, t_2) \), where \( t_1 = 8n + 16 \log \frac{\delta}{\sigma} \), \( t_2 = \left( \frac{16 \mu(\sqrt{4n} + \sqrt{2 \log \frac{\delta}{\sigma}})}{\sigma_x} \right)^2 \).

We have the following result.

Theorem 1: Let Assumption 1 hold. Fix \( \delta > 0 \) and let \( t \geq t_3 = 2(n + l) \log \frac{9}{\delta} \).

We have the following result.

Remark 3: Theorem 1 shows that the error is \( O(\frac{1}{\sqrt{n}}) \). Note that when the mean \( \mu_{i,j} \) is non-zero but invariant for all \( t \), the bound could become more conservative when \( n > 1 \) (since that makes \( \mu \) in \( C_1 \) larger, but we still have \( \lambda_{\min}(I_n + \mu_{i,t}) = 1 \)). If this is known in advance, one could define a new pair of sequences \( \tilde{y}_{i,t} = y_{i,t} - y_{i,t-1} \) and \( \tilde{x}_{i,t} = x_{i,t} - x_{i,t-1} \).
One then has \( \hat{y}_{i,t} = \Theta \hat{x}_{i,t} + \hat{\eta}_{i,t} \), where \( \hat{x}_{i,t} \sim N(0, 2\sigma^2 \hat{\Theta} \in) \) and \( \hat{\eta}_{i,t} \sim N(0, 2\sigma^2 \hat{\eta} \in) \). The same bound will still apply to the least squares solution using the transformed dataset, i.e., with the price of reducing the amount of samples by one-half, one could force the mean-dependent terms in Theorem 1 to go to zero. Such a transformation could result in a smaller bound when \( \hat{\mu} \) is large enough.

### B. Global Estimation Error

Next, we look at the estimation error of the least squares estimate supposing that one has access to all samples across the network up to time step \( t \). The global estimate and its associated estimation error are

\[
\hat{\Theta}_{t+1} \triangleq \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{T} y_{i,t} x_{i,t}^* \left( \sum_{i=1}^{m} \sum_{t=1}^{T} x_{i,t} x_{i,t}^* \right)^{-1},
\]

\[
\Vert \hat{\Theta}_{t+1} - \Theta \Vert = \Vert \left( \sum_{i=1}^{m} \sum_{t=1}^{T} \eta_{i,t} x_{i,t}^* \right) \left( \sum_{i=1}^{m} \sum_{t=1}^{T} x_{i,t} x_{i,t}^* \right)^{-1} \Vert,
\]

assuming the matrix \( \sum_{i=1}^{m} \sum_{t=1}^{T} x_{i,t} x_{i,t}^* \) is invertible. The proof of the forthcoming theorem entirely follows Theorem 1 due to Assumption 1, with slight adjustments to accommodate possibly different means of \( x_{i,n} \) across the network.

**Theorem 2:** Let Assumption 1 hold. Fix \( \delta > 0 \), and let \( t \geq \frac{1}{m} \max(t_1, t_2, t_3) \), where \( t_1 = 8n + 16 \log \frac{2}{\delta}, t_2 = (\frac{16\mu(4\sqrt{t_1} + 2\log \frac{2}{\delta})}{m} + 2) \), and \( t_3 = 2(n + 1) \log \frac{1}{\delta} \). Letting \( \bar{\mu}_t = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{T} \mu_{i,t} \), we have probability at least \( 1 - 4\delta \),

\[
\Vert \hat{\Theta}_{t+1} - \Theta \Vert \leq \frac{C_1}{\sqrt{m} \sigma^2 \lambda_{\min}(I_n + \bar{\mu}_t)},
\]

where \( C_1 \) is defined in Theorem 1.

**Remark 4:** Theorem 2 indicates that the global estimation error bound is approximately \( \frac{1}{m} \) of the local estimation error bound for agent \( i \in V \) in Theorem 1 (when \( \mu_t \) in Theorem 2 is approximately equal to \( \bar{\mu}_t \) in Theorem 1). Next, we will analyze the local estimation error after finite communication steps, which shows how communication could help agents benefit from the global dataset.

### C. Local Estimation Error After Communication

To derive the error bound of the local estimate after communication, we first define some quantities for notational simplicity. Recall the roles of the stopping time \( S \) and the communication period \( \varsigma \) in Algorithm 1. For \( t \) satisfying \( t \mod \varsigma = 0 \) and \( t \leq S \) (note that we will only consider such \( t \) in this section), define \( \hat{\alpha}_{t+1} \triangleq \frac{1}{m} \sum_{i=1}^{m} \alpha_{i,t+1} \) and \( \hat{\beta}_{t+1} \triangleq \frac{1}{m} \sum_{i=1}^{m} \beta_{i,t+1} \). For any \( i \in V \), note that

\[
\Vert \hat{\Theta}_{i,t+1} - \Theta \Vert = \Vert \alpha_{i,t+1}^T (\beta_{i,t+1}^T)^{-1} - \Theta \Vert \\
= \Vert \alpha_{i,t+1}^T (\beta_{i,t+1}^T)^{-1} - \alpha_{i,t+1}^T \hat{\beta}_{i,t+1}^T + \hat{\alpha}_{i,t+1}^T \hat{\beta}_{i,t+1}^T - \Theta \Vert \\
\leq \Vert \alpha_{i,t+1}^T (\beta_{i,t+1}^T)^{-1} - \alpha_{i,t+1}^T \hat{\beta}_{i,t+1}^T + \hat{\alpha}_{i,t+1}^T \hat{\beta}_{i,t+1}^T - \Theta \Vert,
\]

under the invertibility assumption.

The second portion of the above inequality can be bounded using Theorem 2, since \( \Vert \alpha_{t+1}^T (\beta_{t+1}^T)^{-1} - \Theta \Vert = \Vert \hat{\Theta}_{t+1} - \Theta \Vert \). Now we will focus on bounding the first term, which corresponds to the error due to network convergence at time step \( t \), using \( T \) steps of communication. For \( t \) satisfying \( t \mod \varsigma = 0 \) and \( t \leq S \), fixing \( i \in V \) and defining \( \epsilon_{i,t+1} \triangleq \alpha_{i,t+1}^T - \alpha_{t+1} \\
and \epsilon_{(\hat{\alpha}_{i,t+1})^{-1}} \triangleq (\beta_{i,t+1}^T)^{-1} - \beta_{i,t+1}^{-1} \), we have

\[
\Vert \alpha_{i,t+1}^T (\beta_{i,t+1}^T)^{-1} - \alpha_{t+1}^T \beta_{t+1}^{-1} \Vert = \Vert (\alpha_{i,t+1} + \epsilon_{i,t+1}) (\beta_{i,t+1}^{-1} + \epsilon_{(\beta_{i,t+1})^{-1}}) - \alpha_{t+1}^T \beta_{t+1}^{-1} \Vert \\
\leq \Vert \alpha_{i,t+1} \epsilon_{(\beta_{i,t+1})^{-1}} \Vert + \Vert \epsilon_{i,t+1} \beta_{i,t+1}^{-1} \Vert + \Vert \epsilon_{(\beta_{i,t+1})^{-1}} \epsilon_{(\beta_{i,t+1})^{-1}} \Vert \\
\leq \Vert \alpha_{i,t+1} \Vert \Vert \epsilon_{(\beta_{i,t+1})^{-1}} \Vert + \Vert \epsilon_{i,t+1} \Vert \Vert \beta_{i,t+1}^{-1} \Vert \\
+ \Vert \epsilon_{(\beta_{i,t+1})^{-1}} \Vert \Vert \epsilon_{(\beta_{i,t+1})^{-1}} \Vert,
\]

and we will bound the above terms separately.

Before we proceed, we will define some probabilistic events. Let \( t \) satisfy \( t \mod \varsigma = 0 \) and \( t \leq S \). Fix \( \delta > 0 \). With the replacement of \( \delta \) by \( \hat{\delta} \), let \( t \geq 2 \max(t_1, t_2, t_3) \) defined in Lemma 1 and Lemma 2. Let \( E_1 \) be the event such that the event in Lemma 1 occurs for all \( i \in V \) at time step \( t \), i.e.,

\[
E_1 \triangleq \bigcap_{i=1}^{m} \left\{ \left\{ \sum_{j=1}^{t} x_{i,j} x_{i,j}^* \leq t (\frac{19}{8} \sigma^2 + \mu^2) \right\} \cap \left\{ \lambda_{\min}(\sum_{j=1}^{t} x_{i,j} x_{i,j}^*) \geq \frac{t \sigma^2 \delta}{8} \lambda_{\min}(I_n + \bar{\mu}_t) \right\} \right\},
\]

where \( \bar{\mu}_t = \frac{1}{m} \sum_{i=1}^{m} \mu_{i,t} \). Similarly, let \( E_2 \) be the event such that the event in Lemma 2 occurs for all \( i \in V \) at time step \( t \), i.e.,

\[
E_2 \triangleq \bigcap_{i=1}^{m} \left\{ \left\{ \sum_{j=1}^{t} \eta_{i,j} x_{i,j}^* \right\} \leq \sqrt{4 \sigma^2 (n + l) \log \frac{2}{\delta}} + \mu (\sqrt{2(n + l)} + \sqrt{2 \log \frac{2}{\delta}}) \right\}.
\]

Applying a union bound over all \( i \in V \), we have

\[
E_3 \triangleq E_1 \cap E_2
\]

occurs with probability at least \( 1 - 4m \hat{\delta} \).

We have the following results.

**Proposition 1:** Conditioning on event \( E_3 \) in (11), we have

\[
\Vert \hat{\alpha}_{i,t+1} \Vert \leq t c_1 + \sqrt{t c_2},
\]

where \( c_1 \triangleq \Vert \Theta \Vert (\frac{19}{8} \sigma^2 + \mu^2), c_2 \triangleq \sigma^2 (4 \sigma^2 (n + l) \log \frac{2}{\delta} + \mu (\sqrt{2(n + l)} + \sqrt{2 \log \frac{2}{\delta}})) \).

**Proposition 2:** Let Assumption 2 hold. Conditioning on event \( E_3 \) in (11), for all \( i \in V \), we have

\[
\Vert \epsilon_{(\hat{\alpha}_{i,t+1})^{-1}} \Vert \leq m \hat{\sigma} \sqrt{t (\rho(W))^T (t c_1 + \sqrt{t c_2})},
\]

where \( c_1 \) and \( c_2 \) are defined in Proposition 1.

**Proposition 3:** Conditioning on event \( E_3 \) in (11), we have

\[
\Vert \beta_{i,t+1}^{-1} \Vert \leq \frac{8}{\sigma^2 \delta}.
\]
Proposition 4: Let Assumption 2 hold. Conditioning on event $E_3$ in (11), for all $i \in \mathcal{V}$, we have
\[
\|\epsilon(\bar{\beta}^T_{i,t+1} \cdot 1) - \bar{\Theta}_{i,t+1} - \Theta\| \leq \frac{(\rho(W))^T C_3}{t},
\]
where $C_3 = \frac{152n^2 \sqrt{2m}}{\sigma_x^2} + \frac{64n^2 \sqrt{2m}}{\rho^2} \hat{\mu}^2$.

Now we are ready to bound the local estimation error after communication.

Theorem 3: Let Assumptions 1 and 2 hold. Fix $\delta > 0$ and let $t \geq \max(t_1, t_2, t_3)$, where $t_1 = 8n + 16 \log \frac{\delta}{\rho}, t_2 = \frac{16\mu(\sqrt{4n + \sqrt{2m}})}{\sigma_x}$, $t_3 = 2(n + l) \log \frac{1}{\delta}$. For $t \mod \zeta = 0$ and $t \leq S$, fix $\delta > 0$ and denote $\hat{\mu}_t = \frac{4}{m t \sigma_x^2} \sum_{i=1}^{m} \sum_{j=1}^{t} \mu_{i,j}^{d, t, j}$. We have with probability at least $1 - 4m \delta - 4\delta$,
\[
\|\hat{\Theta}_{i,t+1} - \Theta\| \leq \frac{(\rho(W))^T C_0}{C_1} + \frac{\sqrt{m t \sigma_x^2\min(I_n + \hat{\mu}_t)}}{C_1},
\]
for all $i \in \mathcal{V}$, where $C_0 = c_3(c_1 + t^{-1/2}c_2) + \frac{8n^2 \sqrt{\zeta (c_1 + t^{-1/2}c_2)}}{\sigma_x} + (\rho(W))^T C_3 \sqrt{\zeta (c_1 + t^{-1/2}c_2)}$, and $\mu_1, c_1, c_2, c_3$ are defined in Theorem 1, Proposition 1 and Proposition 4.

Proof: Recall the decomposition of error from (9) and (10). Note that the event $E_3$ in (11) occurs with probability at least $1 - 4m \delta$ when $t \geq \max(t_1, t_2, t_3)$. Combine event $E_3$ and the event in Theorem 2 using a union bound. Applying Propositions 1, 2, 3 and 4, we get the desired result.

Remark 5: Theorem 3 demonstrates a trade-off between estimation error and communication costs. By choosing $\delta$ small, as $T$ tends to infinity, the first term in the bound tends to zero, and agent $i \in \mathcal{V}$ can almost recover the same performance guarantee as if it had access to all samples across the network up to time step $t$ (note that the second term in the error bound reduces the local estimation error bound in Theorem 1 by approximately $\frac{1}{\sqrt{m}}$). The speed at which the first term goes to zero depends on the network topology. Further, this result implies that by choosing $T$ large such that the first term in the bound is small, communication becomes less important as $t$ increases (i.e., as each agent keeps collecting samples), since the second term goes to zero more slowly. Consequently, the improvements of the new local estimate after communication over the old estimates $\hat{\Theta}_{i,t+1}$ and $\hat{\Theta}_{i,t+1}$ will become smaller. In the next section, we discuss how to choose those user specified parameters to balance the trade-off between estimation error and communication costs, leveraging the above observation.

VI. DETERMINING THE COMMUNICATION PERIOD, THE STOPPING TIME, AND THE NUMBER OF COMMUNICATION STEPS

In short, the communication can be stopped when the minimum between the largest local error bound (over agents) in Theorem 1 and the error bound in Theorem 3 is less than some pre-specified threshold value $\epsilon \in \mathbb{R}_{>0}$. To achieve that, one needs to first specify the communication period $\zeta$. Note that larger $\zeta$ corresponds to sparser communication. Further, one can specify how much error at most due to network convergence in Theorem 3 (first term in (12)) can be tolerated, denoted as $\epsilon_N \in \mathbb{R}_{>0}$. Smaller $\epsilon_N$ would require more communication steps. Based on that, one can compute the number of communication steps $T$ that makes the error due to network convergence in Theorem 3 always less than $\epsilon_N$. Consequently, one can then evaluate the bounds in Theorem 1 and Theorem 3 and determine the stopping time $S$. Note that the bounds in Theorem 1 and Theorem 3 involve parameters that may be unknown in practice. However, it suffices to replace $\hat{\mu}_t, \hat{\mu}_i, t$ by $0$, and $\sigma_x, \sigma_y, \hat{\mu}, \|\Theta\|$ by their corresponding estimated upper/lower bounds.

Although communication could still help to reduce estimation error after $t > S$, even infinite communication steps can only allow each agent to recover the same estimation error bound as if it had access to the global dataset, under which the reduction of error could be negligible in practice when $\|\hat{\Theta}_{i,t+1} - \Theta\|$ or $\|\hat{\Theta}_{i,t+1} - \Theta\|$ is already small enough. Consequently, it might be preferable for these agents to start updating purely based on local data, considering the communication costs. We will illustrate this idea in the next section empirically.

VII. NUMERICAL EXPERIMENT

In this example, we consider a network of $m = 6$ agents trying to learn model (1), where
\[
\Theta = \begin{bmatrix} 1.6 & 0.3 \\ 0.8 & 0.3 \end{bmatrix}, \sigma_x = 3, \sigma_y = 1,
\]
and $\mu_{i,t} = 0$ for all $i$ and $t$. The weight matrix associated with the communication graph is
\[
W = \begin{bmatrix} 1/3 & 1/3 & 0 & 0 & 0 & 1/3 \\ 1/3 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}.
\]

We set $\zeta = 20$ and assume that all parameters in Theorem 1 and Theorem 3 are known for simplicity. The number of communication steps is set to $T = 38$, which is computed based on the guidelines suggested in Section VI such that the error due to network convergence in Theorem 3 is always less than $0.01$ (using $\delta = 0.001$). The communication is stopped when the smallest error bound between the one in Theorem 1 (using $\delta = 0.05$) and the one in Theorem 3 (using $\delta = 0.05, \delta = 0.001$) is less than 0.5, which leads to $S = 1620$. We plot the average (over agents) local estimation error without communication $\|\hat{\Theta}_{i,t+1} - \Theta\|$, the average local estimation error after communication $\|\hat{\Theta}_{i,t+1} - \Theta\|$, and the global estimation error $\|\hat{\Theta}_{t+1} - \Theta\|$. All results are averaged over 10 independent runs.
As expected, the error \( \|\hat{\Theta}_{t+1} - \Theta\| \) is almost the same as \( \|\bar{\Theta}_{i,t+1} - \Theta\| \) when communication happens. Further, the error \( \|\hat{\Theta}_{t+1} - \Theta\| \) decreases relatively rapidly, and is much smaller than \( \|\bar{\Theta}_{i,t+1} - \Theta\| \) at the beginning. However, the error \( \|\bar{\Theta}_{i,t+1} - \Theta\| \) decreases more slowly, and its improvement over \( \|\hat{\Theta}_{t+1} - \Theta\| \) becomes smaller, as each agent gathers more samples. Although the communication is stopped at \( t = 1620 \), leveraging the global dataset has only marginal improvements over the estimates \( \hat{\Theta}_{t+1}, \bar{\Theta}_{i,t+1} \) after \( t = 1620 \), implying communication becomes less important, which confirms our observation in Theorem 3. On the other hand, although the minimum between \( \|\hat{\Theta}_{t+1} - \Theta\| \) and \( \|\bar{\Theta}_{i,t+1} - \Theta\| \) is less than 0.5 after \( t = 1620 \), the simulation also implies that our finite time bound is conservative. It is of interest to develop tighter bounds in future work.

![Graph](image_url)

Fig. 1: Average \( \|\hat{\Theta}_{i,t+1} - \Theta\| \), average \( \|\bar{\Theta}_{i,t+1} - \Theta\| \), and \( \|\hat{\Theta}_{t+1} - \Theta\| \). The communication is stopped after \( t = 1620 \).

### VIII. CONCLUSION AND FUTURE WORK

In this paper, we proposed an online distributed parameter estimation algorithm with finite time performance guarantees. Our results demonstrate a trade-off between estimation error and communication costs, and we show that one can leverage the error bounds to determine a time at which the communication can be stopped. We believe our results can be extended to more general graph conditions, e.g., leveraging [17, Proposition 1]. Future work could focus on developing similar bounds by using gradient-based methods.

### REFERENCES

[1] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data.* AMLBook New York, 2012, vol. 4.

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

[3] M. Zhu and S. Martinez, “On distributed convex optimization under inequality and equality constraints,” *IEEE Transactions on Automatic Control,* vol. 57, no. 1, pp. 151–164, 2011.

[4] S. Kar, J. M. Moura, and K. Ramanan, “Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication,” *IEEE Transactions on Information Theory,* vol. 58, no. 6, pp. 3575–3605, 2012.

[5] Q. Zhang and J.-F. Zhang, “Distributed parameter estimation over unreliable networks with markovian switching topologies,” *IEEE Transactions on Automatic Control,* vol. 57, no. 10, pp. 2545–2560, 2012.

[6] S. Hosseini, A. Chapman, and M. Mesbahi, “Online distributed optimization via dual averaging,” in *52nd IEEE Conference on Decision and Control.* IEEE, 2013, pp. 1484–1489.

[7] ——, “Online distributed convex optimization on dynamic networks,” *IEEE Transactions on Automatic Control,* vol. 61, no. 11, pp. 3545–3550, 2016.

[8] D. Yuan, A. Proutiere, and G. Shi, “Distributed online linear regressions,” *IEEE Transactions on Information Theory,* vol. 67, no. 1, pp. 616–639, 2020.

[9] L. Su and S. Shahrampour, “Finite-time guarantees for byzantine-resilient distributed state estimation with noisy measurements,” *IEEE Transactions on Automatic Control,* vol. 65, no. 9, pp. 3758–3771, 2019.

[10] R. Tron and R. Vidal, “Distributed computer vision algorithms through distributed averaging,” in *CVPR 2011.* IEEE, 2011, pp. 57–63.

[11] Y. Sun, S. Oymak, and M. Fazel, “Finite sample system identification: Optimal rates and the role of regularization,” in *Proc. Learning for Dynamics and Control Conference,* 2020, pp. 16–25.

[12] H. V. Henderson and S. R. Searle, “On deriving the inverse of a sum of matrices,” *SIAM Review,* vol. 23, no. 1, pp. 53–60, 1981.

[13] L. Ye and S. Sundaram, “Distributed maximization of submodular and approximately submodular functions,” in *2020 59th IEEE Conference on Decision and Control (CDC).* IEEE, 2020, pp. 2979–2984.

[14] L. Xin, G. Chiu, and S. Sundaram, “Finite sample guarantees for distributed online parameter estimation with communication costs,” *arXiv preprint,* 2022.

[15] R. Vershynin, “Introduction to the non-asymptotic analysis of random matrices,” *arXiv preprint arXiv:1011.3027,* 2010.

[16] S. Oymak and N. Ozay, “Non-asymptotic identification of lti systems from a single trajectory,” *arXiv preprint arXiv:1806.05722,* 2018.

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