FINE: A Framework for Distributed Learning on Incomplete Observations for Heterogeneous Crowdsensing Networks

Luoyi Fu[1], Songjun Ma[2], Lingkun Kong[1], Shiyu Liang[2], Xinbing Wang[1,2]
Dept. of {Computer Science and Engineering[1], Electronic Engineering[2]}
Shanghai Jiao Tong University, Shanghai, China
Email: {yiluofu,masongjun,klk316980786,lsy18602808513,xwang8}@sjtu.edu.cn

***Abstract*—In recent years there has been a wide range of applications of crowdsensing in mobile social networks and vehicle networks. As centralized learning methods lead to unreliabitlity of data collection, high cost of central server and concern of privacy, one important problem is how to carry out an accurate distributed learning process to estimate parameters of an unknown model in crowdsensing. Motivated by this, we present the design, analysis and evaluation of FINE, a distributed learning Framework for Incomplete-data and Non-smooth Estimation. Our design, devoted to develop a feasible framework that efficiently and accurately learns the parameters in crowdsensing networks, well generalizes the previous learning methods in that it supports heterogeneous dimensions of data records observed by different nodes, as well as minimisation based on non-smooth error functions. In particular, FINE uses a novel *Distributed Record Completion* algorithm that allows each node to obtain the global consensus by an efficient communication with neighbours, and a *Distributed Dual Average* algorithm that achieves the efficiency of minimizing non-smooth error functions. Our analysis shows that all these algorithms converge, of which the convergence rates are also derived to confirm their efficiency. We evaluate the performance of our framework with experiments on synthetic and real world networks.**

I. Introduction

Recently, there emerge massive applications of crowdsensing/participatory sensing in mobile social networks and vehicle networks [1]–[7]. The crowd acquire some (potentially high dimensional) data from the environment and each user in the crowd can exploit the cooperatively acquired data to perform a learning process for an accurate estimation of the parameters of some specific models. This, in turn, leads to an accurate prediction of future events and correct decision of the following action.

In this paper, we aim to address the issue of the accurate learning in the undirected-static-random crowdsensing networks. In order to solve this problem, there have been various proposed approaches [8]–[10] whose learning processes are usually formulated as optimizations of the total training error, likelihood function and *etc* [11] (*e.g.,*, liner regression, support vector machines or expectation-maximization [12]). However, these methods usually employ centralized learning algorithms, which leads to three major problems. First, in real world crowdsensing settings, mobile devices are likely to be located over an enormous space, which makes it both energy consuming and prone to error for central server collecting data from all mobile devices, especially those who are distributed far away from the server. Second, dealing with large volume of data by centralized algorithms requires an expensive high-configuration data center that possesses huge memory for data storage and processing. Third, managing data by central servers make the private information of users more likely to be exposed to the adversary [13]–[15], which might cause severe information leakage.

The three problems above imply the necessity of a distributed realization of parameter learning in crowdsensing environments. However, when applying existing distributed learning methods to our scenarios, restrictions of two characteristics in the common distributed framework spawn additional problems. To illustrate, first, for mathematical tractability, the error function is usually assumed to be *smooth* and convex for the design of efficient algorithms; while as the emerging crowdsensing applications may incorporate different properties, the training error functions may be *non-smooth* in nature [16], [17]. For instance, in distributed detections [18], source intensity functions may be non-smooth, resulting in non-smoothness in training error as well. Second, the common framework requires each terminal to acquire a set of *complete* records, *i.e.,* each record with data elements in all dimensions to ensure the accuracy of the learning process. This implicitly assumes that the terminals are *homogeneous* in functionalities so that each of them should record the same types of signals (*e.g.,* each mobile phone can record traveling speed, waiting time as well as ambient noise at every position). In contrast, it is impossible for each terminal to record full-dimension data in the crowdsensing applications. For example, one mobile phone can only be responsible for data acquisition at its own position, leaving the observation of elements in other positions (i.e., dimensions) a job of other mobile phones. Moreover, mobile phones may hold different types of sensors, and therefore are unable to acquire all kinds of signals.

We are thus motivated to propose a distributed learning Framework of Incomplete-data and Non-smooth Estimation (FINE), which aims to exhibit high compatibility to learning applications in crowdsensing environments. There are two major challenges in the design: 1) It is difficult for each node to supplement the unknown dimensions of the observed vector,

especially when users hesitate to upload the acquired data to the central server for privacy concerns.[1] 2) Due to the huge amount of data collected and the high dimensionality of each record, it is often required that the learning process should be operated by each terminal in a distributed fashion. The efficiency of minimizing a single non-smooth function is notoriously low [19], yet a distributed processing is even more challenging due to intricate interdependency among the multiple non-smooth optimizations processed by each respective terminal.

We overcome the difficulties above by designing two algorithms in FINE. First, we design a Distributed Record Completion (DRC) algorithm to allow each node to obtain global consensus. Specifically, each terminal consistently obtains incomplete record and completes the missing elements from its neighbors. The combined successive observation and consensus design ensure that each node can acquire unbiased and accurate multidimensional global parameters in spite of the originally fragmentary inputs. Second, we design a novel Distributed Dual Average (DDA) algorithm to solve the non-smooth convex optimization problems with efficiency. To sum up, our major contributions are listed as follows:

- We propose FINE, a novel framework addressing a class of distributed learning problems in heterogeneous crowdsensing networks. FINE is robust to observation noises, capable of handling fragmentary data inputs as well as non-smooth objective functions, and efficient to solve distributed learning problems with a convergence rate[2] of $\mathcal{O}(\frac{\log\sqrt{|\mathcal{V}|}}{1-C})$.
- We design two important algorithms in FINE: a DRC algorithm to ensure each node to acquire complete information based on its incomplete data acquisition, and a DDA algorithm to solve non-smooth convex optimizations with efficiency.
- We formally prove the convergence of the above two algorithms, and further derive their convergence rates. We provide the insights on the relationship between the convergence rate and the network topology, and reveal important design principles for such networks.

The rest of the paper is organized as follows. In Section II, we provide problem description and the design insights of our framework FINE. In section III, we propose two important algorithms and demonstrate the implementation of such a framework. Section IV presents a detailed analysis of proposed algorithms. Section V provides detailed proofs of lemmas and theorems. We give performance evaluation of our framework in Section VI and literature review in Section VII. Section VIII is contributed to the concluding remarks.

II. System Model and Problem Formulation

In this section, we first present the system model and then formulate our problems, followed by giving insights of our algorithm design.

A. System Model

We consider a heterogeneous sensor network described by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of $|\mathcal{V}|$ nodes. We use $\mathcal{V}$ and $\mathcal{E}$ to denote the set of nodes and edges, respectively. The undirected graph implies that the sensor network allows each pair of connected sensors can communicate in bi-direction.

The heterogeneity refers to the fact that terminals observe different dimensions of data. We let $\boldsymbol{y}_k(i)$ denote an incomplete data vector observed by each terminal k at time i. For practical consideration, we allow the observation to include noise $\varepsilon_k(i)$. Assume that $\boldsymbol{y}^*$ is an M-dimensional complete correct data vector, the incomplete observations of the k-th agent, $\boldsymbol{y}_k(i)$, could be expressed by a linear mapping, H_k, from the global data vector $\boldsymbol{y}^*$:

$$\boldsymbol{y}_k(i) = H_k \boldsymbol{y}^* + \boldsymbol{\varepsilon}_k(i) \in \mathbb{R}^M, \tag{1}$$

where H_k is a matrix in $\mathbb{R}^{M\times M}$.

Each terminal requires the full dimensional data to conduct learning process. Assume that terminals have various error functions f_k. We allow each local error function to be non-smooth. The local training error ϵ_k at node k is a function of global tunable parameter vector $\boldsymbol{\theta}$ and global data $\boldsymbol{y}$, *i.e.,* $\epsilon_k = f_k(\boldsymbol{\theta}, \boldsymbol{y})$, which is convex and non-smooth on $\boldsymbol{\theta}$.

B. Problem Formulation

In consideration of the above settings, we should first guarantee that each node can obtain an accurate estimation $\hat{\boldsymbol{y}}$ on the global data vector. This encourages us to design the DRC algorithm. In short, the goal of the DRC is to make each node k hold the complete estimation which satisfies:

$$\lim_{i\to\infty} \hat{\boldsymbol{y}}_k(i) = \boldsymbol{y}^*.$$

Then, with the estimations and the settings of error functions, we find the optimal parameter $\boldsymbol{\theta}^*$ to minimize the total training error:

$$\boldsymbol{\theta}^* := \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{|\mathcal{V}|} f_k(\boldsymbol{\theta}, \hat{\boldsymbol{y}}),\ \text{s.t., } \boldsymbol{\theta} \in \Theta, \hat{\boldsymbol{y}} \in \mathcal{Y},$$

where $\Theta \subset \mathbb{R}^Q, \mathcal{Y} \subset \mathbb{R}^M$ are both convex sets. This encourage us to design the DDA algorithm. Note that the solution needs information exchange between a node and its neighbors, which involves the consideration of the network topology.

To sum up, our formulation deals with the non-smooth learning. We extend the previous dual averaging approach [20], [21] by taking the incomplete observation and observation noise into account.

Remark: As a contrast, the traditional formulation [8]–[10], [22], [23] considers the smooth optimization problem based on a homogeneous sensor network. The homogeneity implies that each agent k should record the same types of signals, and

[1]The privacy concern of information leakage is not new in crowdsensing networks, which often stems from the centralized management of crowdsensing systems [13]–[15]. Therefore, users might be reluctant to upload the acquired data to the central server..

[2]$|\mathcal{V}|$ is the number of agents and $1-C$ represents the spectral gap of the network.

full dimensional observations $\boldsymbol{y}_k$. The learning problem is to find $\boldsymbol{\theta}^*$ which minimizes the summation of the local training error:

$$\boldsymbol{\theta}^* := \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{|\mathcal{V}|} f(\boldsymbol{\theta}, \boldsymbol{y}_k), \text{s.t.}, \boldsymbol{\theta} \in \Theta, \boldsymbol{y}_k \in \mathcal{Y},$$

where Θ and $\mathcal{Y}$ are both convex sets. $\boldsymbol{\theta}$ is a globally tunable parameter vector and the function f is assumed to be convex and smooth on the parameter $\boldsymbol{\theta}$.

C. Algorithm Design Insights

In this section, we briefly describe the insights of algorithm design and explain how FINE can efficiently and accurately deal with the challenging distributed learning problems in heterogeneous crowdsensing networks. FINE uses a Distributed Record Completion (DRC) algorithm (Section III-A) to ensure each node to obtain global consensus in spite of the incomplete local observations and a Distributed Dual Average (DDA) algorithm (Section III-B) to efficiently solve the non-smooth optimization problems.

1) DRC Algorithm: Our DRC allows each node to communicate with each other to complete the data and achieve consensus, on the condition that each node's successive observations are required. This combined successive observation and consensus design ensure that each node can acquire unbiased and accurate multidimensional global parameters $\boldsymbol{y}^*$. On the one hand, if nodes cooperate but only process the initial noisy observations (*i.e.,* only $\{\boldsymbol{y}_k(0)\}$ are employed), as in traditional consensus [24]–[26], they all converge to the average of initial estimates which might result in severe bias. Therefore, the requirement of both successive observations and consensus in DRC ensures that each node converges to an unbiased and accurate estimate of the global vector $\boldsymbol{y}^*$.

Remark: The communication overhead, which quantifies the weight of communication among sensors or mobile phone users during reaching to a consensus, is an important performance metric of consensus-based learning algorithms [27]–[30]. However, in our paper, we are interested in measuring the *time* it takes for the whole network to discover the convergence of optimization, i.e., the optimal parameter $\boldsymbol{\theta}^*$. Here the time refers to communication complexity in reaching convergence of optimization, while neglecting local computation complexity inside mobile devices as it is highly determined by the sensors required in a crowdsensing task and is considerably small – can be completed during clock ticks.

2) DDA Algorithm: A well-known sub-gradient method [31] used for solving a single non-smooth optimization has a time efficiency of $\mathcal{O}(\frac{1}{\varepsilon^2})$, where ε is the tolerable error. For solving distributed non-smooth optimization problem consisting of multiple interdependent functions, the efficiency of the sub-gradient would be $\mathcal{O}(\frac{|\mathcal{V}|^3}{\varepsilon^2})$ [21], [32], [33], where $|\mathcal{V}|$ is the network size. This order does not reflect the explicit influence of the network topology, but only reflects the worst case of efficiency. References [23], [34], [35] provide the convergence rate of the optimization error under some special network topologies. Aiming at the general network topology with the network size $|\mathcal{V}|$ and networking topological parameter C, literature [36]'s method can achieve the convergence rate of the error ε with $\mathcal{O}(\frac{\log|\mathcal{V}|}{1-C})$.

In the context of crowdsensing, more practical factors, *i.e.,* partial observation and observation noise, need to be considered, and these factors will lead to the increase of the time complexity. Therefore, it is worthwhile to propose a non-smooth algorithm to improve the efficiency. Towards this end, we build our algorithm on an efficient non-smooth optimization methodology called the *dual averaging method* introduced in [20]. The dual averaging in nature is an efficient non-smooth convex optimization method. By realizing the dual averaging method into a distributed manner, as will be prescribed in later sections, we will show that our algorithm can return a consistent estimate $\boldsymbol{\theta}^*$ for each node. Additionally, as we will also provably demonstrate later, the convergence rate of our distributed algorithm can also achieve $\mathcal{O}(\frac{\log\sqrt{|\mathcal{V}|}}{1-C})$.

Remark: The DDA algorithm, as we will unfold in sequel, is designed by extending the centralized dual averaging method into a distributed form, requiring that each node performs local information exchange that follows a weighting process (where each edge is assigned a weight). Thus, intuitively the process is heavily correlated with the network topology. Compared to [36], the DDA algorithm provides consistent efficiency despite that it is applied to solve more complicated distributed learning problems that take practical factors like partial observation and observation noise into account.

Up till now, we have presented our problem, and articulate all insights which inform FINE's design. In the sequel, we will detail the design of DRC and DDA in FINE.

III. Algorithms

In this section, we describe the details of the design of DRC and DDA algorithms used in FINE. We firstly introduce the notations. Let E_n denote $n \times n$ identity matrix; let $\mathbf{1}_n$, $\mathbf{0}_n$ denote the column vectors of ones and zeros, defined in $\mathbb{R}^n$ respectively; let $\|\cdot\|$ denote the standard Euclidean 2-norm for a vector and the induced 2-norm for matrixes, equavelent to the matrix spectral radius for symmetric matrixes. Besides, we use $\|\cdot\|_*$ to denote the dual norm to $\|\cdot\|$,defined by $\|u\|_* := \sup_{\|v\|}\langle u, v\rangle$, which refer to the value of the linear function $u \in X^*$ at $v \in X$ (X is a vector space and X^* is its dual space). We further let $\mathbb{P}[\cdot]$ and $\mathbb{E}[\cdot]$ denote the probability and the expectation operators, respectively. We use the symbol $\otimes$ to denote the Kronecker product manipulation, commonly used in matrix manipulations. For instance, the Kronecker product of the $n \times n$ matrix L and E_m is an $nm \times nm$ matrix, denoted by $L \otimes E_m$.

In the undirected graph $\mathcal{G}$ we consider, we denote the neighbor of an arbitrary node $v \in \mathcal{V}$ by $N_v = \{u \in \mathcal{V} | e_{uv} \in \mathcal{E}\}$, the number of edges incident to v by the degree $d_v = |N_v|$ of node v, and the degree matrix by $\mathcal{D} = \text{diag}(d_1, ..., d_{|\mathcal{V}|})$. Then, we use an adjacent matrix, $\mathcal{A} = [\mathcal{A}_{uv}]_{|\mathcal{V}|\times|\mathcal{V}|}$, to describe the network connectivity. We set $\mathcal{A}_{uv} = 1$, if $e_{uv} \in \mathcal{E}$; or 0

otherwise. Further, we define the graph Laplacian matrix[3] $\mathcal{L}$ as $\mathcal{L} = \mathcal{D} - \mathcal{A}$, a positive semidefinite matrix with ordered eigenvalues $0 \leq \lambda_1(\mathcal{L}) \leq \lambda_2(\mathcal{L}) \leq ... \leq \lambda_{|\mathcal{V}|}(\mathcal{L})$. Moreover, for an $n \times n$ matrix B, it has a series of order singular values $\sigma_1(B) \geq \sigma_2(B) \geq ... \geq \sigma_n(B) \geq 0$. Interested readers may refer to [37], [38] for more details on spectral graph theory.

A. Distributed Record Completion

The Distributed Record Completion, or DRC algorithm, allows each node k to obtain an accurate and complete global data vector $\hat{\boldsymbol{y}}_k$. DRC algorithm is an iteratively updating process as described below.

Algorithm 1 Framework of DRC for Each Terminal.

Start:

Initial non-random observation $\boldsymbol{y}_k(0) \in \mathbb{R}^M$, and let $\boldsymbol{u}_k(0) = \boldsymbol{y}_k(0)$.

Output:

Estimation on the global data, $\hat{\boldsymbol{y}}_k$.

1: **for** $i = 1$ to T **do**
2: Receiving neigbors' estimation on the global data, *i.e.,* $\{\boldsymbol{u}_v(i), v \in N_k\}$.
3: Comparing its own estimation with its incomplete observation, *i.e.,* $H_k\boldsymbol{u}_k(i) - \boldsymbol{y}_k(i)$;
4: Comparing its own estimation with its neighbors estimation, *i.e.,* $\sum_{v\in N_k}(\boldsymbol{u}_k(i) - \boldsymbol{u}_v(i))$;
5: Updating estimation $\boldsymbol{u}_k(i+1)$ based on Eq. (2);
6: **end for**
7: **return** $\hat{\boldsymbol{y}}_k = \boldsymbol{u}_k(T)$;

Alg. 1 summarizes the outline of DRC. To illustrate, the sequence $\{\boldsymbol{u}_k\}_{k\geq 0}$ is defined to represent the estimated records on all dimensions of data, generated by each node k as follows. Starting from the initial non-random observation $\boldsymbol{y}(0) \in \mathbb{R}^M$, at each iteration i, after observing an incomplete data $\boldsymbol{y}_k(i)$, each node k updates $\boldsymbol{u}_k(i)$ by a distributed iterative algorithm. In this algorithm, each node compares its estimated record $\boldsymbol{u}_k(i)$ with its neighbors', and also with the observation $\boldsymbol{y}_k(i)$. Then he determines the estimated record of the next time slot with the difference between $\boldsymbol{u}_k(i)$ and the deviations, as shown in what follows:

$$\begin{aligned}\boldsymbol{u}_k(i+1) = \boldsymbol{u}_k(i) &- \alpha(i)\sum_{v\in N_k}(\boldsymbol{u}_k(i) - \boldsymbol{u}_v(i)) \\ &- \beta(i)H_k^T(H_k\boldsymbol{u}_k(i) - \boldsymbol{y}_k(i)),\end{aligned} \tag{2}$$

where $\alpha(i)\sum_{v\in N_k}(\boldsymbol{u}_k(i) - \boldsymbol{u}_v(i))$ is the deviation of records from neighbors and $\beta(i)H_k^T(H_k\boldsymbol{u}_k(i) - \boldsymbol{y}_k(i))$ implies the deviation from observations. Since $\boldsymbol{u}_k$ is node k's estimation on all dimensions, for the comparison between the record and the observation, we use the linear mapping H_k. Both the positive weight sequence $\{\alpha(i)\}_{i\geq 0}$ and $\{\beta(i)\}_{i\geq 0}$ satisfy the persistence condition **C.5** given in Appendix A. For the ease of notation, we rewrite iterations in Equation (2) in a compact form, which can describe the consensus process of all nodes. To begin with, we store the incomplete observations of all nodes at iteration i in a long vector $\boldsymbol{y}(i) = [\boldsymbol{y}_1^T(i), ..., \boldsymbol{y}_{|\mathcal{V}|}^T(i)]^T$, store updates of the i-th iteration in another long vector $\boldsymbol{u}(i) = [\boldsymbol{u}_1^T(i), ..., \boldsymbol{u}_{|\mathcal{V}|}^T(i)]^T$, and define the following two matrices:

$$\bar{\mathcal{H}} = \text{diag}\left[H_1^T, ..., H_{|\mathcal{V}|}^T\right], \tag{3}$$

$$\tilde{\mathcal{H}} = \text{diag}\left[H_1^T H_1, ..., H_{|\mathcal{V}|}^T H_{\mathcal{V}}\right]. \tag{4}$$

Then, using the Kronecker product symbol, we can rewrite the Equation (2) in a compact form as

$$\begin{aligned}\boldsymbol{u}(i+1) = \boldsymbol{u}(i) &- \alpha(i)(\mathcal{L}\otimes E_M)\boldsymbol{u}(i) \\ &- \beta(i)\bar{\mathcal{H}}(\bar{\mathcal{H}}^T\boldsymbol{u}(i) - \boldsymbol{y}(i)).\end{aligned} \tag{5}$$

Given the total number of iteration steps T, each node k will obtain a data vector $\boldsymbol{u}_k(T)$ in the end. Let $\hat{\boldsymbol{y}}_k = \boldsymbol{u}_k(T)$, in Section IV-A, we will show $\hat{\boldsymbol{y}}_k$ is in fact an unbiased estimate on $\boldsymbol{y}$. As now we have the detailed updating process used in DRC algorithm, we will solve the distributed non-smooth minimization problem based on $\hat{\boldsymbol{y}}_k$.

B. Distributed Dual Average

Based on $\hat{\boldsymbol{y}}_k$, we now use Distributed Dual Average, or DDA algorithm to provide an accurate estimate $\hat{\boldsymbol{\theta}}_k^*$ on the optimal parameter $\boldsymbol{\theta}^*$ for each node in a distributed style. Formally, we need to solve the following minimization:

$$\min_{\boldsymbol{\theta}} \sum_{k=1}^{|\mathcal{V}|} f_k(\boldsymbol{\theta}, \hat{\boldsymbol{y}}_k), \text{ s.t., } \boldsymbol{\theta} \in \Theta, \hat{\boldsymbol{y}}_k \in \mathcal{Y}. \tag{6}$$

Note that f_k is a non-smooth function, where non-smoothness implies that the function does not have a continuous gradient, which makes solving such function more difficult than the smooth function. To deal with the non-smooth function, the sub-gradient method should be employed, while a slow convergence has to be endured [39]. For example, solving a single non-smooth optimization has an efficiency estimate of $\mathcal{O}(\frac{1}{\varepsilon^2})$ [40], where ε is the desired accuracy of the approximation solution, while minimizing a single smooth function only requires an efficiency estimate of the order $\mathcal{O}(\sqrt{\frac{1}{\varepsilon}})$ [31]. Furthermore, solving the distributed non-smooth optimization problem consisting of multiple interdependent non-smooth functions has even lower efficiency. Therefore, we propose the DDA algorithm to improve the efficiency of the distributed non-smooth optimization of Eq. (6) in crowdsensing networks.

Distributed Dual Averaging (DDA) Algorithm:

In Alg. 2, we summarizes the outline of DDA. It is designed by extending the centralized dual averaging method [20] into a distributed form. Now we provide the details of the algorithm.

The DDA algorithm requires each node to exchange information with its neighbors, and the exchange follows a weighting process, where the edge is assigned a weight. Thus, the process is strongly correlated with the network topology.

[3]Numerical natures of the graph can be investigated with the graph Laplacian matrix, for example, connectivity, expanding properties, diameter and *etc*. In this paper, we define the network connectivity using the Laplacian spectrum, which will be illustrated in the following assumption A.2.

Algorithm 2 Framework of DDA for Each Terminal.

Start:

Initial pair of vectors $(\boldsymbol{\theta}_k(0), \boldsymbol{\mu}_k(0)) \in \Theta \times \mathbb{R}^M$, and let $\boldsymbol{\mu}_k(0) = \hat{\boldsymbol{y}}_k$.

Output:

Estimation on the optimal parameter $\boldsymbol{\theta}^*$.

1: **for** $i = 1$ to T **do**
2: Computing the sub-gradient $\boldsymbol{g}_k(t) \in \nabla_\theta f_k(\boldsymbol{\theta}_k(t), \hat{\boldsymbol{y}}_k)$;
3: Receiving estimated information from neigobors, *i.e.,* $\{\boldsymbol{\mu}_j(t), j \in N_k\}$;
4: Updating $(\boldsymbol{\theta}_k(t), \boldsymbol{\mu}_k(t))$ with Eq. (7) and (8);
5: **end for**
6: **return** $\hat{\boldsymbol{\theta}}_k(T)$ with Eq. (9);

At each iteration t, each node k maintains a pair of vectors $(\boldsymbol{\theta}_k(t), \boldsymbol{\mu}_k(t)) \in \Theta \times \mathbb{R}^M$, computes its own sub-gradient $\boldsymbol{g}_k(t) \in \nabla_\theta f_k(\boldsymbol{\theta}_k(t), \hat{\boldsymbol{y}}_k)$, and receives information on sequences from its neighbor nodes, *i.e.,* $\{\boldsymbol{\mu}_j(t), j \in N_k\}$. Next, at each iteration, each node updates its maintained vector $(\boldsymbol{\theta}_k(t), \boldsymbol{\mu}_k(t))$ by weighting values from its neighbors. To model this weighting process, we use $P \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ to denote the edge weights matrix of the graph $\mathcal{G}$. Thus, $P_{kl} > 0$ if and only if $e_{kl} \in \mathcal{E}$ and $k \neq l$. This matrix represents the weight of each link, which can capture some natures of the link. For instance, the value can represent the intimacy between two nodes. A higher value implies the neighbor on this link will contribute more in the information exchange. Note that $P_{kl} > 0$ only if $(k, l) \in \mathcal{E}$, and $P_{kl} > 0$ only if $(k, l) \in \mathcal{E}$, the weight update is described with the following equations:

$$\boldsymbol{\mu}_k(t+1) = \sum_{l \in N_k} P_{kl}\boldsymbol{\mu}_l(t) + g_k(t), \tag{7}$$

$$\boldsymbol{\theta}_k(t+1) = \pi_{\omega(t+1)}(-\boldsymbol{\mu}_k(t+1)), \tag{8}$$

where the function $\pi_\omega(u)$ is defined by

$$\pi_\omega(\boldsymbol{\mu}) = \arg\min_{\boldsymbol{\xi}\in\Theta}\{-\langle \boldsymbol{\mu}, \boldsymbol{\xi}\rangle + \omega\phi(\boldsymbol{\xi})\},$$

and $\{\omega(t)\}$ is the non-decreasing sequence of positive step-sizes.

Specifically, we assume the matrix P is a doubly stochastic matrix, so

$$\sum_{l=1}^{|\mathcal{V}|} P_{kl} = \sum_{l\in N_k} P_{kl} = 1 \text{ for all } k \in \mathcal{V};$$

$$\sum_{k=1}^{|\mathcal{V}|} P_{kl} = \sum_{k\in N_l} P_{kl} = 1 \text{ for all } l \in \mathcal{V}.$$

To sum up, each node k computes its new dual sequence $\boldsymbol{\mu}_k(t+1)$ by weighting both its own sub-gradient $\boldsymbol{g}_k(t)$ and the sequences $\{\boldsymbol{\mu}_l(t), l \in N_k\}$ stored in its neighborhood N_k, and the node also computes its next local primal parameters $\boldsymbol{\theta}_k(t+1)$ by a projection defined by the proximal function ϕ and step-size $\omega(t) > 0$.

The intuition behind this method is: based on its current iteration $(\boldsymbol{\theta}_k(t), \boldsymbol{\mu}_k(t))$, each node k chooses its next iteration $\boldsymbol{\theta}_k(t+1)$ so as to minimize an averaged first-order approximation to the function $f = \sum_k f_k$, while the proximal function ϕ and step-size $\omega(t) > 0$ ensure that $\{\boldsymbol{\theta}_k(t)\}$ does not oscillate wildly during iterations.

At the end of iteration T, each node k has obtained a sequence $\{\boldsymbol{\theta}_k(t)\}_{1\leq t\leq T}$. We run a local average for each node as follows:

$$\hat{\boldsymbol{\theta}}_k(T) = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\theta}_k(t). \tag{9}$$

This means if we let $\hat{\boldsymbol{\theta}}_k^* = \hat{\boldsymbol{\theta}}_k(T)$ at the end of iteration T, we will have $\lim_{T\to\infty} f(\hat{\boldsymbol{\theta}}_k^*) = f(\boldsymbol{\theta}^*)$. Thus, with this iteration, each node k can obtain an estimate of the optimal parameter with any desired accuracy. We will prove the convergence of DDA in Section IV.

To sum up, in order to solve distributed non-smooth minimization problems in heterogeneous crowdsensing networks, we first present a DRC algorithm to allow each heterogeneous node to obtain an accurate estimate on the globally required data vector $\boldsymbol{y}$. Based on this, we design a DDA algorithm to ensure that each node obtains an accurate estimate on the optimal parameter $\boldsymbol{\theta}^*$. In the next section, we will present a formal analysis on the *convergence* and *convergent rates* of both algorithms.

IV. Main Properties of DRC and DDA

In this section, we present the main properties of the DRC and DDA algorithms. We defer detailed proofs in Section V.

A. Main Properties of DRC

To begin with, we present main properties with regard to the asymptotic unbiasedness and the consistency of the DRC algorithm. Furthermore, we carry out a convergence rate analysis by studying the deviation characteristic of DRC.

The results rely on the following three assumptions:

(A.1) Observation Noise: For $i \leq 0$, the noise $\left\{\boldsymbol{\varepsilon}(i) = [\varepsilon_1^T(i), ..., \varepsilon_{|\mathcal{V}|}^T(i)]^T\right\}_{i\geq 0}$ is *i.i.d.* zero mean. Moreover, at each node k, the noise sequence $\{\boldsymbol{\varepsilon}_k(i)\}_{1\leq k\leq|\mathcal{V}|, i\geq 0}$ is independent with each other, and the covariance of the observing noise, $S_{\boldsymbol{\varepsilon}}$, is independent over time i, *i.e.,*

$$\mathbb{E}[\varepsilon(i)\varepsilon(j)^T] = S_\varepsilon\delta_{ij}, \forall i, j \geq 0, \tag{10}$$

where $\delta_{ij} = 1$ *iff* $i = j$ or 0 otherwise.

(A.2) Networking Connectivity: The second eigenvalue of graph Laplacian $\mathcal{L}$ is non-negative, *i.e.,* $\lambda_2(\mathcal{L}) \geq 0$. We require the graph to be connected to allow communication among nodes. This can be guaranteed if $\lambda_2(\mathcal{L}) > 0$. See [37], [38] for details.

Before presenting the final assumption, we first give the following definition:

Definition 1. *The observations formulated by Equation (1) is* ***distributedly observable*** *if the matrix $\mathcal{H}$, defined by $\mathcal{H} = \sum_{k=1}^{|\mathcal{V}|} H_k^T H_k$, is of full rank.*

Remark: This distributed observability is essentially an extension of the observability condition for the centralized

observing system which is designed to obtain consistent and complete observation on the vector $\boldsymbol{y}$.

Now let us present the last assumption.

(A.3) Observability: The observations formulated by Equation (1) is *distributedly observable* defined by Definition 1.

1) Unbiasedness and consistency of DRC: In this part, we show the unbiasedness and the consistency of DRC algorithm, and we provide two theorems to illustrate them respectively.

Theorem 1. *Consider the DRC algorithm is under the assumptions* **A.1-A.3** *(Section III-A), the record sequence* $\{\boldsymbol{u}_k(i)\}_{i\geq 0}$ *at node* k *is asymptotic unbiased*

$$\lim_{i\to\infty}\mathbb{E}[\boldsymbol{u}_k(i)]=\boldsymbol{y}^*,\forall 1\leq k\leq|\mathcal{V}|. \tag{11}$$

We defer the proof in Section V-A. Theorem 1 shows the unbiasedness of the algorithm. It indicates that each node's estimation on the global data would be correct on the average in the long run. The consistency of DRC algorithm is guaranteed by the following theorem.

Theorem 2. *Consider the DRC algorithm is under the assumptions* **A.1-A.3** *(Section III-A),the records sequence* $\{\boldsymbol{u}_k(i)\}_{i\geq 0}$ *at node* k *is consistent*

$$\mathbb{P}\left[\lim_{i\to\infty}\boldsymbol{u}_k(i)=\boldsymbol{y}^*,\forall 1\leq k\leq|\mathcal{V}|\right]=1.$$

We provide the proof in Appendix B. Based on Theorem 2, record sequence $\{\boldsymbol{u}_k(i)\}_{i\geq 1}$ at every node, with probability 1, converges to the true vector $\boldsymbol{y}^*$.

2) Convergence rate analysis: We now analyze the convergence rate of the DRC algorithm via its deviation characteristic. We first present a relative definition which is used to characterized the convergence rate of sequential process.

Definition 2. *A sequence of records* $\{\boldsymbol{u}(i)\}_{i\geq 0}$ *is* ***asymptotically normal*** *if a positive semidefinite matrix* $S(\boldsymbol{y})$ *exists and satisfies that*

$$\lim_{i\to\infty}\sqrt{i}(\boldsymbol{u}_k(i)-\boldsymbol{y}^*)\to\mathcal{N}(\boldsymbol{0}_M,S_{kk}(\boldsymbol{y}(i))),\forall 1\leq k\leq n.$$

The matrix $S(\boldsymbol{y}(i))$ *is called the asymptotic variance of the observing sequence* $\{\boldsymbol{y}(i)\}_{i\geq 0}$*, and* $S_{kk}(\boldsymbol{y})\in\mathbb{R}^{M\times M}$ *denotes the* k*-th principal block of* $S(\boldsymbol{y}(i))$*.*

In the following part, we analyze the asymptotic normality of the DRC algorithm. Let $\lambda_{\min}(\gamma\mathcal{L}\otimes E_M+\tilde{\mathcal{H}})$ denote the smallest eigenvalue of $[\gamma\mathcal{L}\otimes E_M+\tilde{\mathcal{H}}]$. Recalling the noise covariance S_ϵ in (10), we present the following theorem to establish the asymptotic normality of the DRC algorithm.

Theorem 3. *Consider the DRC algorithm is under the assumptions* **A.1-A.3** *(Section III-A), with weight sequence* $\{\alpha(i)\}_{i\geq 0}$ *and* $\{\beta(i)\}_{i\geq 0}$ *that are given by*

$$\alpha(i)=\frac{a}{i+1},\lim_{i\to\infty}\frac{\alpha(i)}{\beta(i)}=\gamma>0,$$

for some $a>0$*. Let the record sequence* $\{\boldsymbol{u}(i)\}_{i\geq 0}$ *be the state sequence generated by (5). Then, for* $a>\frac{1}{2\lambda_{\min}(\gamma\mathcal{L}\otimes E_M+\tilde{\mathcal{H}})}$*, we obtain*

$$\sqrt{(i)}(\boldsymbol{u}(i)-\boldsymbol{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*)\Longrightarrow\mathcal{N}(\boldsymbol{0},S(\boldsymbol{y}(i))),$$

where

$$S(\boldsymbol{y}(i)) = a^2\int_0^\infty e^{\Sigma v}S_0e^{\Sigma v}dv, \tag{12}$$

$$\Sigma = -a[\gamma\mathcal{L}\otimes E_M+\tilde{\mathcal{H}}]+\frac{1}{2}E_{M|\mathcal{V}|}, \tag{13}$$

and

$$S_0 = \bar{\mathcal{H}}S_\epsilon\bar{\mathcal{H}}^T. \tag{14}$$

Especially, the record sequence $\{\boldsymbol{u}_k(i)\}_{i\geq 0}$ *at any node* k *is asymptotically normal:*

$$\sqrt{(i)}(\boldsymbol{u}_k(i)-\boldsymbol{y}^*)\Longrightarrow\mathcal{N}(\boldsymbol{0},S_{kk}(\boldsymbol{y}(i))).$$

We provide the proof in Appendix B. Therefore, the error sequence $\{\boldsymbol{u}_k(i)-\boldsymbol{y}^*\}_{i\geq 0}$ at each node can be regarded as being convergent to a normal distribution with a rate of $\frac{1}{\sqrt{i}}$.

Up until now, we have presented asymptotic unbiasedness, the consistence and the asymptotic normality of the DRC algorithm. In the next section, we present main properties of the DDA algorithm.

B. Main Properties of DDA

In this section, we prove the convergency of running average $\hat{\boldsymbol{\theta}}_k(T)$ to the optimal parameter $\boldsymbol{\theta}^*$ and derive the convergence rate of the DDA algorithm.

Now we present the following theorem.

Theorem 4. *The random family* $\{\boldsymbol{\theta}_k(t)\}_{t=0}^\infty$ *and* $\{\boldsymbol{\mu}_k(t)\}_{t=0}^\infty$ *are generated by iteration (8) and (7), with the positive non-decreasing step-size sequence* $\{\omega(t)\}_{t=0}^\infty$*, where* ϕ *is strongly convex with respect to the norm* $\|\cdot\|$ *with dual norm* $\|\cdot\|_*$*. Let the record error* $\left\|\mathbb{E}[\hat{\boldsymbol{y}}_k]-\boldsymbol{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\right\|$ *be bounded by an arbitrary small constant* C_{err}*. For any* $\boldsymbol{\theta}^*\in\Theta$ *and each node* $k\in\mathcal{V}$*, we have*

$$f(\hat{\boldsymbol{\theta}}_k(T),\hat{\boldsymbol{y}}_k)-f(\boldsymbol{\theta}^*,\boldsymbol{y}^*)\leq OPT+NET+SAMP,$$

where

$$OPT=\frac{\omega(T)}{T}\phi(\boldsymbol{\theta}^*)+\frac{L^2}{2T\tau}\sum_{t=1}^T\frac{1}{\omega(t)},$$

$$NET=\frac{L}{T}\sum_{t=1}^T\frac{1}{\omega(t)}\mathbb{E}\Bigg[\frac{2}{|\mathcal{V}|}\sum_{j=1}^{|\mathcal{V}|}\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_j(t)\right\|_*+\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_k(t)\right\|\Bigg],$$

$$SAMP=LC_{err},$$

$$\bar{\boldsymbol{\mu}}(t)=\frac{1}{|\mathcal{V}|}\sum_{k=1}^{|\mathcal{V}|}\boldsymbol{\mu}_k(t).$$

Recall that τ *is the convexity parameter.*

Theorem 4 explicitly shows the difference between the estimated results from the true optimality. It is bounded by a value which is a sum of three types of errors: (1) The OPT error can be viewed as the optimization error; (2) the NET error is induced by various estimations of nodes; and (3) the SAMP error is incurred on account of the input noisy. The

theorem indicates the relationship between the difference and T, which will help us understand the convergency of the DDA algorithm. The detailed proof will be given in Section V.

We next investigate the relationship between the convergence rates and the spectral property of the network. For a given graph $\mathcal{G}$, we assume that communications between nodes are controlled by a double stochastic matrix P. In the following, we show that the spectral gap of the network, *i.e.,* $\gamma(P) = 1 - \sigma_2(P)$ of P severely influences the convergence rate of DDA, where $\sigma_2(P)$ is the second largest singular value of P.

Theorem 5. *Following Theorem 4 and recalling that $\phi(\boldsymbol{\theta}^*) \leq A^2$, if we define the step-size $\omega(t)$ and the record error $\left\|\mathbb{E}[\hat{\boldsymbol{y}}_k] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*\right\|$ as:*

$$\omega(t) = A\sqrt{t} \text{ and } C_{err} = \frac{2L}{A\sqrt{T}} \cdot \frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)},$$

we will have

$$f(\hat{\boldsymbol{\theta}}_k(T), \boldsymbol{y}) - f(\boldsymbol{\theta}, \boldsymbol{y}^*) \leq \frac{16L^2}{A\sqrt{T}} \frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)}, \text{ for all } k \in \mathcal{V}.$$

We defer the proof in Section V-C. Theorem 5 shows that the convergence rate of distributed subgradient methods heavily relies on the graph spectral property. The dependence on the spectral quantity $1-\sigma_2(P)$ is quite natural, since lots of work have noticed that the propagation of information severely relies on the spectral property of the underlying network.

As we have presented all main properties of our algorithms, we will next turn to the detailed proof of each theorem.

V. Proof of Theorems

A. Proof of Theorem 1

Proof: Taking the expectation of both sides of Eq. (5), it follows

$$\begin{aligned}\mathbb{E}[\boldsymbol{u}(i+1)] = \mathbb{E}[\boldsymbol{u}(i)] - \alpha(i)(\mathcal{L} \otimes E_M)\mathbb{E}[\boldsymbol{u}(i)] \\ + \beta(i)\tilde{\mathcal{H}}\mathbb{E}[\boldsymbol{y}(i)] - \beta(i)\tilde{\mathcal{H}}\bar{\mathcal{H}}^T\mathbb{E}[\boldsymbol{u}(i)].\end{aligned} \tag{15}$$

Given that

$$(\mathcal{L} \otimes E_M)(\mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*) = \mathbf{0}_{|\mathcal{V}|M}, \tag{16}$$

$$\tilde{\mathcal{H}}(\mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*) = \bar{\mathcal{H}}\mathbb{E}[\boldsymbol{y}(i)], \tag{17}$$

subtracting both sides of Eq. (15) by $\mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*$, we have

$$\begin{aligned}\mathbb{E}[\boldsymbol{u}(i+1)] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^* = [E_{|\mathcal{V}|M} - \alpha(i)\mathcal{L} \otimes E_M \\ - \beta(i)\tilde{\mathcal{H}}][\mathbb{E}[\boldsymbol{u}(i)] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*].\end{aligned} \tag{18}$$

Continuing the iteration in (18), we have, for each $i \geq i_0 = \max\{i_1, i_2\}$,

$$\begin{aligned}\left\|\mathbb{E}[\boldsymbol{u}(i)] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*\right\| \leq \left(\prod_{j=i_0}^{i-1} \left\| E_{M|\mathcal{V}|} - \alpha(j)\mathcal{L} \otimes E_M \right.\right. \\ \left.\left. -\beta(j)\tilde{\mathcal{H}}\right\|\right) \times \left\|\mathbb{E}[\boldsymbol{x}(i_0)] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*\right\|.\end{aligned} \tag{19}$$

To further derive the above formulation, we have the following facts.

First, since $\frac{\alpha(i)}{\beta(i)} \to \gamma$, we have

$$\exists i_1 \ni: \frac{\gamma}{2} \leq \frac{\alpha(i)}{\beta(i)} \leq 2\gamma, \forall i \geq i_1. \tag{20}$$

Second, let $\lambda_{\min}(\gamma\mathcal{L} \otimes E_M + \tilde{\mathcal{H}})$ be the smallest eigenvalue of the positive definite matrix[4] $[\gamma\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}]$. Since $\alpha(i) \to 0$, we have

$$\exists i_2 \ni: \alpha(i) \leq \frac{1}{\lambda_{\min}(\gamma\mathcal{L} \otimes E_M + \tilde{\mathcal{H}})}. \forall i \geq i_2 \tag{21}$$

Third, the other facts include: 1) $\lambda_{\min}(A+B) \geq \lambda_{\min}(A) + \lambda_{\min}(B)$ (Courant-Fischer Minimax Theorem [41]), 2) $\lambda_{\min}(\mathcal{L} \otimes E_M) = \lambda_{\min}(\mathcal{L}) \geq 0$.

Based on above facts, the multiplicand of Equation (19) follows from (21), for each $j \geq i_0$

$$\begin{aligned}&||E_{M|\mathcal{V}|} - \alpha(j)\mathcal{L} \otimes E_M - \beta(j)\tilde{\mathcal{H}}|| \\ &= \left\| E_{M|\mathcal{V}|} - \beta(j)(\frac{\alpha(j)}{\beta(j)}\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}) \right\| \\ &= 1 - \beta(j)\lambda_{\min}(\frac{\alpha(j)}{\beta(j)}\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}) \\ &= 1 - \beta(j)\lambda_{\min}((\frac{\alpha(j)}{\beta(j)} - \frac{\gamma}{2})\mathcal{L} \otimes E_M + \frac{\gamma}{2}\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}) \\ &\leq 1 - \beta(j)\lambda_{\min}(\frac{\gamma}{2}\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}).\end{aligned} \tag{22}$$

From (19) and (22), we now have for each $i > i_0$,

$$\begin{aligned}\left\|\mathbb{E}[\boldsymbol{u}(i)] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}\right\| \leq \left(\prod_{j=i_0}^{i-1} (1 - \beta(j)\lambda_{\min}(\frac{\gamma}{2}\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}))\right) \\ \times \left\|\mathbb{E}[\boldsymbol{u}(i_0)] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*]\right\|.\end{aligned} \tag{23}$$

Finally, from the inequality $1 - a \leq e^{-a}, 0 \leq a \leq 1$, we get

$$\begin{aligned}\left\|\mathbb{E}[\boldsymbol{u}(i)] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*\right\| \leq \exp\left[-\lambda_{\min}(\frac{\gamma}{2}\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}) \sum_{j=i_0}^{i-1} \beta(j)\right] \\ \times \left\|\mathbb{E}[\boldsymbol{u}(i_0)] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*]\right\|, i > i_0.\end{aligned} \tag{24}$$

With the facts that $\lambda_{\min}(\gamma\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}) > 0$ and the sum of $\beta(j)$ approaches to infinity, we have

$$\lim_{i \to \infty} \left\|\mathbb{E}[\boldsymbol{u}(i)] - \mathbf{1}_{|\mathcal{V}|} \otimes \boldsymbol{y}^*\right\| = 0.$$

Thus we complete the proof. ■

B. Proof of Theorem 4

Before proving the theorem of algorithm convergency, we present here some basic assumptions and necessary lemmas.

(A.4) A *prox-function* $\phi : \Theta \to \mathbb{R}$ exists to be τ-strongly convex with respect to the norm $\|\cdot\|$, *i.e.,*

$$\phi(\boldsymbol{\theta}_1) \geq \phi(\boldsymbol{\theta}_2) + \langle \nabla_\theta \phi(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 \rangle + \frac{\tau}{2} \left\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\right\|^2,$$

for $\theta_1, \theta_2 \in \Theta$. Function ϕ is non-negative over Θ and $\phi(\mathbf{0}) = 0$. The *prox-center* of Θ is given by $\theta_0 = \arg\min_\theta\{\phi(\theta) : \theta \in$

[4]Due to the page limit, we omit the proof of the positive semidefiniteness of the matrix.

$\Theta\}$. Moreover, we assume that for the optimal parameter $\boldsymbol{\theta}^*$, $\phi(\boldsymbol{\theta}^*) \leq A^2$.

(A.5) The error function f_k at each node k is *L-Lipschitz* with respect to the norm $\|\cdot\|$, *i.e.,* for $\theta_1, \theta_2 \in \Theta$, we have

$$|f_k(\boldsymbol{\theta}_1, \hat{\boldsymbol{y}}_k) - f_k(\boldsymbol{\theta}_2, \hat{\boldsymbol{y}}_k)| \leq L \|\theta_1 - \theta_2\|.$$

Lemma 1. *Define the function*

$$V_\omega(\boldsymbol{\theta}) = \max_{\boldsymbol{\zeta} \in \Theta} \{\langle \boldsymbol{\theta}, \boldsymbol{\zeta} - \boldsymbol{\theta}_0 \rangle - \omega\phi(\boldsymbol{\zeta})\}.$$

Then function $V_\omega(\cdot)$ is convex and differentiable on Θ. Moreover, its gradient is L-Lipschitz continuous with respect to the norm $\|\cdot\|$

$$\|\nabla V_\omega(u) - \nabla V_\omega(v)\| \leq \frac{1}{\omega\tau} \|u - v\|, \forall u, v \in \Theta,$$

where the gradient is defined as follows

$$\nabla V_\omega(u) = \pi_\omega(u) - u_0, \pi_\omega(u) = \arg\min_{v \in \Theta} \{-\langle u, v \rangle + \omega\phi(v)\}. \tag{25}$$

Note that $u_0 = \pi_\omega(0)$.

Lemma 2. *Let $\{\boldsymbol{g}(t)\}_{t=1}^\infty$ be an arbitrary sequence of vectors, and consider the sequence $\{\boldsymbol{\theta}(t)\}_{t=1}^\infty$ generated by*

$$\begin{aligned}\boldsymbol{\theta}(t+1) &= \arg\min_{\boldsymbol{\theta} \in \Theta} \left\{ \sum_{r=1}^t \langle \boldsymbol{g}(r), \boldsymbol{\theta} \rangle + \omega(t)\phi(\boldsymbol{\theta}) \right\} \\ &= \pi_{\omega(t)} \left(-\sum_{r=1}^t \boldsymbol{g}(r) \right).\end{aligned}$$

For any non-decreasing positive step-sizes $\{\omega(t)\}_{t=0}^\infty$, and any $\hat{\boldsymbol{\theta}} \in \Theta_C$, we have

$$\sum_{t=1}^T \left\langle \boldsymbol{g}(t), \boldsymbol{\theta}(t) - \hat{\boldsymbol{\theta}} \right\rangle \leq \frac{1}{2\tau} \sum_{t=1}^T \frac{\|\boldsymbol{g}(t)\|_*^2}{\omega(t)} + \omega(T)C.$$

For any $\hat{\boldsymbol{\theta}} \in \Theta_C \subset \Theta^*$, we have

$$\sum_{t=1}^T \left\langle \boldsymbol{g}(t), \boldsymbol{\theta}(t) - \hat{\boldsymbol{\theta}} \right\rangle \leq \frac{1}{2\tau} \sum_{t=1}^T \frac{\|\boldsymbol{g}(t)\|_*^2}{\omega(t)} + \omega(T)\phi(\boldsymbol{\theta}^*).$$

In addition, we establish the convergency of algorithm via two auxiliary sequences

$$\boldsymbol{\varphi}(t+1) = \pi_{\omega(t)}(\bar{\boldsymbol{\mu}}(t+1)) \tag{26}$$

and present the following lemma.

Lemma 3. *With definitions of the random family $\{\boldsymbol{\theta}_k(t)\}_{t=0}^\infty$, $\{\boldsymbol{\mu}_k(t)\}_{t=0}^\infty$ and $\{\boldsymbol{\varphi}_k(t)\}_{t=1}^\infty$ in Eq. (7), (8) and (26), and the L-Lipschitz condition of each f_k, for each node $k \in \mathcal{V}$, we have*

$$\begin{aligned}\sum_{t=1}^T [f(\boldsymbol{\theta}_k(t), \boldsymbol{y}_k) - f(\boldsymbol{\theta}^*, \boldsymbol{y}^*)] &\leq \sum_{t=1}^T [f(\boldsymbol{\varphi}(t), \boldsymbol{y}_k) - f(\boldsymbol{\theta}^*, \boldsymbol{y}_k)] \\ &+ \sum_{t=1}^T [L \|\boldsymbol{\theta}_k(t) - \boldsymbol{\varphi}(t)\| + L \|\boldsymbol{y}_k - \boldsymbol{y}^*\|].\end{aligned}$$

Similarly, defining $\hat{\boldsymbol{\varphi}}(T) = \frac{1}{T}\sum_{t=1}^T \boldsymbol{\varphi}(t)$ and $\hat{\boldsymbol{\theta}}_k(T) = \frac{1}{T}\sum_{t=1}^T \boldsymbol{\theta}_k(t)$, we have

$$\begin{aligned}f(\hat{\boldsymbol{\theta}}_k(t), \boldsymbol{y}_k) - f(\boldsymbol{\theta}^*, \boldsymbol{x}^*) &\leq f(\hat{\boldsymbol{\varphi}}(t), \boldsymbol{y}_k) - f(\boldsymbol{\theta}^*, \boldsymbol{y}_k) \\ &+ \frac{L}{\omega(t)T} \sum_{t=1}^T \|\boldsymbol{\theta}_k(t) - \boldsymbol{\varphi}(t)\| + L \|\boldsymbol{y}_k - \boldsymbol{y}^*\|.\end{aligned}$$

Based on above lemmas, we now present the proof of Theorem 4.

Proof: We perform our proof by analyzing the random family $\{\boldsymbol{\varphi}(t)\}_{t=0}^\infty$. Given an arbitrary $\boldsymbol{\theta}^* \in \Theta$, we have

$$\begin{aligned}&\sum_{t=1}^T [f(\boldsymbol{\varphi}(t), \hat{\boldsymbol{y}}_k) - f(\boldsymbol{\theta}^*, \hat{\boldsymbol{y}}_k)] \\ &= \frac{1}{|\mathcal{V}|} \sum_{t=1}^T \sum_{k=1}^{|\mathcal{V}|} [f_k(\boldsymbol{\varphi}(t), \hat{\boldsymbol{y}}_k) - f_k(\boldsymbol{\theta}^*, \hat{\boldsymbol{y}}_k)] \\ &\leq \sum_{t=1}^T \frac{1}{|\mathcal{V}|} \left[\sum_{k=1}^{|\mathcal{V}|} [L \|\boldsymbol{\varphi}(t) - \boldsymbol{\theta}_k(t)\| + f_k(\boldsymbol{\varphi}(t)) - f_k(\boldsymbol{\theta}_k(t))] \right].\end{aligned}$$

The inequality of the above equation is resulted by the L-Lipschitz condition on f_k.

Let $\boldsymbol{g}_k(t) \in \partial f_k(\boldsymbol{\theta}_k(t))$ and use the convexity of the function, then we will obtain the following bound:

$$\begin{aligned}&\sum_{k=1}^{|\mathcal{V}|} [f_k(\boldsymbol{\theta}_k(t)) - f_k(\boldsymbol{\theta}^*)] \leq \sum_{k=1}^{|\mathcal{V}|} \langle \boldsymbol{g}_k(t), \boldsymbol{\theta}_k(t) - \boldsymbol{\theta}^* \rangle \\ &= \sum_{k=1}^{|\mathcal{V}|} \langle \hat{\boldsymbol{g}}_k(t), \boldsymbol{\varphi}(t) - \boldsymbol{\theta}^* \rangle + \sum_{k=1}^{|\mathcal{V}|} \langle \hat{\boldsymbol{g}}_k(t), \boldsymbol{\theta}_k(t) - \boldsymbol{\varphi}(t) \rangle \\ &\quad + \sum_{k=1}^{|\mathcal{V}|} \langle \boldsymbol{g}_k(t) - \hat{\boldsymbol{g}}_k(t), \boldsymbol{\theta}_k(t) - \boldsymbol{\theta}^* \rangle\end{aligned} \tag{27}$$

For the first term in the right hand side of Equation (27), from the Lemma 2, it follows that

$$\begin{aligned}&\frac{1}{|\mathcal{V}|} \sum_{t=1}^T \left\langle \sum_{k=1}^{|\mathcal{V}|} \hat{\boldsymbol{g}}_k(t), \boldsymbol{\varphi}(t) - \boldsymbol{\theta}^* \right\rangle \\ &\leq \frac{1}{2\tau} \sum_{t=1}^T \frac{1}{\omega(t)} \left\| \frac{1}{|\mathcal{V}|} \sum_{k=1}^{|\mathcal{V}|} \hat{\boldsymbol{g}}_k(t) \right\|_*^2 + \omega(T)\phi(\boldsymbol{\theta}^*).\end{aligned} \tag{28}$$

Holder's inequality implies that $\mathbb{E}[\|\hat{\boldsymbol{g}}_l(t)\|_* \|\hat{\boldsymbol{g}}_k(s)\|_*] \leq L^2$ and $\mathbb{E}[\|\hat{\boldsymbol{g}}_k(t)\|_*] \leq L^2$ since $\|\hat{\boldsymbol{g}}_k(t)\|_* \leq L$ for any k, l, s, t. We use these two inequalities to bound (28),

$$\mathbb{E} \left\| \frac{1}{|\mathcal{V}|} \sum_{k=1}^{|\mathcal{V}|} \hat{\boldsymbol{g}}_k(t) \right\|_*^2 \leq \frac{1}{|\mathcal{V}|^2} \sum_{k,l=1}^{|\mathcal{V}|} \mathbb{E}[\|\hat{\boldsymbol{g}}_k(t)\|_* \|\hat{\boldsymbol{g}}_l(t)\|_*] \leq L^2.$$

For the second term in the right hand side of Equation (27), $\boldsymbol{\theta}_k \in \mathcal{F}_{t-1}$ and $\boldsymbol{\varphi}(t) \in \mathcal{F}_{t-1}$ by assumption, so

$$\begin{aligned}\mathbb{E} \langle \hat{\boldsymbol{g}}_k(t), \boldsymbol{\theta}_k(t) - \boldsymbol{\varphi}(t) \rangle &\leq \mathbb{E} \|\hat{\boldsymbol{g}}_k(t)\| \|\boldsymbol{\theta}_k(t) - \boldsymbol{\varphi}(t)\| \\ &= \mathbb{E}(\mathbb{E}[\|\hat{\boldsymbol{g}}_k(t)\| \,|\mathcal{F}_{t-1}] \|\boldsymbol{\theta}_k(t) - \boldsymbol{\varphi}(t)\|) \\ &\leq L\mathbb{E} \|\boldsymbol{\theta}_k(t) - \boldsymbol{\varphi}(t)\| \\ &\leq \frac{L}{\omega(t)\tau} \mathbb{E} \|\bar{\boldsymbol{\mu}}(t) - \boldsymbol{\mu}_k(t)\|_*.\end{aligned} \tag{29}$$

Thus we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{k=1}^{|\mathcal{V}|}\|\boldsymbol{\varphi}(t)-\boldsymbol{\theta}_k(t)\|+\sum_{t=1}^{T}\sum_{k=1}^{|\mathcal{V}|}\langle\hat{\boldsymbol{g}}_k(t),\boldsymbol{\theta}_k(t)-\boldsymbol{\varphi}(t)\rangle\right]$$
$$\leq 2L\sum_{t=1}^{T}\sum_{k=1}^{|\mathcal{V}|}\frac{\mathbb{E}\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_k(t)\right\|_*}{\omega(t)}.$$

For the third term in the right hand side of Equation (27), recalling that $\boldsymbol{\theta}_k(t)\in\mathcal{F}_{t-1}$, we get

$$\begin{aligned}&\mathbb{E}[\langle\boldsymbol{g}_k(t)-\hat{\boldsymbol{g}}_k(t),\boldsymbol{\theta}_k(t)-\boldsymbol{\theta}^*\rangle]\\&=\mathbb{E}[\langle\mathbb{E}(\boldsymbol{g}_k(t))-\hat{\boldsymbol{g}}_k(t)|\mathcal{F}_{t-1},\boldsymbol{\theta}_k(t)-\boldsymbol{\theta}^*\rangle]=0.\end{aligned}\tag{30}$$

Combining these equations, we obtain the running sum bound

$$\sum_{t=1}^{T}[f(\boldsymbol{\varphi}(t))-f(\boldsymbol{\theta}^*)]\leq\omega(T)\phi(\boldsymbol{\theta}^*)+\frac{L^2}{2\tau}\sum_{t=1}^{T}\frac{1}{\omega(t)}+\frac{2L}{|\mathcal{V}|}\sum_{t=1}^{T}\sum_{k=1}^{|\mathcal{V}|}\frac{\mathbb{E}\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_k(t)\right\|_*}{\omega(t)\tau}.\tag{31}$$

Applying Lemma 4 to (31), it gives that

$$\begin{aligned}\sum_{t=1}^{T}[f(\boldsymbol{\theta}_k(t),\hat{\boldsymbol{y}}_k)-f(\boldsymbol{\theta}^*,\boldsymbol{y}^*)]&\leq\omega(T)\phi(\boldsymbol{\theta}^*)+\frac{L^2}{2\tau}\sum_{t=1}^{T}\frac{1}{\omega(t)}\\&+\frac{2L}{|\mathcal{V}|}\sum_{t=1}^{T}\sum_{k=1}^{|\mathcal{V}|}\frac{\mathbb{E}\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_k(t)\right\|_*}{\omega(t)\tau}+L\sum_{t=1}^{T}\|\hat{\boldsymbol{y}}_k-\boldsymbol{y}^*\|\\&+\sum_{t=1}^{T}\frac{\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_k(t)\|}{\omega(t)\tau}.\end{aligned}$$

Dividing both sides of the inequality by T, we can obtain the theorem based on convexity of f. ∎

C. Proof of Theorem 5

Proof: The proof concentrates on deriving the bound of the network error in Theorem 4, $\sum_{j=1}^{|\mathcal{V}|}\frac{\mathbb{E}\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_j(t)\right\|_*}{\omega(t)}$. We first define the matrix $\mathcal{P}(t,l)=P^{t-l+1}$, where $[\mathcal{P}(t,l)]_{ij}$ is the element in the i-th row and j-th column of the matrix $\mathcal{P}(t,l)$. From Equation (7), based on the record at time l, i.e. $\boldsymbol{\mu}_j(l)$, we can obtain the record at time $t+1$ as follows:

$$\begin{aligned}\boldsymbol{\mu}_j(t+1)&=\sum_{i=1}^{|\mathcal{V}|}[\mathcal{P}(t,l)]_{ij}\boldsymbol{\mu}_j(l)\\&+\sum_{k=l+1}^{t}\sum_{i=1}^{|\mathcal{V}|}[\mathcal{P}(t,k)]_{ij}\hat{\boldsymbol{g}}_i(k-1)+\hat{\boldsymbol{g}}_j(t).\end{aligned}\tag{32}$$

If $t=l$, this iteration will be terminated in our algorithm. From the definition of $\bar{\boldsymbol{\mu}}(t)$ in Lemma 2, it follows:

$$\begin{aligned}\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_j(t)&=\sum_{k=1}^{t-1}\sum_{i=1}^{|\mathcal{V}|}\left(\frac{1}{|\mathcal{V}|}-[\mathcal{P}(t-1,k)]_{ij}\right)\hat{\boldsymbol{g}}_i(k-1)\\&+\frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}[\hat{\boldsymbol{g}}_i(t-1)-\hat{\boldsymbol{g}}_j(t-1)],\end{aligned}\tag{33}$$

which further implies that

$$\begin{aligned}\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_j(t)\right\|_*&\leq\sum_{k=1}^{t-1}\sum_{i=1}^{|\mathcal{V}|}\left|\frac{1}{|\mathcal{V}|}-[\mathcal{P}(t-1,k)]_{ij}\right|\|\hat{\boldsymbol{g}}_i(k-1)\|_*\\&+\frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}[\|\hat{\boldsymbol{g}}_i(t-1)\|_*+\left\|\hat{\boldsymbol{g}}_j(t-1)\right\|_*].\end{aligned}\tag{34}$$

Taking the expectation on both sides of Inequality (34) and using the fact that $\mathbb{E}\left\|\hat{\boldsymbol{g}}_i(t)\right\|_*\leq L$, we have

$$\mathbb{E}\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_j(t)\right\|_*\leq\sum_{k=1}^{t-1}L\left\|\frac{\mathbf{1}_{|\mathcal{V}|}}{|\mathcal{V}|}-\mathcal{P}(t-1,k)e_j\right\|_1+2L,\tag{35}$$

where e_j represents the j-th standard basis vector in the $|\mathcal{V}|$-dimensional Euclidean space. To further bound $\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_j(t)\right\|_*$, we break the sum in Inequality (35) into two terms, with a cutoff point $\tilde{t}$. From the Perron-Frobenius theory presented in [42], we have

$$\left\|\mathcal{P}(t,k)e_j-\frac{\mathbf{1}_{|\mathcal{V}|}}{|\mathcal{V}|}\right\|_1\leq\sqrt{|\mathcal{V}|}\sigma_2^{t-k+1}(P).$$

From the above inequality, it follows that if

$$1\leq k\leq t-\frac{\ln C_{err}}{\ln\sigma_2(P)}+1,\text{ then }\left\|\mathcal{P}(t,k)e_j-\frac{\mathbf{1}_{|\mathcal{V}|}}{|\mathcal{V}|}\right\|_1\leq\sqrt{|\mathcal{V}|}C_{err}.$$

Specifically, setting $C_{err}=1/T\sqrt{|\mathcal{V}|}$, for $\forall l:1\leq k\leq t-\frac{\ln C_{err}}{\ln\sigma_2(P)}+1$, we have

$$\left\|\mathcal{P}(t,k)e_j-\frac{\mathbf{1}_{|\mathcal{V}|}}{|\mathcal{V}|}\right\|_1\leq\frac{1}{T}.$$

For $k>t-\frac{\ln C_{err}}{\ln\sigma_2(P)}+1$, we have

$$\left\|\mathcal{P}(t,k)e_j-\frac{\mathbf{1}_{|\mathcal{V}|}}{|\mathcal{V}|}\right\|_1\leq\|\mathcal{P}(t,k)e_j\|_1+\frac{1}{|\mathcal{V}|}\left\|\mathbf{1}_{|\mathcal{V}|}\right\|_1=2.\tag{36}$$

The above clearly suggests that the cutoff point is $\tilde{t}=\frac{\ln C_{err}}{\ln\sigma_2(P)}$. Since there are at most T steps in the summation, we have

$$\begin{aligned}\mathbb{E}\left\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_j(t)\right\|_*&\leq L\sum_{k=t-\tilde{t}}^{t-1}\left\|\mathcal{P}(t-1,k)e_j-\frac{\mathbf{1}_{|\mathcal{V}|}}{|\mathcal{V}|}\right\|_1\\&+L\sum_{k=1}^{t-\tilde{t}-1}\left\|\mathcal{P}(t-1,k)e_j-\frac{\mathbf{1}_{|\mathcal{V}|}}{|\mathcal{V}|}\right\|_1+2L\\&\leq 2L\tilde{t}+\frac{L}{T}(t-\tilde{t}-1)+2L\\&\leq 2L\tilde{t}+L+2L\ (\text{by } t\leq T)\\&=2L\frac{\ln(T\sqrt{|\mathcal{V}|})^{-1}}{\ln(\sigma_2(P))}+3L\\&=2L\frac{\ln(T\sqrt{|\mathcal{V}|})}{\ln\sigma_2^{-1}(P)}+3L\\&\leq 2L\frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)}+3L,\end{aligned}$$

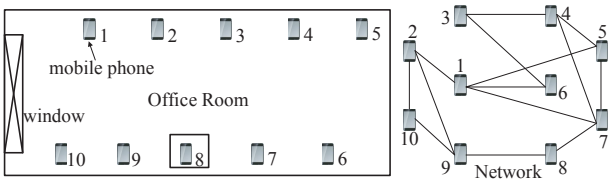


Fig. 1. The real world test and the communication network

which follows the upper bound of the network error in (4)

$$NET \leq \frac{3L}{T}\left(2L\frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)}+3L\right)\sum_{t=1}^{T}\frac{1}{\omega(t)}.$$

Therefore, the learning error $f(\hat{\theta}_i(T),\hat{\boldsymbol{y}}_i)-f(\theta,\boldsymbol{y}^*)$ at the i-th node can be further bounded by

$$f(\hat{\boldsymbol{\theta}}_i(T),\hat{\boldsymbol{y}}_i)-f(\boldsymbol{\theta}^*,\boldsymbol{y}^*)\leq\frac{\omega(T)}{T}\phi(\boldsymbol{\theta}^*)+LC_{err}+\left(\frac{6L^2}{T}\frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)}+\frac{9L^2}{T}+\frac{L^2}{2T\tau}\right)\sum_{t=1}^{T}\frac{1}{\omega(t)}.$$

If we choose the weight sequence $\{\omega(t)\}_{t=1}^{T}$ and arbitrary small error C_{err} to be

$$\omega(t)=A\sqrt{t},\quad C_{err}=\frac{2L}{A\sqrt{T}}\cdot\frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)},$$

then we have

$$\begin{aligned}f(\hat{\boldsymbol{\theta}}_i(T),\hat{\boldsymbol{y}}_i)-f(\boldsymbol{\theta}^*,\boldsymbol{y}^*)&\leq\frac{A\phi(\boldsymbol{\theta}^*)}{\sqrt{T}}+\frac{2L^2}{A\sqrt{T}}\frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)}\\&\quad+\frac{2}{A}\left(\frac{6L^2}{\sqrt{T}}\frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)}+\frac{9L^2}{\sqrt{T}}+\frac{L^2}{2\sqrt{T}\tau}\right)\\&\leq\frac{16L^2}{A\sqrt{T}}\frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)}.\end{aligned}$$

Therefore, we obtain

$$f(\hat{\boldsymbol{\theta}}_i(T),\hat{\boldsymbol{y}}_i)-f(\boldsymbol{\theta}^*,\boldsymbol{y}^*)=\mathcal{O}\left(\frac{1}{\sqrt{T}}\frac{\ln(T\sqrt{|\mathcal{V}|})}{1-\sigma_2(P)}\right)$$

and thereby completing the proof. ■

VI. Performance Evaluation

In this section, we present our testing results on the convergence feature of both DRC and DDA algorithms. Besides, we compare the DDA with other methods in [33], [43] and show the efficiency of our algorithms.

We perform simulations on the network of $|\mathcal{V}|=K$ nodes with two different graph topologies, *e.g.,* Random (RD) and Small-World (SW). These graphs are generated by NetworkX [44]. We also perform the real world test with ten cellphones, as shown in Fig. 1. The cellphones are held by each person in an office room. Each cellphone records the brightness of the natural light at its location and communicates with others in a network. The task is to learn the linear relationship between one certain cellphone's record with others'.

A. Performance of DRC

We first evaluate the convergence feature of DRC. The evaluation compares the estimated set $\boldsymbol{u}_k$ against the global set of data $\boldsymbol{y}$ to measure how incomplete and accurate the estimated set is. We use the step size $\alpha(i)$ and $\beta(i)$ specified in Theorem 3. Besides, for the simulation, we preset an M-dimensional global required data vector $\boldsymbol{y}$ and an M-dimensional orthogonal vector $\boldsymbol{\theta}^*$, *i.e.,* $\langle\boldsymbol{y},\boldsymbol{\theta}^*\rangle=0$. Each node can only observe a single element of $\boldsymbol{y}$ with an additive Gaussian noise. Without loss of generality, we let the k-th node observe the k-th element of $\boldsymbol{y}$. For example, node 1 observes a vector $\boldsymbol{y}_k(i)$ with the first element $y_k^1(i)=y^1+\xi_1(i)$, while all other elements $y_k^j=0, j\neq 1$, where y^1 is the first element in vector $\boldsymbol{y}$ and $\xi_1(i)$ is the observation error, *i.e.,* local variation, following $\mathcal{N}(0,1)$. We learn the relation between the normalized error for each node and the number of iterations. The normalized error for k-th node is defined as $\|\boldsymbol{u}_k(i)-\boldsymbol{y}\|/K$, *i.e.,* the estimation error normalized by the dimension of the vector $\boldsymbol{y}$ [5]. As Fig. 2 demonstrates, the error first rises and then decreases sharply. After the inflection point of the curve, the error declines gradually, when the error is small and close to zero, as proved in Theorem 1. The real world test (Fig. 2-c) presents more fluctuating than the simulation since the $\xi_k(i)$ in the real world is not only affected by the optical noise and the sensor noise, but also deviated by the participator's uncertainty of the direction of holding the cellphone. The latter results in a larger variance than the simulation's setting. Nevertheless, the normalized error still approaches to zero.

B. Performance of DDA

In this section, we evaluate the DDA algorithm based on the results obtained above. Via comparison with other algorithms, we prove that our algorithm has higher efficiency.

In the simulation, we consider a distributed minimization of a sum of loss functions in an l_1-regression based on the data generated by DRC. After $I=100$ iterations, each node obtains an estimate $\boldsymbol{u}_k(I)$ on $\boldsymbol{y}$. Here, we let $\hat{\boldsymbol{y}}_k=\boldsymbol{u}_k$. Thus, the problem becomes that given K vectors, $\hat{\boldsymbol{y}}_k$, to estimate the orthogonal vector $\boldsymbol{\theta}^*$, *i.e.,* a vector $\boldsymbol{\theta}$ is needed for minimizing

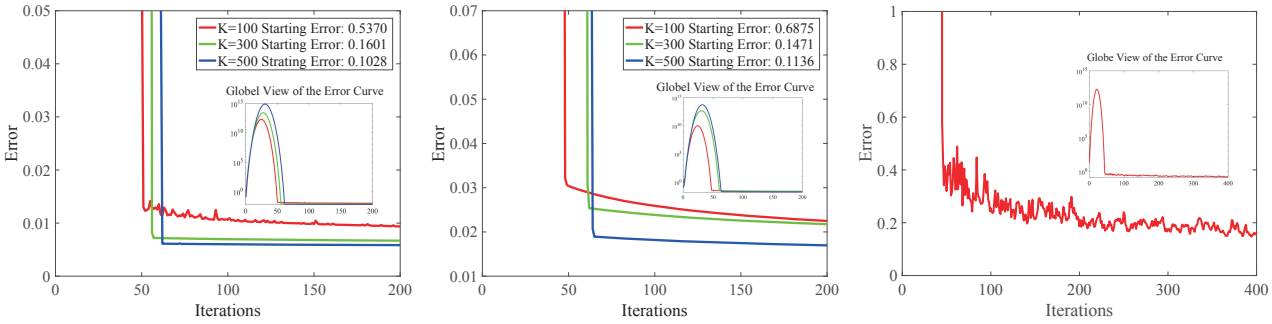
$$f(\boldsymbol{\theta}):=\frac{1}{K}\sum_{i=1}^{K}|\langle\hat{\boldsymbol{y}}_k,\boldsymbol{\theta}\rangle|. \tag{37}$$

We find that f is L-Lipschitz and non-smooth at point $\langle\hat{\boldsymbol{y}}_k,\boldsymbol{\theta}\rangle=0$. We perform simulations on two graph structures, *e.g.,* Random and Small-World as the simulation on DRC. In addition, the step size ω is chosen in the way presented in Theorem 5.

Fig. 3 shows the plot of the function error $\max_k[f(\hat{\boldsymbol{\theta}}_k(T))-f(\boldsymbol{\theta}^*))]$ vs. the number of iterations T for Small-World graphs with size $K\in[100,300,500]$. Also, we show how the convergence time varies as a function of the graph size K.

Next, we show the comparison between our theoretical results (Theorem 5 provides an upper bound of the required iterations) with the simulation results. We show how the

[5]In the figure, we use the average value of all the nodes' estimation errors.



(a) Random. (b) Small-World. (c) Real World test. Starting error is 12.36.

Fig. 2. The estimated error of $\boldsymbol{y}$. The starting error before the first iteration and the global view of the curve are given.

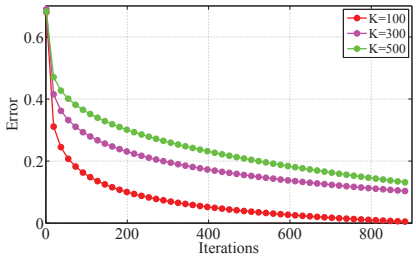


Fig. 3. Function error $\max_k[f(\hat{\boldsymbol{\theta}}_k(T)) - f(\boldsymbol{\theta}^*)]$ decreases as the number of iterations increases (simulations in Small-World network).

number of iterations required varies with respect to the graph size K to achieve a given error $\epsilon = 0.1$. Red curve and red triangles in Fig. 4 represent theoretical bound and simulation results respectively, and both the random graph and small-world graph are considered. In the plot, each point on the line curve denotes the average of 20 trials. Fig. 4 shows a match between our theoretical analysis and simulation results. Moreover, the algorithm presents different trends under the two types of network topology. This implies that the convergence of the algorithm is correlated with the network topology, which corresponds to Theorem 5.

Fig. 4 also compares DDA algorithm with the traditional methods for solving the non-smooth optimization, *i.e.,* the Distributed sub-gradient method (DGM) in [33] and the incremental gradient descent (MIGD) method in [43]. Fig. 4 clearly shows that DDA algorithm has higher efficiency than the other methods.

In the real world test, we use 70% of the data to train the function and the remaining data to estimate the test error. Fig. 5 displays the function error as well as the test error vs. the number of iterations. For $\epsilon = 0.1$, the three methods (DDA, DGM and MIGD) require 103, 237, 678 iterations to achieve the accuracy. These results display similarly to the simulation results, and the test error implies that the process does not result in overfitting problem.

VII. Related Work

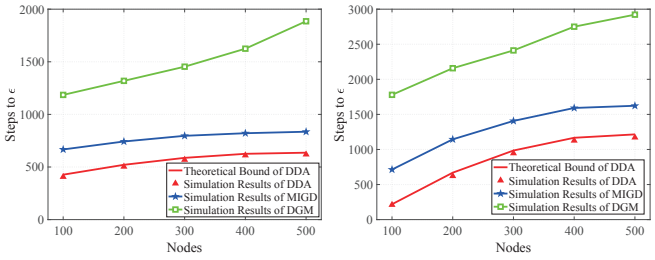
In this section, we briefly discuss the related works that have inspired the design of FINE. As the crowdsoucing/crowdsensing has been the promising technique, many works, for instance [1]–[7], have focused on this issue recently. In these literature, participants report their observations to the coordinator (who is always the decision maker); While in our scenario, every node can be both the participant and the decision maker, who can share its record with neighbors and exploit the data from the network to make a decision.

One of motivations of our work is the concern of privacy. As centralized management of is susceptible to information leakage, FINE chooses to launch the learning process in a distributed manner. However, in distributed systems, components are still vulnerable to adversarial attacks. Previous works [29], [30] therefore constructed consensus-bassed learning model, which leveraged Byzantine Gradient Descent to protect users' privacy in distributed learning. Inspired by their methods, in FINE, we use a simple Distributed Record Completion algorithm to allow each node to obtain global consensus and complete data. To be noticed, as we focus on measuring the communication complexity FINE takes for the whole network to discover the convergence of optimization, the analysis of the cost of local data volume is unfortunately shelved, while existing literatures [27]–[30] provide proper design of reducing the communication overhead in consensus-based learning.

In conventional sensor networks, the existing methods such as [8]–[10], [23], [45]–[48] focus on solving the learning problems in homogeneous networks where each node can obtain complete datasets. In contrast, in order to deal with our crowdsensing scenario, we must consider some practical factors, *i.e.,* heterogeneous in functionality and data errors.

The design of DRC algorithm allows each node to reach global consensus. Different from the previous proposed consensus approaches [24]–[26], [49]–[53], FINE tries to address a more challenging problem in which nodes are heterogeneous in functionality and thereby having incomplete inputs. Moreover, the traditional consensus only processes the initial noisy observations, which might result in severe bias. Conversely, our DRC algorithm processes the successive observation data, and we can thus ensure that all nodes converge to the same global information.

In terms of distributed learning problem, the earlier works [22], [54] based on the well known sub-gradient methods which target at smooth function, produce the convergence rate scaling exponentially in the network size. Then [32], [33] sharpened the result, and obtain an efficiency of $\mathcal{O}(\frac{|\mathcal{V}|^3}{\varepsilon^2})$. This expression, nevertheless, cannot capture the explicit in-



(a) Random Graph. (b) Small-World Graph.

Fig. 4. The iterations number required to achieve a given precision ϵ for DDA and MIGD vs. network size K for (a) Random Graph and (b) Small-World Graph.

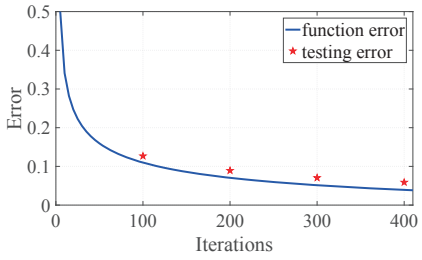


Fig. 5. Function error $\max_k[f(\hat{\boldsymbol{\theta}}_k(T)) - f(\boldsymbol{\theta}^*)]$ decreases as the number of iterations increase (tests in the cellphone network).

fluence of the network topology [21]. Moreover, in [22], the observation is only a scalar rather than a vector in our paper, which is easier to be handled. For the non-smooth function, the incremental based approach in [43], [55]–[57] are argued that has a slow asymptotic convergence rate [21]. Therefore, in [39], researchers proposed a consensus based approach, where the non-smooth function has to be pre-processed by the smooth approximation algorithm. In contrast, in our approach, we do not require the approximation. Besides, the above literature did not handle the issue of data errors. The design of DDA algorithm is inspired by the efficient non-smooth optimization methodology introduced in [20], [21]. Inspired by the method proposed in [20], we realize it in a distributed manner to support the distributed learning problems, and extend [21]'s analysis on the convergence rate by taking the observation noise into consideration.

VIII. Conclusion

In this paper, we present FINE, a learning framework addressing a class of distributed learning problems in heterogeneous crowdsensing networks. FINE allows that (1) terminals obtain incomplete datasets, (2) local error functions are non-smooth, and (3) observation noise is taken into account. Therefore FINE is adaptive to a much wider range of real-world learning applications. We design two important algorithms in FINE: a Distributed Record Completion (DRC) algorithm to ensure each node to acquire complete information in spite of its originally incomplete data acquisition, and a Distributed Dual Average (DDA) algorithm to efficiently solve non-smooth convex optimization problems with observation noise. We prove the convergence of the two algorithms and further derive their convergence rates that guarantee the efficiency of the algorithms. Via extensive simulations and real world examinations, we validate the effectiveness of our design.

Appendix A

In this section, we introduce some classic results which are used to prove the theorems of DRC. We summarize the results from [58] into the following theorem.

Theorem 6. *Define the random sequence* $\left\{\boldsymbol{u}(i) \in \mathbb{R}^N\right\}_{i\geq 0}$ *as:*

$$\boldsymbol{u}(i+1) = \boldsymbol{u}(i) + \alpha(i)\left[R(\boldsymbol{u}(i)) + \Gamma(i+1, \boldsymbol{u}(i))\right],$$

where $R(\cdot) : \mathbb{R}^N \to \mathbb{R}^N$ *is Borel measurable and* $\{\Gamma(i, \boldsymbol{u}(i))\}_{i\geq 0}$ *is a random sequence in* $\mathbb{R}^N$ *on a probability space* $\mathcal{F}$, $\mathcal{P}$, *we assume the following five assumptions*

C.1 *Let* $\mathcal{B}$ *be the Borel algebra of* $\mathbb{R}^N$, *then for time* i, $\Gamma(i, \cdot) : \mathbb{R}^N \times \Omega \to \mathbb{R}^N$ *is* $\mathcal{B} \otimes \mathcal{F}$ *measurable;*

C.2 *The zero-mean random family* $\Gamma(i, \boldsymbol{u}(i))$ *is* $\mathcal{F}_i$ *measurable, where* $\mathcal{F}_i \in \mathcal{F}$, *and it is also independent of* $\mathcal{F}_{i-1}$.

C.3 *We have the function* $V(\boldsymbol{u}(i))$ *and its gradient* $V_{\boldsymbol{y}}$ *that satisfies*

$$V(\boldsymbol{y}^*) = 0,\ V(\boldsymbol{y}) > 0,\ \boldsymbol{y} \neq \boldsymbol{y}^*,\ \lim_{\|\boldsymbol{y}\|\to\infty} V(\boldsymbol{y}) = \infty$$

$$\sup_{\epsilon < \|\boldsymbol{y}-\boldsymbol{y}^*\| < \frac{1}{\epsilon}} (R(\boldsymbol{y}), V_{\boldsymbol{y}}(\boldsymbol{y})) < 0, \forall \epsilon > 0,$$

and the function's second-order partial derivatives are bounded.

C.4 *We can find a pair of numbers* k_1, k_2 *to make the following inequality hold*

$$\|R(\boldsymbol{u}(i))\|^2 + \mathbb{E}\left[\|\Gamma(i+1, \boldsymbol{u}(i))\|^2\right] < k_1(1 + V(\boldsymbol{u}(i))) - k_2(R(\boldsymbol{u}(i)), V_{\boldsymbol{y}}(\boldsymbol{y})).$$

C.5 $\{\alpha(i)\}_{i>0}$ *can be defined properly such that*

$$\alpha(i) > 0,\ \sum_{i\geq 0} \alpha(i) < \infty,\ \sum_{i\geq 0} \alpha^2(i) < \infty.$$

D.1 $R(\boldsymbol{u}(i))$ *can be represented by*

$$R(\boldsymbol{u}(i)) = \boldsymbol{B}(\boldsymbol{u}(i) - \boldsymbol{y}^*) + \delta(\boldsymbol{u}(i)), \tag{38}$$

where

$$\lim_{\boldsymbol{u}(i)\to\boldsymbol{y}^*} \frac{\|\delta(\boldsymbol{u}(i))\|}{\|\boldsymbol{u}(i) - \boldsymbol{y}^*\|} = 0,$$

and $\boldsymbol{B}$ *is a matrix.*

D.2 *Following C.5, we define the* $\{\alpha(i)\}_{i\geq 0}$ *as*

$$\alpha = \frac{a}{i+1},\ \forall i \geq 0, \tag{39}$$

where $a > 0$ *is a constant.*

D.3 *We also define the stable matrix as* $\sum = a\boldsymbol{B} + \frac{1}{2}E$, *where* E *is the* $M \times M$ *identity matrix.*

D.4 *For the matrix,*

$$M(i, \boldsymbol{u}(i)) = \mathbb{E}\left[\Gamma(i+1, \boldsymbol{u}(i))\Gamma^T(i+1, \boldsymbol{u}(i))\right],$$

we have

$$\lim_{i\to\infty, \boldsymbol{u}(i)\to\boldsymbol{y}^*} M(i, \boldsymbol{u}(i)) = S_0.$$

D.5 *There exists $\varepsilon > 0$ such that*

$$\lim_{R\to\infty} \sup_{\|\boldsymbol{u}(i)-\boldsymbol{y}^*\|<\varepsilon} \sup_{i\geq 0} \int_{\|\Gamma(i+1,\boldsymbol{u}(i))\|>R} \|\Gamma(i+1,\boldsymbol{u}(i))\|^2 \, dP = 0.$$

Then we can obtain the following results.

1) *With* **C.1-C.5**,

$$\mathbb{P}\left[\lim_{i\to\infty} \boldsymbol{u}(i) = \boldsymbol{y}^*\right] = 1.$$

2) *With* **C.1-C.5** *and* **D.1-D.5**, *if $i \to \infty$, we have*

$$\sqrt{i}(\boldsymbol{u}(i) - \boldsymbol{y}^*) \to \mathcal{N}(\boldsymbol{0}, S), \tag{40}$$

where $\to$ represents the weak convergence, and

$$S = a^2 \int_0^\infty e^{\Sigma v} S_0 e^{\Sigma^T v} dv.$$

Proof: For a proof, see [58]. ∎

APPENDIX B

In this section, we present proofs of rest theorems.

A. Proof of Theorem 2

Proof: Based on Theorem 6, we demonstrate that the sequence $\{\boldsymbol{y}(i)\}_{i\geq 0}$ meet the assumptions **C.1-C.5**.

1) Verification of Assumptions **C.1-C.2**: Now we reorganize Eq. (5) as Theorem 6. First, noting that $(\mathcal{L}\otimes E_M)(\mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*) = \mathbf{0}_{|\mathcal{V}|M}$, for Eq. (5), we have

$$\begin{aligned}\boldsymbol{u}(i+1) &= \boldsymbol{u}(i) - \alpha(i)(\mathcal{L}\otimes E_M)(\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*) \\ &- \beta(i)\tilde{\mathcal{H}}(\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*) + \beta(i)\bar{\mathcal{H}}(\boldsymbol{y}(i) - \bar{\mathcal{H}}^T(\mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*)).\end{aligned} \tag{41}$$

Then, we define $R(\boldsymbol{u}(i))$ and $\Gamma(i+1,\boldsymbol{u}(i))$ as in (42) and (43), and we then obtain the form of equation in Theorem 6:

$$\boldsymbol{u}(i+1) = \boldsymbol{u}(i) + \alpha(i)[R(\boldsymbol{u}(i)) + \Gamma(i+1,\boldsymbol{u}(i))] \tag{42}$$

$$R(\boldsymbol{u}(i)) = -[(\mathcal{L}\otimes E_M) + \frac{\beta(i)}{\alpha(i)}\tilde{\mathcal{H}}](\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*) \tag{43}$$

$$\Gamma(i+1,\boldsymbol{u}(i)) = \frac{\beta(i)}{\alpha(i)}\bar{\mathcal{H}}(\boldsymbol{y}(i) - \bar{\mathcal{H}}^T(\mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*)) \tag{44}$$

Thus, for a given i, the random family is $\{\Gamma(i+1,\boldsymbol{u}(i))\}_{\boldsymbol{u}(i)\in\mathbb{R}^{M|\mathcal{V}|}}$ which satisfies Assumptions **C.1-C.2**.

2) Verification of Assumption **C.3**: Now we define

$$\begin{aligned}&V(\boldsymbol{u}(i)) = \\ &(\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*)^T \left[(\mathcal{L}\otimes E_M) + \frac{\beta(i)}{\alpha(i)}\tilde{\mathcal{H}}\right](\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*).\end{aligned} \tag{45}$$

It is obviously that the function $V(\boldsymbol{u}(i)) \in \mathbb{C}_2$ and its second-order partial derivatives are bounded. With Lemma 4, it follows that

$$V(\mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*) = 0; V(\boldsymbol{u}(i)) > 0, \forall \boldsymbol{u}(i) \neq \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*, \tag{46}$$

and hence, we can find a constant δ_1 such that

$$\begin{aligned}&(\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*)^T \left[(\mathcal{L}\otimes E_M) + \frac{\beta(i)}{\alpha(i)}\tilde{\mathcal{H}}\right]^2 (\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*) \\ &\geq \delta_1 \left\|\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*\right\|^2, \forall \boldsymbol{u}(i) \in \mathbb{R}^{M|\mathcal{V}|}.\end{aligned} \tag{47}$$

Therefore, we have the supremum of the inner product of $R(\boldsymbol{u}(i))$ and $V_{\boldsymbol{y}}(\boldsymbol{u}(i))$,

$$\begin{aligned}&\sup_{\|\boldsymbol{u}(i)-\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\|>\sigma} (R(\boldsymbol{u}(i)), V_{\boldsymbol{y}}(\boldsymbol{u}(i))) \\ &= -2 \inf_{\|\boldsymbol{u}(i)-\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\|>\sigma} \left\{(\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*)^T \left[(\mathcal{L}\otimes E_M)\right.\right. \\ &\left.\left. + \frac{\beta(i)}{\alpha(i)}\tilde{\mathcal{H}}\right]^2 (\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*)\right\} \\ &\leq -2 \inf_{\|\boldsymbol{u}(i)-\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\|>\sigma} \delta_1 \left\|\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*\right\|^2 \\ &\leq -2\delta_1\sigma^2 < 0.\end{aligned} \tag{48}$$

Thus, **C.3** is satisfied.

3) Verification of Assumption **C.4**: Based on Eq. (43) and (44), we can obtain

$$\begin{aligned}&\|R(\boldsymbol{u}(i))\|^2 = \\ &= (\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*)^T \left[(\mathcal{L}\otimes E_M) + \frac{\beta(i)}{\alpha(i)}\tilde{\mathcal{H}}\right]^2 (\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*) \\ &= -\frac{1}{2}(R(\boldsymbol{u}(i)), V_{\boldsymbol{y}}(\boldsymbol{u}(i))).\end{aligned} \tag{49}$$

$$\begin{aligned}&\mathbb{E}\left[\|\Gamma(i+1,\boldsymbol{u}(i))\|^2\right] \\ &= \frac{\beta^2(i)}{\alpha^2(i)}\mathbb{E}\left[\left\|\bar{\mathcal{H}}(\boldsymbol{y}(i) - \bar{\mathcal{H}}^T(\mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*))\right\|^2\right].\end{aligned} \tag{50}$$

With the Assumption A.1-A.4, the term

$$\mathbb{E}\left[\left\|\bar{\mathcal{H}}(\boldsymbol{y}(i) - \bar{\mathcal{H}}^T(\mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*))\right\|^2\right] \leq \delta_2,$$

where δ_2 is a finite positive constant. From $\frac{\alpha(i)}{\beta(i)} \to \gamma > 0$, we thus have

$$\mathbb{E}\left[\|\Gamma(i+1,\boldsymbol{u}(i))\|^2\right] < \delta_3, \tag{51}$$

where δ_3 is a positive finite constant. We then have

$$\begin{aligned}&\|R(\boldsymbol{u}(i))\|^2 + \mathbb{E}\left[\|\Gamma(i+1,\boldsymbol{u}(i))\|^2\right] \\ &\leq -\frac{1}{2}(R(\boldsymbol{u}(i)), V_{\boldsymbol{y}}(\boldsymbol{u}(i))) + \delta_3 \\ &\leq -\frac{1}{2}(R(\boldsymbol{u}(i)), V_{\boldsymbol{y}}(\boldsymbol{u}(i))) + \delta_3(1 + \left\|\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*\right\|^2) \\ &\leq -\frac{1}{2}(R(\boldsymbol{u}(i)), V_{\boldsymbol{y}}(\boldsymbol{u}(i))) + \delta_3\left(1 + \frac{1}{\delta_1}(\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*)^T \cdot\right. \\ &\left.\left[(\mathcal{L}\otimes E_M) + \frac{\beta(i)}{\alpha(i)}\tilde{\mathcal{H}}\right]^2 (\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes \boldsymbol{y}^*)\right) \text{(by Equation (49))} \\ &\leq -\frac{1}{2}(R(\boldsymbol{u}(i)), V_{\boldsymbol{y}}(\boldsymbol{u}(i))) + \delta_4(1 + V(\boldsymbol{u}(i))),\end{aligned} \tag{52}$$

where $\delta_4 = \max\left\{\delta_3, \frac{\delta_3}{\delta_1}\right\} > 0$. Thus **C.4** is satisfied.

4) Verification of Assumption **C.5**: We can choose appropriate $\{\alpha(i)\}_{i\geq 0}$ to meet **C.5**. For instance, we can use the form in D.2.

To sum up, the Assumptions C.1-C.5 are satisfied, and the theorem is proved. ∎

B. Proof of Theorem 3

Proof: The proof is also based on Theorem 6. Since Assumptions C.1-C.5 are satisfied, we now demonstrate that **D.1-D.5** are satisfied.

1) Verification of Assumptions **D.1-D.3***:* Recalling the definitions of $R(\boldsymbol{u}(i))$ in Eq. (43), we define $\delta(\boldsymbol{u}(i)) \equiv 0$ and

$$\boldsymbol{B} = -[\gamma\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}], \tag{53}$$

where $\gamma \in \mathbb{R}$. Thus **D.1** is satisfied.

Assumption **D.2** can be easily satisfied by choosing the weight sequence appropriate weight sequence $\{\alpha(i)\}_{i\geq 0}$ and $\{\beta(i)\}_{i\geq 0}$.

For Assumption **D.3**, we then let $a > \frac{1}{2\lambda_{\min}(\gamma\mathcal{L}\otimes E_M+\tilde{\mathcal{H}})}$, which can make the following equation stable:

$$\Sigma = -a[\gamma\mathcal{L} \otimes E_M + \tilde{\mathcal{H}}] + \frac{1}{2}E_{M|\mathcal{V}|} = a\boldsymbol{B} + \frac{1}{2}E_{M|\mathcal{V}|}. \tag{54}$$

2) Verification of Assumption **D.4***:* With the i.i.d assumptions, the function

$$\begin{aligned} A(i,\boldsymbol{u}(i)) &= \mathbb{E}[\Gamma(i+1,\boldsymbol{u}(i))\Gamma^T(i+1,\boldsymbol{u}(i))] \\ &= \mathbb{E}\left[(\bar{\mathcal{H}}\boldsymbol{y}(i) - \tilde{\mathcal{H}}\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*)(\bar{\mathcal{H}}\boldsymbol{y}(i) - \tilde{\mathcal{H}}\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*)^T\right]. \end{aligned} \tag{55}$$

is independent of i, and in particular, $A(i,\boldsymbol{u}(i))$ is a constant, because

$$\begin{aligned} &\mathbb{E}\left[(\bar{\mathcal{H}}\boldsymbol{y}(i) - \tilde{\mathcal{H}}\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*)(\bar{\mathcal{H}}\boldsymbol{y}(i) - \tilde{\mathcal{H}}\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*)^T\right] \\ &= \mathbb{E}\left[\bar{\mathcal{H}}\epsilon\epsilon^T\bar{\mathcal{H}}^T\right] = \bar{\mathcal{H}}S_\epsilon\bar{\mathcal{H}}^T. \end{aligned} \tag{56}$$

Thus we have

$$\lim_{i\to\infty,\boldsymbol{u}(i)\to\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*} A(i,\boldsymbol{u}(i)) = \bar{\mathcal{H}}S_\epsilon\bar{\mathcal{H}}^T = S_0. \tag{57}$$

3) Verification of Assumption **D.5***:* From (44) , it follows that

$$\begin{aligned} \|\Gamma(i+1,\boldsymbol{u}(i))\|^2 &= \frac{\beta^2(i)}{\alpha^2(i)}\left\|\bar{\mathcal{H}}\boldsymbol{y}(i) - \tilde{\mathcal{H}}\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\right\|^2 \\ &\leq \frac{4}{\gamma^2}\left\|\bar{\mathcal{H}}\boldsymbol{y}(i) - \tilde{\mathcal{H}}\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\right\|^2. \end{aligned} \tag{58}$$

Given a fixed $\sigma > 0$, for $\left\|\boldsymbol{u}(i) - \mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\right\| < \sigma$, we define another random family as

$$\left\{\tilde{\Gamma}(i+1,\boldsymbol{u}(i))\right\}_{i\geq 0} = \left\{\frac{4}{\gamma^2}\left\|\bar{\mathcal{H}}\boldsymbol{y}(i) - \tilde{\mathcal{H}}\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\right\|^2\right\}_{i\geq 0}. \tag{59}$$

The boundedness of $\mathbb{E}[\left\|\tilde{\Gamma}(i+1,\boldsymbol{u}(i))\right\|^2]$ follows from Chebyshev's inequality that as $R\to\infty$,

$$\begin{aligned} &\sup_{\|\boldsymbol{u}(i)-\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\|<\sigma}\sup_{i\geq 0}\mathbb{P}\left[\left\|\bar{\mathcal{H}}\boldsymbol{y}(i) - \tilde{\mathcal{H}}\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\right\|^2 > R\right] \\ &\leq \frac{1}{R}\sup_{\|\boldsymbol{u}(i)-\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\|<\sigma}\sup_{i\geq 0}\mathbb{E}\left[\left\|\bar{\mathcal{H}}\boldsymbol{y}(i) - \tilde{\mathcal{H}}\mathbf{1}_{|\mathcal{V}|}\otimes\boldsymbol{y}^*\right\|^2\right] \\ &< \frac{\delta_2}{R} \to 0. \end{aligned} \tag{60}$$

The family (59) is thus uniformly integrable. Then $\|\Gamma(i+1,\boldsymbol{u}(i))\|^2$ is also uniformly integrable. The Assumption **D.5** is satisfied.

Up till now, Assumptions **D.1**-**D.5** are satisfied and we can obtain the theorem. ■

APPENDIX C

In this section, we present proofs of the lemmas.

A. Proof of Lemma 2

Proof:

For the convenience of notification, we define the set Θ_C as $\Theta_C = \{\boldsymbol{\theta}|\boldsymbol{\theta}\in\Theta, \phi(\boldsymbol{\theta}) \leq C\}$ and $\Theta^* = \{\boldsymbol{\theta}|\boldsymbol{\theta}\in\Theta, \phi(\boldsymbol{\theta}) \leq \phi(\boldsymbol{\theta}^*)\}$, where the function ϕ is the prox-function defined in **A.4**. Clearly, we have $\Theta_C \subset \Theta^*$ if $\phi(\boldsymbol{\theta}^*) \geq C$.

Then we define the following two functions, Eq. (61) and (62), for the proof.

$$\begin{aligned} \delta_T(C) &= \max_{\boldsymbol{\theta}\in\Theta_C}\left\{\sum_{t=0}^{T}\langle\boldsymbol{g}(t),\boldsymbol{\theta}(t)-\boldsymbol{\theta}\rangle\right\}, \\ &= \max_{\boldsymbol{\theta}\in\Theta_C}\left\{\sum_{t=0}^{T}\langle\boldsymbol{g}(t),\boldsymbol{\theta}(t)-\boldsymbol{\theta}_0\rangle + \sum_{t=0}^{T}\langle\boldsymbol{g}(t),\boldsymbol{\theta}_0-\boldsymbol{\theta}\rangle\right\} \\ &= \sum_{t=0}^{T}\langle\boldsymbol{g}(t),\boldsymbol{\theta}(t)-\boldsymbol{\theta}_0\rangle + \max_{\boldsymbol{\theta}\in\Theta_C}\left\{\left\langle\sum_{t=0}^{T}\boldsymbol{g}(t),\boldsymbol{\theta}_0-\boldsymbol{\theta}\right\rangle\right\} \\ &= \sum_{t=0}^{T}\langle\boldsymbol{g}(t),\boldsymbol{\theta}(t)-\boldsymbol{\theta}_0\rangle + \varepsilon_C(-\boldsymbol{s}_{T+1}), \end{aligned} \tag{61}$$

where $\boldsymbol{s}_{T+1} = \sum_{t=0}^{T}\boldsymbol{g}(t)$, and

$$\begin{aligned} \varepsilon_C(\boldsymbol{s}) &= \max_{\boldsymbol{\theta}\in\Theta_C}\min_{\omega\geq 0}\{\langle\boldsymbol{s},\boldsymbol{\theta}-\boldsymbol{\theta}_0\rangle + \omega(C-\phi(\boldsymbol{\theta}))\} \\ &\leq \min_{\omega\geq 0}\max_{\boldsymbol{\theta}\in\Theta_C}\{\langle\boldsymbol{s},\boldsymbol{\theta}-\boldsymbol{\theta}_0\rangle - \omega\phi(\boldsymbol{\theta})\} + \omega C \\ &= V_\omega(\boldsymbol{s}) + \omega C, \end{aligned} \tag{62}$$

where $V_\omega(\boldsymbol{s})$ is the function in Lemma 1.

Thus we have $\varepsilon_C(\boldsymbol{s}) \leq V_\omega(\boldsymbol{s}) + \omega C$, which further indicates

$$\delta_T(C) \leq \sum_{t=0}^{T}\langle\boldsymbol{g}(t),\boldsymbol{\theta}(t)-\boldsymbol{\theta}_0\rangle + V_\omega(-\boldsymbol{s}_{T+1}) + \omega C \tag{63}$$

$$\begin{aligned} V_{\omega(t+1)}(-\boldsymbol{s}_{t+1}) &\leq V_{\omega(t)}(-\boldsymbol{s}_{t+1}) \\ &\leq V_{\omega(t)}(-\boldsymbol{s}_t) - \langle\boldsymbol{g}(t),\nabla V_{w(t)}(-\boldsymbol{s}_t)\rangle + \frac{\|\boldsymbol{g}(t)\|_*^2}{2\omega(t)\tau} \\ &= V_{\omega(t)}(-\boldsymbol{s}_t) - \langle\boldsymbol{g}(t),\boldsymbol{\theta}(t)-\boldsymbol{\theta}_0\rangle + \frac{\|\boldsymbol{g}(t)\|_*^2}{2\omega(t)\tau}. \end{aligned} \tag{64}$$

Thus, we have for $\forall 1 \leq t \leq T$,

$$\langle\boldsymbol{g}(t),\boldsymbol{\theta}(t)-\boldsymbol{\theta}_0\rangle \leq V_{\omega(t)}(-\boldsymbol{s}_t) - V_{\omega(t+1)}(-\boldsymbol{s}_{t+1}) + \frac{\|\boldsymbol{g}(t)\|_*^2}{2\omega(t)\tau}. \tag{65}$$

Summing all inequalities, we obtain

$$\sum_{t=0}^{T}\langle \boldsymbol{g}(t),\boldsymbol{\theta}(t)-\boldsymbol{\theta}_0\rangle \le V_{\omega(1)}(-\boldsymbol{s}_1)-V_{\omega(T+1)}(-\boldsymbol{s}_{T+1}) + \frac{1}{2\tau}\sum_{t=1}^{T}\frac{\|\boldsymbol{g}(t)\|_*^2}{\omega(t)}. \tag{66}$$

In view of [20],

$$V_{\omega(1)}(-\boldsymbol{s}_1)\le \frac{1}{2\tau\omega(1)}\|\boldsymbol{g}(0)\|_*^2 \le \frac{1}{2\tau\omega(0)}\|\boldsymbol{g}(0)\|_*^2. \tag{67}$$

Thus, we have

$$\sum_{t=0}^{T}\langle \boldsymbol{g}(t),\boldsymbol{\theta}(t)-\boldsymbol{\theta}_0\rangle \le -V_{\omega(T+1)}(-\boldsymbol{s}_{T+1})+\frac{1}{2\tau}\sum_{t=0}^{T}\frac{\|\boldsymbol{g}(t)\|_*^2}{\omega(t)}. \tag{68}$$

Thus, we have

$$\delta_T(C)\le \frac{1}{2\tau}\sum_{t=0}^{T}\frac{\|\boldsymbol{g}(t)\|_*^2}{\omega(t)}+\omega C, \tag{69}$$

which further indicates for $\forall\hat{\boldsymbol{\theta}}\in\Theta_C\subset\Theta^*$,

$$\begin{aligned}\sum_{t=0}^{T}\left\langle \boldsymbol{g}(t),\boldsymbol{\theta}(t)-\hat{\boldsymbol{\theta}}\right\rangle &\le \frac{1}{2\tau}\sum_{t=0}^{T}\frac{\|\boldsymbol{g}(t)\|_*^2}{\omega(t)}+\omega C\\ &\le \frac{1}{2\tau}\frac{\|\boldsymbol{g}(t)\|_*^2}{\omega(t)}+\omega\phi(\boldsymbol{\theta}^*).\end{aligned} \tag{70}$$

■

B. Proof of Lemma 4

Proof: Based on f_k's *L-Lipschitz* continuity, we have

$$\begin{aligned}f(\boldsymbol{\theta}_k(t),\boldsymbol{y}_k)-f(\boldsymbol{\theta}^*,\boldsymbol{y}^*) &= f(\boldsymbol{\theta}_k(t),\boldsymbol{y}_k)-f(\boldsymbol{\varphi}(t),\boldsymbol{y}_k)\\ &+f(\boldsymbol{\varphi}(t),\boldsymbol{y}_k)-f(\boldsymbol{\theta}^*,\boldsymbol{y}_k)+f(\boldsymbol{\theta}^*,\boldsymbol{y}_k)-f(\boldsymbol{\theta}^*,\boldsymbol{y}^*)\\ &\le L\|\boldsymbol{\theta}_k(t)-\boldsymbol{\varphi}(t)\|+f(\boldsymbol{\varphi}(t),\boldsymbol{y}_k)-f(\boldsymbol{\theta}^*,\boldsymbol{y}_k)\\ &+L\|\boldsymbol{y}_k-\boldsymbol{y}^*\|.\end{aligned} \tag{71}$$

Lemma 1 implies that :

$$\|\boldsymbol{\theta}_k(t)-\boldsymbol{\varphi}(t)\|\le\frac{1}{\omega\tau}\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_k(t)\|, \tag{72}$$

Substituting the above inequality into Eq. (71), we obtain the result

$$\begin{aligned}f(\boldsymbol{\theta}_k(t),\boldsymbol{y}_k)-f(\boldsymbol{\theta}^*,\boldsymbol{y}^*) &\le f(\boldsymbol{\varphi}(t),\boldsymbol{y}_k)-f(\boldsymbol{\theta}^*,\boldsymbol{y}_k)\\ &+\frac{L}{\omega\tau}\|\bar{\boldsymbol{\mu}}(t)-\boldsymbol{\mu}_k(t)\|+L\|\boldsymbol{y}_k-\boldsymbol{y}^*\|.\end{aligned} \tag{73}$$

■

C. Proof of Positive Semidefiniteness

In this part, we present a lemma to demonstrate the semidefiniteness of matrix $\left[\alpha(i)(\mathcal{L}\otimes E_M)+\beta(i)\tilde{\mathcal{H}}\right]$.

Lemma 4. *If the DRC algorithm is under the Assumption* **A.1-A.3**, *and both* $\{\alpha(i)\}_{i\ge0}$ *and* $\{\beta(i)\}_{i\ge0}$ *are positive consequences, then for each* $i\ge0$, $\left[\alpha(i)(\mathcal{L}\otimes E_M)+\beta(i)\tilde{\mathcal{H}}\right]$ *is a symmetric positive definite matrix.*

Proof: The symmetricity of matrix $\left[\alpha(i)(\mathcal{L}\otimes E_M)+\beta(i)\tilde{\mathcal{H}}\right]$ for each i is obvious. To prove the positive semidefinite property, we assume, on the contrary, that the matrix $\left[\alpha(i)(\mathcal{L}\otimes E_M)+\beta(i)\tilde{\mathcal{H}}\right]$ is not positive semidefinite. Therefore, according to the definition of positive semidefiniteness, there exists a nonzero vector $\boldsymbol{y}(\neq0)\in\mathbb{R}^{M|\mathcal{V}|}$ and

$$\boldsymbol{y}^T\left[\alpha(i)(\mathcal{L}\otimes E_M)+\beta(i)\tilde{\mathcal{H}}\right]\boldsymbol{y}=0. \tag{74}$$

Due to the positive semidefiniteness of matrix $\mathcal{L}\otimes E_M$ and $\tilde{H}$ as well as the limitation that for each i, both $\alpha(i)$ and $\beta(i)$ are positive, we have

$$\boldsymbol{y}^T(\mathcal{L}\otimes E_M)\boldsymbol{y}=0,\boldsymbol{y}^T\tilde{\mathcal{H}}\boldsymbol{y}=0. \tag{75}$$

Combining the partition as $\boldsymbol{y}=\left[\boldsymbol{y}_1^T,...,\boldsymbol{y}_k^T,\boldsymbol{y}_{|\mathcal{V}|}^T\right]^T,\boldsymbol{y}_k\in\mathbb{R}^M,\forall1\le k\le|\mathcal{V}|$ and (75), we get

$$\begin{aligned}\boldsymbol{y}^T(\mathcal{L}\otimes E_M)\boldsymbol{y} &= \sum_{r=1}^{|\mathcal{V}|}\sum_{s=1}^{|\mathcal{V}|}(\boldsymbol{y}_r,\boldsymbol{y}_j)\mathcal{L}_{rs}\\ &=\sum_{r=1}^{|\mathcal{V}|}\sum_{s=1}^{|\mathcal{V}|}\mathcal{L}_{rs}(\sum_{t=1}^{M}x_r^tx_s^t)=\sum_{t=1}^{M}(\sum_{r=1}^{|\mathcal{V}|}\sum_{s=1}^{|\mathcal{V}|}\mathcal{L}_{rs}x_r^tx_s^t)=0.\end{aligned} \tag{76}$$

Construct a new column vector $\tilde{\boldsymbol{y}}=[\tilde{\boldsymbol{y}}_1^T,...,\tilde{\boldsymbol{y}}_M^T]^T,\tilde{\boldsymbol{y}}_t\in\mathbb{R}^{|\mathcal{V}|},\forall1\le t\le M$. Define each $\tilde{\boldsymbol{y}}$ as $\tilde{\boldsymbol{y}}_l=[x_1^t,...,x_{|\mathcal{V}|}^t]^T$, which indicates that the k-th element of vector $\tilde{\boldsymbol{y}}_t$ is the t-th element of vector $\boldsymbol{y}_k$. From the fact that $\lambda_2(\mathcal{G})>0$, and the definition of second eigenvalue of graph, it follows

$$\lambda_2(\mathcal{G})=\min_{\boldsymbol{y}\perp\mathbf{1}_{|\mathcal{V}|},\boldsymbol{y}\neq\mathbf{0}_{|\mathcal{V}|}}\frac{(\mathcal{L}\boldsymbol{y},\boldsymbol{y})}{(\boldsymbol{y},\boldsymbol{y})}>0. \tag{77}$$

We further assume column vectors $\boldsymbol{\eta}_1,...,\boldsymbol{\eta}_{|\mathcal{V}|}\in\mathbb{R}^{|\mathcal{V}|}$ represent a group of orthonormal basis in $|\mathcal{V}|$-dimensional Euclidean space, and $\boldsymbol{\eta}_1=\frac{1}{\sqrt{|\mathcal{V}|}}\mathbf{1}_{|\mathcal{V}|}$. Thus, vector $\tilde{\boldsymbol{y}}_t$ can be written as a linear summation of orthonormal basis, which is given by

$$\tilde{\boldsymbol{y}}_t=\sum_{h=1}^{|\mathcal{V}|}a_{lh}\boldsymbol{\eta}_h,\forall1\le t\le M. \tag{78}$$

From (77) and (78) we obtain

$$\begin{aligned}\sum_{r=1}^{|\mathcal{V}|}\sum_{s=1}^{|\mathcal{V}|}\mathcal{L}_{rs}x_r^tx_s^t &= (\mathcal{L}\tilde{\boldsymbol{y}}_t,\tilde{\boldsymbol{y}}_t)=\left(\sum_{h=1}^{|\mathcal{V}|}a_{lh}\mathcal{L}\boldsymbol{\eta}_h,\sum_{h=1}^{|\mathcal{V}|}a_{lh}\boldsymbol{\eta}_h\right)\\ &=\sum_{i=1,j=1}^{|\mathcal{V}|}a_{li}a_{lj}(\mathcal{L}\boldsymbol{\eta}_i,\boldsymbol{\eta}_j)=\sum_{j=1}^{|\mathcal{V}|}a_{lj}^2(\mathcal{L}\boldsymbol{\eta_j},\boldsymbol{\eta}_j)\ge0.\end{aligned} \tag{79}$$

Specially, *iff* $\boldsymbol{y}_t\parallel\mathbf{1}_{|\mathcal{V}|}$, we have

$$\sum_{r=1}^{|\mathcal{V}|}\sum_{s=1}^{|\mathcal{V}|}\mathcal{L}_{rs}x_r^tx_s^t=0. \tag{80}$$

Thus, *iff* $\boldsymbol{y}_t \parallel \mathbf{1}_{|\mathcal{V}|}, \forall 1 \leq t \leq M$, we have

$$\boldsymbol{y}^T(\mathcal{L} \otimes E_M)\boldsymbol{y} = \sum_{t=1}^{M}\left(\sum_{r=1}^{|\mathcal{V}|}\sum_{s=1}^{|\mathcal{V}|}\mathcal{L}_{rs}x_r^t x_s^t\right) = 0. \tag{81}$$

Therefore, we have

$$\boldsymbol{y}_k = \boldsymbol{c}, \forall 1 \leq k \leq |\mathcal{V}|, \tag{82}$$

where $\boldsymbol{c} \in \mathbb{R}^M$ and $\boldsymbol{c} \neq \mathbf{0}_M$. Meanwhile, (75) implies that

$$\sum_{k=1}^{|\mathcal{V}|}\boldsymbol{y}_k^T H_k^T H_k \boldsymbol{y}_k = 0. \tag{83}$$

According to the assumption **A.3**, equations (82) and (83) imply

$$\boldsymbol{c}^T\mathcal{H}\boldsymbol{c} = 0. \tag{84}$$

This is a contradiction, since $\mathcal{H}$ is full rank and $\boldsymbol{c} \neq 0$. Thus matrix $[\alpha(i)(\mathcal{L} \otimes E_M) + \beta(i)\tilde{\mathcal{H}}]$ is positive semidefinite for each $i \geq 0$. ■

ACKNOWLEDGEMENT

This work was supported by NSF China (No. 61532012, 61325012, 61521062, 61602303, 61428205).

REFERENCES

[1] K. Han, C. Zhang, and J. Luo, "Taming the uncertainty: Budget limited robust crowdsensing through online learning," *IEEE/ACM Trans. Networking*, vol. 24, no. 3, pp. 1462–1475, 2016.

[2] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Trans. Networking*, vol. 24, no. 3, pp. 1732–1744, 2016.

[3] P. Naghizadeh and M. Liu, "Perceptions and truth: A mechanism design approach to crowd-sourcing reputation," *IEEE/ACM Trans. Networking*, vol. 24, no. 1, pp. 163–176, 2016.

[4] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *Proc. ACM UbiComp*, New York, NY, Sept. 2012.

[5] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Trans. Mobile Computing*, vol. 14, no. 2, pp. 444–457, 2015.

[6] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *Proc. IEEE INFOCOM*, Hong Kong, China, Apr. 2015.

[7] Y. Wang, J. Jiang, and T. Mu, "Context-aware and energy-driven route optimization for fully electric vehicles via crowdsourcing," *IEEE Trans. Intelli. Transpor. Systems*, vol. 14, no. 3, pp. 1331–1345, 2013.

[8] A. Agarwal, S. Chakrabarti, and S. Aggarwal, "Learning to rank networked entities," in *Proc. ACM SIGKDD*, Philadelphia, PA, Aug. 2006.

[9] D. Gavinsky, "Optimally-smooth adaptive boosting and application to agnostic learning," *J. Mach. Learn. Res.*, vol. 4, pp. 101–117, Dec. 2003.

[10] R. A. Servedio, "Smooth boosting and learning with malicious noise," *J. Mach. Learn. Res.*, vol. 4, pp. 633–648, Dec. 2003.

[11] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[12] P. Flach, *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

[13] I. J. Vergara-Laurens, L. G. Jaimes, and M. A. Labrador, "Privacy-preserving mechanisms for crowdsensing: Survey and research challenges," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 855–869, 2017.

[14] L. Pournajaf, L. Xiong, D. A. Garcia-Ulloa, and V. Sunderam, "A survey on privacy in mobile crowd sensing task management," *Dept. Math. Comput. Sci., Emory Univ., Atlanta, GA, USA, Tech. Rep. TR-2014-002*, 2014.

[15] I. Krontiris, M. Langheinrich, and K. Shilton, "Trust and privacy in mobile experience sharing: future challenges and avenues for research," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 50–55, 2014.

[16] D. R. Easterling, L. T. Watson, M. L. Madigan, B. S. Castle, and M. W. Trosset, "Direct search and stochastic optimization applied to two nonconvex nonsmooth problems," in *Proc. Symp. High Performance Computing*, Orlando, FL, Mar. 2012.

[17] Y. Jin, S. Vishwanathan, S. Günter, and N. Schraudolph, "A quasi-newton approach to nonsmooth convex optimization problems in machine learning," *J. Mach. Learn. Res.*, vol. 11, pp. 1145–1200, Mar. 2010.

[18] N. Rao, J. Chin, D. Yau, and C. Ma, "Localization leads to improved distributed detection under non-smooth distributions," in *Proc. Conf. Information Fusion*, Edinburgh, UK, July 2010.

[19] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

[20] ——, "Primal-dual subgradient methods for convex problems," *Mathematical programming*, vol. 120, no. 1, pp. 221–259, 2009.

[21] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *IEEE Trans. on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[22] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[23] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[24] S. Kar and J. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Trans. Signal Proc.*, vol. 56, no. 7, pp. 3315–3326, July 2008.

[25] ——, "Distributed consensus algorithms in sensor networks: Quantized data and random link failures," *IEEE Trans. Signal Proc.*, vol. 58, no. 3, pp. 1383–1400, March 2010.

[26] G. Scutari and S. Barbarossa, "Distributed consensus over wireless sensor networks affected by multipath fading," *IEEE Trans. Signal Proc.*, vol. 56, no. 8, pp. 4100–4106, Aug 2008.

[27] L. Valerio, A. Passarella, and M. Conti, "A communication efficient distributed learning framework for smart environments," *Pervasive and Mobile Computing*, vol. 41, pp. 46–68, 2017.

[28] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *arXiv preprint arXiv:1605.07689*, 2016.

[29] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *arXiv preprint arXiv:1705.05491*, 2017.

[30] L. Su, "Defending distributed systems against adversarial attacks: consensus, consensus-based learning, and statistical learning," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2017.

[31] Y. Nesterov and I. E. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer, 2004, vol. 87.

[32] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Trans. Automatic Control*, vol. 54, no. 11, pp. 2506–2517, 2009.

[33] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Jour. Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

[34] A. Olshevsky, "Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control," *arXiv preprint arXiv:1411.4186*, 2014.

[35] A. Nedić, A. Olshevsky, and C. A. Uribe, "Nonasymptotic convergence rates for cooperative learning over time-varying directed graphs," in *American Control Conference*, Chicago, IL, Jul. 2015, pp. 5884–5889.

[36] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-time analysis and impact of network topology," *arXiv preprint arXiv:1409.8606*, 2014.

[37] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[38] B. Mohar and Y. Alavi, "The laplacian spectrum of graphs," *Graph theory, combinatorics, and applications*, vol. 2, pp. 871–898, 1991.

[39] J. Li, C. Wu, Z. Wu, and Q. Long, "Gradient-free method for nonsmooth distributed optimization," *Journal of Global Optimization*, vol. 61, no. 2, pp. 325–340, 2015.

[40] K. Kiwiel and A. Ruszczynski, *Minimization Methods for Non-differentiable Functions*. Springer, 1985, vol. 3.

[41] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[42] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

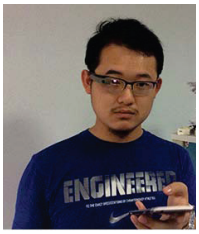
[43] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Jour. Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.

[44] NetworkX, "Networkx: Python software for complex networks," http://github.com/networkx/.
[45] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Trans. Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.
[46] A. Defazio, J. Domke, and T. Caetano, "Finito: A faster, permutable incremental gradient method for big data problems," in *Proc. ICML*, Beijing, China, June 2014.
[47] A. Nedich, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *arXiv preprint arXiv:1607.03218*, 2016.
[48] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Trans. Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.
[49] C. N. Hadjicostis, N. H. Vaidya, and A. D. Domínguez-García, "Robust distributed average consensus via exchange of running sums," *IEEE Trans. on Automatic Control*, vol. 61, no. 6, pp. 1492–1507, 2016.
[50] S. Zhu and B. Chen, "Quantized consensus by the admm: probabilistic versus deterministic quantizers," *IEEE Trans. on Signal Processing*, vol. 64, no. 7, pp. 1700–1713, 2016.
[51] D. Li, Q. Liu, X. Wang, and Z. Yin, "Quantized consensus over directed networks with switching topologies," *Systems & Control Letters*, vol. 65, pp. 13–22, 2014.
[52] K. Cai and H. Ishii, "Average consensus on arbitrary strongly connected digraphs with time-varying topologies," *IEEE Trans. Automatic Control*, vol. 59, no. 4, pp. 1066–1071, 2014.
[53] C. N. Hadjicostis and T. Charalambous, "Average consensus in the presence of delays in directed graph topologies," *IEEE Trans. Automatic Control*, vol. 59, no. 3, pp. 763–768, 2014.
[54] I. Lobel and A. Ozdaglar, "Distributed subgradient methods over random networks," in *Proc. Allerton Conf. Commun., Control, Comput*, Monticello, IL, Sept. 2008.
[55] A. Nedić and D. Bertsekas, "Convergence rate of incremental subgradient algorithms," *Stochastic optimization: algorithms and applications*, pp. 223–264, 2001.
[56] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.
[57] D. P. Bertsekas, "Incremental proximal methods for large scale convex optimization," *Mathematical programming*, vol. 129, no. 2, pp. 163–195, 2011.
[58] M. B. Nevel'son and Khas'minskiĭ, *Stochastic approximation and recursive estimation*. American Mathematical Society Providence, RI, 1973, vol. 47.

Luoyi Fu received her B. E. degree in Electronic Engineering from Shanghai Jiao Tong University, China, in 2009 and Ph.D. degree in Computer Science and Engineering in the same university in 2015. She is currently an Assistant Professor in Department of Computer Science and Engineering in Shanghai Jiao Tong University. Her research of interests are in the area of social networking and big data, scaling laws analysis in wireless networks, connectivity analysis and random graphs.

Songjun Ma received his B. E. degree in Department of Computer Science and Engineering at Shanghai Jiao Tong University, China, 2017. During his undergraduate study, he was working as an research intern supervised by Prof. Xinbing Wang. His research interests include combinatorial optimizationasymptotic analysis and privacy protection in social networks. He will pursue Ph. D. degree in the Massachusetts Institute of Technology (MIT), Massachusetts, USA, 2017.

Lingkun Kong is an undergraduate student in Department of Computer Science at Shanghai Jiao Tong University, China. He is currently working as a research intern supervised by Prof. Xinbing Wang. His research interests include distributed system, theoretical networking and big-scale network analysis.



Shiyu Liang received his bachelor degree from the Department of Electronic Engineering at Shanghai Jiao Tong University, China. He is currently pursuing the Ph.D. degree under the supervision of Prof. R. Srikant in University of Illinois at Urbana-Champaign, USA.

Xinbing Wang received the B.S. degree (with honors.) from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1998, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001. He received the Ph.D. degree, major in the Department of electrical and Computer Engineering, minor in the Department of Mathematics, North Carolina State University, Raleigh, in 2006. Currently, he is a professor in the Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. Dr. Wang has been an associate editor for IEEE/ACM Transactions on Networking and IEEE Transactions on Mobile Computing, and the member of the Technical Program Committees of several conferences including ACM MobiCom 2012, ACM MobiHoc 2012-2014, IEEE INFOCOM 2009-2017.