Consensus and Approximation-based Distribution Statistics in Network Systems

Yushan Li, Qing Jiao, Han Wang and Jianping He

***Abstract*—The data distribution statistics in network systems have gained increasing attention. In this paper, we study the problem of obtaining the global probability density function (PDF) across the network, where each agent only holds a portion of the overall distribution in the network. Our work significantly deviates from traditional distribution/density problems, which infer the underlying common distribution from multiple *i.i.d.* random samples. The agents initially hold interval-wise and related data distribution, and no central node is required while no prior knowledge about the network structure is available. To practice, we propose a novel consensus and approximation-based distribution statistics (CADS) algorithm. The key insight is utilizing polynomials to approximate the initial distribution function of each agent, such that a unified and compact representation for various function forms is used to improve the storage flexibility and computation efficiency during the consensus-based interaction rounds. Another salient design is that when the intervals of all agents are different, the proposed algorithm is able to adapt the dynamically changing interval range, and avoid massive interval storage cost by a refit operation for large-scale networks. We prove the convergence and the bounded PDF approximation error of the proposed algorithm, and analyze the statistics performance and algorithm complexity. Simulations illustrate the effectiveness of the CADS algorithm.**

I. Introduction

With the rapid development of computing and networking technologies, numerous distributed systems such as social networks, smart grids, wireless sensor networks, have been widely deployed in various environments. As the scales of these systems grow, there is an increasing requirement for an efficient method to infer the underlying regularity or model from the massive data, which are distributed across multiple agents (nodes). Particularly, exploiting the statistics from the massive data has been an ambitious goal in database research and beyond in recent years [1].

In this paper, we aim to obtain the global probability density function (PDF) of the data distribution across the network, where each agent has only a portion of the overall distribution initially. This problem is motivated by the popularity of distributed networks and efficient data statistics requirements, and is of great importance in many applications. For example, a sensor network can be deployed in various positions of a lake to collect local states of water quality and cooperatively compute the overall distribution [2], [3]. In social networks, an IT company can improve the service quality by merging the user's information among its massively-deployed data nodes and analyzing the overall distribution characteristics.

In the literature, a lot of methods concerning distribution estimation have been developed. For example, the authors in [4] proposed a geometric approach with communication constraints to establish information-theoretic lower bounds. In [5], a gossip-based distributed kernel density estimation algorithm was designed for various forms of distribution. Furthermore, multiple estimation schemes were developed for other important considerations, e.g., high dimensional distribution [6] and privacy preservation [7], [8]. Pleasant results under different performance requirements are achieved by these methods. However, there still remain two notable issues. First, in most of them, each agent holds a *i.i.d.* sample of the unknown distribution, and is required to communicate with a central agent [6]. Second, the data distribution is commonly represented in histogram form [9], which may degenerate the estimation performance in many situations. Overall, the major goal of the traditional distribution estimation problem is to optimize the storage and communication cost, the privacy concern, and the estimation error or the tradeoff between these performance requirements [10], [11].

By contrast, our work emphasizes more on the "statistics" side, and can be seen as the extended direction of the above works, i.e., each agent has obtained the distribution function of the data statistics of its own interest, and needs to further obtain the global density function. There are a few works that involve a similar setting. In [12], a distributed method was proposed to estimate the distribution using stochastic approximation, but the data range is known to all agents beforehand, and every single value is needed to be recorded. A consensus-based data statistics algorithm was designed in [13] to obtain the PDF of the data distribution in the network, and a distributed social sampling algorithm in [14] also took a similar idea to estimate the opinion formation. Nevertheless, both methods only work for single-point data on each agent.

Based on the above observations, this paper is dedicated to designing a distributed and general method to obtain the global PDF in the network, where the initial data distribution in each agent is interval-wise and no prior global knowledge about the network structure is needed. Specifically, we utilize the tools of classic consensus algorithm [15], [16] and function approximation theory [17], [18] to design an Consensus and Approximation-based Distribution Statistics (CADS) al-

The authors are with the Dept. of Automation, Shanghai Jiao Tong University, the Key Laboratory of System Control and Information Processing, Ministry of Education of China, and Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai, China. E-mail: {yushan_li, jiaoqing, JTUniversity, jphe}@sjtu.edu.cn.

This research work is partially sponsored by the National Key R&D Program of China 2017YFE0114600 and NSF of China 61973218.

gorithm. The challenges lie in two aspects. First, the function forms of the initial distribution are various and no prior knowledge about the network is available, making it difficult to efficiently implement the information interaction among agents, especially considering the data processing ability of each agent is limited. Second, even if a compact form is approximated to represent various distributions, the obtained representation space and the data intervals of all agents are not necessarily identical, further complicating the algorithm design to deal with the dynamically changing dimension and interval. The main contributions are summarized as follows.

- We develop a novel CADS algorithm exploiting ideas from consensus algorithm and approximation theory to obtain the global distribution statistic, without relying on a central agent and any other prior knowledge about the network structure or global distributions.
- By approximating the initial distribution functions in polynomial forms and designing a refit mechanism for massive intervals situation, the proposed algorithm enjoys a compact representation for a majority of functions with finite elements, high computation efficiency during the interaction rounds, and flexible adaptability for the dynamic interval changes.
- We prove the convergence of the CADS algorithm and the bounded estimation error for the obtained global PDF. The performance analysis in terms of expectation, variance and complexity analysis is provided. Extensive simulations illustrate the flexibility and effectiveness of the proposed method.

The remainder of this paper is organized as follows. In Section II, some basics of graph, consensus and approximation theory are introduced, and the problem is stated. The CADS algorithm is proposed in Section III. Simulation results are shown in Section IV. Finally, Section V concludes the paper.

II. Preliminaries And Problem Formulation

A. Network Model

A distributed network system is modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \cdots, n\}$ is the finite set of nodes (agents) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. An edge $(i, j) \in \mathcal{E}$ indicates that i and j can exchange information with other. Denote $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ as the neighbor set of i, $\mathcal{N}_i^o = \mathcal{N}_i \cup \{i\}$, and d is the diameter of $\mathcal{G}$. A pair of nodes i and j is said to be connected if there exists a path sequence from i to j in $\mathcal{E}$. $\mathcal{G}$ is said to be connected if $\forall i, j \in \mathcal{V}(i \neq j)$, i and j are connected. Assume the computation and information exchange in the network is synchronous and the network is connected with a fixed topology. Each agent in the network broadcasts its current state, and updates its state iteratively following a designated protocol.

B. Classic Consensus Algorithms

There are mainly two types of classic consensus algorithms, max/min and average consensus algorithms. Let $x_i(k)$ be the state of agent i in iteration k and $\boldsymbol{x}(k)$ be the state vector of all agent. A maximum consensus algorithm is described as

$$x_i(k+1) = \max\{x_i(k), x_j(k) | i \in V, j \in \mathcal{N}_i\}.$$

It is proved that maximum consensus is achieved after d iterations when the network is connected with a fixed topology [19]. The result is likewise for minimum consensus.

For average consensus, agent i updates its value by

$$x_i(k+1) = w_{ii} x_i(k) + \sum\nolimits_{j \in \mathcal{N}_i} w_{ij} x_j(k), i \in \mathcal{V}, \quad (1)$$

where $w_{ij} = w_{ji} \geq 0$ is a non-negative weight, satisfying $w_{ij} = 0$ for $j \notin \mathcal{N}_i$, $w_{ii} = 1 - \sum_{j \in \mathcal{N}_i} w_{ij}$. (1) can be rewritten in the matrix form, given by $\boldsymbol{x}(k+1) = W\boldsymbol{x}(k)$, with $W = [w_{ij}]_{n \times n}$. Note that $W\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^\mathsf{T} W = \mathbf{1}^\mathsf{T}$ (where $\mathbf{1}$ denotes the vector of all ones), and it is proven that average consensus is achieved asymptotically [15], i.e.,

$$\lim_{k \to \infty} x_i(k) = \bar{x} = \frac{1}{n} \sum\nolimits_{j=1}^{n} x_i(0). \quad (2)$$

C. Function Approximation by Polynomials

There are plenty of function approximation methods by the feat of various function basis in the literature, e.g., kernel functions, Fourier series and polynomial functions. Among these methods, the mature polynomial approximation method is widely used, due to its flexible differential and interval properties and compact representation form. For example, in the computation implementation, after the polynomial basis is preset, only the coefficients of different terms are used for function operation. The feasibility of polynomial approximation is theoretically guaranteed by the famous Weierstrass approximation theorem.

Lemma 1. *(Weierstrass approximation theorem) For arbitrary bounded continuous function $F(x), x \in [a, b]$, given $\epsilon > 0$, there exists a polynomial function $P_m(x)$ such that*

$$|P_m(x) - F(x)| < \epsilon, \forall x \in [a, b], \quad (3)$$

where m denotes the degree.

Based on Lemma 1, plenty of methods are developed to determine the $P_m(x)$, like Lagrange's interpolation, spline interpolation and Chebyshev approximation. In this paper, we adopt the Chebyshev method. Denote the function basis as $\{T_j(x)|_{j=0}^m\}$, where $T_j(x)$ is the j-th Chebyshev polynomial. Then, for a Lipschitz continuous function $F(x)$ $(x \in [a, b])$, its polynomial approximation of m degree is given by

$$P_m(x) = \boldsymbol{p}^\mathsf{T} \cdot \boldsymbol{T}(x) = \sum\nolimits_{j=0}^{m} c_j T_j(x), x \in [a, b], \quad (4)$$

where $\boldsymbol{p} = [c_0, c_1, \cdots, c_m]^\mathsf{T} \in \mathbb{R}^{m+1}$ is the Chebyshev coefficient vector, and $\boldsymbol{T} = [T_0, T_1, \cdots, T_m]^\mathsf{T} \in \mathbb{R}^{m+1}$ is the function basis vector (the variable x is omitted). Specifically, as m increases, $P_m(x)$ converges to $F(x)$ uniformly [18], i.e.,

$$\lim_{m \to \infty} |P_m(x) - F(x)| = 0. \quad (5)$$

Note that the convergence speed of (5) is determined by the smoothness of $F(x)$ and the detailed steps of the approximation are presented in next Section.

D. Problem of Interest

Considering a group of agents $\mathcal{G}=(\mathcal{V},\mathcal{E})$ with undirected and fixed interaction topology, agent $i\in\mathcal{V}$ initially has a partial data distribution of interest, given by $F_i(x)$, $x\in[a_i,b_i]$, and no prior information about the network structure. The goal is to estimate the real PDF of the overall data distribution in the network, mathematically represented by

$$\min_{\hat{f}_i}\max_{x\in X}|\hat{f}_i(x)-f(x)|,\ i\in\mathcal{V},\tag{6}$$

where $X=\bigcup_{i=1:n}[a_i,b_i]$, and $f(x)=\frac{\sum_{i=1}^{n}F_i(x,0)}{\int_X\sum_{i=1}^{n}F_i(x,0)dx}$ is the real global PDF of overall data distribution.

To solve this problem, the key point is to exploit a unified and compact representation for various function forms, and find a feasible way to obtain the global PDF in a fully distributed way, while without relying on any prior knowledge about the network. Borrowing the idea of consensus and polynomial approximation theories, we are able to design an efficient algorithm to achieve the goal with bounded error. The following assumption holds in this paper.

Assumption 1. *Each distribution function $F_i(x)$ possessed by agent i is Lipschitz continuous on the interval $[a_i,b_i]$, and $\int_{a_i}^{b_i}F_i(x)dx\geq(b_i-a_i)$ holds.*

III. CADS: Design and Analysis

In this section, we propose the CADS algorithm to obtain the global PDF of the data distribution for each agent. In the following, we first consider a situation where the interval of the estimated distribution is the same for all agents. Based on that, different intervals between neighbors are taken into account, and further analysis in terms of approximation accuracy and algorithm complexity is provided.

A. Initial Approximation

In this part, we briefly illustrate how to implement the initial approximation for the data distribution $F_i(x)$.

For simplicity of the expression, the order notation m in $P(x;m)$ is omitted hereafter. According to the adaptive Chebyshev interpolation method [20], [21], for agent $i\in\mathcal{V}$, it constructs the polynomial approximation $P_i(x)$ corresponding to $F_i(x)$ on $[a_i,b_i]$, satisfying

$$|P_i(x)-F_i(x)|\leq\epsilon,\quad\forall x\in[a_i,b_i].\tag{7}$$

To achieve this, agent i first initializes $m_i=2$ and starts to calculate a Chebyshev interpolant of degree m_i. Then, $F_i(x)$ at the (m_i+1)-point grid $S_{m_i}\triangleq\{x_s\}$ is evaluated by

$$x_s=\frac{b_i-a_i}{2}\cos\left(\frac{s\pi}{m_i}\right)+\frac{a_i+b_i}{2},\tag{8}$$

where $s=0,1,\ldots,m_i$. Then, the Chebyshev coefficients of the interpolant $P_i(x)$ is computed by [22]

$$c_{ij}=\frac{F_i(x_0)+F_i(x_{m_i})\cos(j\pi)}{m_i}+\frac{2}{m_i}\sum_{s=1}^{m_i-1}F_i(x_s)\cos\left(\frac{js\pi}{m_i}\right),\tag{9}$$

where $j=0,1,\ldots,m_i$. The degree m_i is doubled at every iteration until the following stopping criterion is met, given by

$$\max_{x_s\in\left(S_{2m_i}-S_{m_i}\right)}|F_i(x_s)-P_i(x_s)|\leq\epsilon.\tag{10}$$

According to [20], except for very few well-designed counterexamples, the above procedures obtain a $P_i(x)$ satisfying (7) for almost all $F_i(x)$ satisfying Assumption 1. For more advanced approximation procedures, which are not the focus of this paper, the readers are referred to [18].

Algorithm 1 CADS Algorithm for Identical Data Intervals

```
Input: Initial local distribution functions $F_i(x), x\in[a,b]$;
Output: The estimation of the global PDF of overall network
for $i\leftarrow 1$ to $n$ do
    Calculate the polynomial $P_i(x,0)=\boldsymbol{p}_i(0)\cdot\boldsymbol{T}(x)$ to
    approximate $F_i(x)$ by (8)-(10);
end for
for $k\leftarrow 0$ to $k_{\max}$ do
    for $i\leftarrow 1$ to $n$ do
        implement (11)-(13) consequently;
    end for
end for
Obtain the PDF $\hat{f}_i(x,k)$ by (14);
```

B. PDF Calculation with Identical Intervals

In this stage, we begin with a simple scenario where $[a_i,b_i]=[a,b](\forall i\in\mathcal{V})$ to illustrate how to obtain the global PDF, and analyze the approximation performance.

With the same data interval, every agent only needs to update and store the coefficient vector during the interaction with its neighbors. Note that the orders of the initially approximated Chebyshev polynomial $P_i(x)$ are not necessarily the same, due to various data distributions on each agent. Therefore, once an agent receives the coefficient vectors of its neighbors, the first step is to uniformize the dimensions of these vectors. Let $\tilde{m}_i(k)$ represent the the maximum dimension of $\{\boldsymbol{p}_j(k),j\in\mathcal{N}_i^o\}$ by $\tilde{m}_i(k)$, and it is calculated by

$$\tilde{m}_i(k)=\max\left\{m_j(k)|j\in\mathcal{N}_i^o\right\}.\tag{11}$$

Based on $\tilde{m}_i$, agent i needs to augment the coefficient vector into the form of $\mathbb{R}^{\tilde{m}_i+1}$, i.e.,

$$\tilde{\boldsymbol{p}}_j(k)=[\mathbf{0}^{1\times(\tilde{m}_j(k)-m_j(k))},\boldsymbol{p}_j^{\mathsf{T}}(k)]^{\mathsf{T}},\ j\in\mathcal{N}_i^o.\tag{12}$$

After the coefficient augmentation, agent i updates its own coefficient vector by the average consensus protocol,

$$\boldsymbol{p}_i(k+1)=\sum\nolimits_{j\in\mathcal{N}_i^o}w_{ij}\tilde{\boldsymbol{p}}_i(k).\tag{13}$$

Then, the approximated PDF by agent i is calculated by

$$\hat{f}_i(x,k+1)=\frac{\boldsymbol{p}_i(k+1)\cdot\boldsymbol{T}(x)}{\int_a^b\boldsymbol{p}_i(k+1)\cdot\boldsymbol{T}(x)dx},\tag{14}$$

where the dimension of $\boldsymbol{T}(x)$ is the same as $\boldsymbol{p}_i^{\mathsf{T}}(k+1)$ at iteration k. The whole procedure is summarized as Algorithm 1, whose performance is demonstrated by the following theorem.

Theorem 1. *By Algorithm 1, when iteration $k\to\infty$, we have*

$$\lim_{k\to\infty} \hat{f}_i(x,k) = \lim_{k\to\infty} \frac{\tilde{P}_i(x,k)}{\int_{\tilde{a}}^{\tilde{b}} \tilde{P}_i(x,k)dx} = \hat{f}(x,0), \tag{15}$$

and the asymptotic estimation error satisfies

$$\lim_{k\to\infty} |\hat{f}_i(x,k) - f(x)| < (1 + \frac{1}{b-a})\epsilon. \tag{16}$$

The result is easy to prove since $\lim\limits_{k\to\infty} \boldsymbol{p}_i(k)$ converges to the average of $\{\boldsymbol{p}_j(0), j \in \mathcal{V}\}$. Then, one substitutes the inequalities $|\bar{P}(x) - \bar{F}(x)| \leq \epsilon$, $Q_f \geq (b-a)$ and $\frac{\bar{P}(x)}{Q_p} < 1$ into $\lim\limits_{k\to\infty} |\hat{f}_i(x,k) - f(x)|$, and obtains the conclusion. Due to the space limit, the details are omitted here. Theorem 1 illustrates that each agent will asymptotically obtain the global approximated PDF of the data distribution without any global information about the network. In addition, the estimation error of the PDF is strictly limited by an upper bound, which is close to the approximation error of the distribution function.

C. PDF Calculation with Different Intervals

Considering the general situation where the data intervals of the agents are not identical, we introduce an extended algorithm to deal with issues of dynamically changing intervals and huge storage costs.

First, for agent i, its data interval at iteration k is

$$\begin{cases} a_i(k) = \min\{a_j(k-1),\ j \in \mathcal{N}_i^o\}, \\ b_i(k) = \max\{b_j(k-1),\ j \in \mathcal{N}_i^o\}. \end{cases} \tag{17}$$

As each agent needs to take into consideration the interval difference at every iteration, extra interval parameters are required to be stored. Denote the intersection points in $\bigcup\limits_{j=1:n}[a_i(k), b_i(k)]$ as $\{a_{i,0}(k), a_{i,1}(k), \cdots, a_{i,s}(k)\}$ with $a_{i,0}(k) = a_i(k)$ and $a_{i,s}(k) = b_i(k)$, satisfying

$$[a_i(k), b_i(k)] = \bigcup_{j=1:s} [a_{i,j-1}(k), a_{i,j}(k)]. \tag{18}$$

Then, at each subinterval $[a_{i,j-1}(k), a_{i,j}(k)]$, a group of coefficient vector is stored, and is updated in the same way as (13). However, as the iteration increases, the cost of storing multiple interval parameters can be tremendously high. To demonstrate this effect, we first present the following result.

Lemma 2. *If the initial data intervals are not identical, then there are at most $(2n-1)$ subintervals needed to stored in each agent after iteration $k \geq d$.*

This result can be easily proved by a limit analysis and recurrence method, and is omitted here. Lemma 2 shows that an agent needs to store $(2n-1)$ groups of coefficient vectors at most. However, the storage cost is not desirable in large-scale networks. To tackle this issue, a refit mechanism is designed.

First, we introduce a preset parameter $S_{\max}$ as the largest number of subintervals that an agent can tolerate. Let $S_i(k)$ be the number of intervals of $P_i(x,k)$ at each iteration. If $S_i(k) \geq S_{\max}$, a refit operation is triggered. Note that $P_i(x,k)$ are discontinuous on the intersection points $\{a_{i,j}, j = 1, \cdots, s\}$. To refit the a new $P_i(x,k)$ with $S_i(k) = 1$, the discontinuity issue of $P_i(x,k)$ is addressed by a local smoothing on the domain $[a_{i,j} - \sigma, a_{i,j} - \sigma]$, where $\sigma > 0$ is a preset small constant. Specifically, we seek to a line equation that cross through points $(a_{i,j} - \sigma, P_i(a_{i,j} - \sigma, k))$ and $(a_{i,j} + \sigma, P_i(a_{i,j} + \sigma, k))$ by solving the following equations of b_1^{ij} and b_2^{ij},

$$\begin{cases} P_i(a_{i,j} - \sigma, k) = b_1^{ij}(a_{i,j} - \sigma) + b_2^{ij}, \\ P_i(a_{i,j} + \sigma, k) = b_1^{ij}(a_{i,j} + \sigma) + b_2^{ij}. \end{cases} \tag{19}$$

Then, $P_i(x,k)$ is reassigned on each $[a_{i,j} - \sigma, a_{i,j} + \sigma]$ by

$$P_i(x,k) = b_1^{ij}x + b_2^{ij},\ x \in [a_{i,j} - \sigma, a_{i,j} + \sigma]. \tag{20}$$

With the continuity of $P_i(x,k)$ on $[a_i(k), b_i(k)]$ guaranteed, agent i implements the Chebyshev interpolation method as formulated in Section III-A with a given approximation error bound δ. Following this, agent i only needs to send the coefficient vector of newly approximated $P_i(x,k)$ with $S_i(k) = 1$ to its neighbors, and updates its coefficient vector and data intervals until $S_i(k) = S_{\max}$ triggers a refit operation again. The whole process is summarized in Algorithm 2.

Algorithm 2 CADS Algorithm for Different Data Intervals

```
Input: $\{F_i(x,0), x \in [a_i, b_i]\}$, interval number bound $S_{\max}$
Output: The estimation of the global PDF of overall network
for $i \leftarrow 1$ to $n$ do
    Calculate the polynomial $P_i(x,0) = \boldsymbol{p}_i(0) \cdot \boldsymbol{T}(x)$ to approximate $F_i(x,0)$ by (8)-(10);
end for
for $k \leftarrow 0$ to $k_{\max}$ do
    for $i \leftarrow 1$ to $n$ do
        Agent $i$ receives $\{\boldsymbol{p}_j(k)\}$ and $\{a_j(k), b_j(k)\}, j \in \mathcal{N}_i$;
        Rearrange $\{a_j(k), b_j(k), j \in \mathcal{N}_i^o\}$ into an ascending order, $\{a_{i,0}(k), a_{i,1}(k), \cdots, a_{i,s}(k)\}$, and $S_i(k) = s$;
        Update coefficient vector $\boldsymbol{p_i}(k+1)$ by (11)-(13) at each interval $[a_{i,j-1}(k), a_{i,j}(k)]$, $j = 1, \cdots, S_i(k)\}$;
        if $S_i(k) > S_{\max}$ then
            Adopt local smoothing on $P_i(x,k+1)$ at the $\sigma$-domain of $\{a_{i,j}(k+1), j = 1, \cdots, S_i(k+1) - 1\}$ by (19)-(20);
            Do the approximation procedures (8)-(10) and obtain a new $P_i(x,k+1)$ with $S_i(k+1) = 1$;
        end if
    end for
end for
Obtain the PDF $\hat{f}_i(x,k)$ by (23);
```

To mathematically demonstrate the performance of Algorithm 2, the domain-augmented versions of $F_i(x,k)$ and $P_i(x,k)$ are defined as

$$\tilde{F}_i(x,k) = \begin{cases} F_i(x,k), & x \in [a_i(k), b_i(k)], \\ 0, & \text{others}. \end{cases} \tag{21}$$

$$\tilde{P}_i(x,k) = \begin{cases} P_i(x,k), & x \in [a_i(k), b_i(k)], \\ 0, & \text{others}. \end{cases} \tag{22}$$

Accordingly, the real and estimated PDF of the data distribution in the network are rewritten as

$$f(x)=\frac{\sum\limits_{i=1}^{N}\tilde{F}_i(x,0)}{\int_{\tilde{a}}^{\tilde{b}}\sum\limits_{i=1}^{N}\tilde{F}_i(x,0)dx},\ \hat{f}_i(x,k)=\frac{\sum\limits_{i=1}^{N}\tilde{P}_i(x,k)}{\int_{\tilde{a}}^{\tilde{b}}\sum\limits_{i=1}^{N}\tilde{P}_i(x,k)dx}. \tag{23}$$

where $\tilde{a}=\min\{a_i(0), i\in\mathcal{V}\}$ and $\tilde{b}=\max\{b_i(0), i\in\mathcal{V}\}$.

Theorem 2. *When the data intervals of all agent are different, by Algorithm 2, the estimation error of $\hat{f}_i(x,k)$ satisfies*

$$\lim_{k\to\infty}|\hat{f}_i(x,k)-f(x)|<(1+\frac{1}{\tilde{b}-\tilde{a}})(\epsilon+\left\lfloor\frac{d}{S_{\max}}\right\rfloor\delta), \tag{24}$$

where $\lfloor\cdot\rfloor$ indicates the floor integer of a variable.

Proof. First, we introduce two auxiliary functions $\bar{P}(x,k)$ and $\bar{F}(x,k)$ as well as their integrals, given by

$$\bar{P}(x,k)=\frac{1}{n}\sum\nolimits_{i=1}^{n}\tilde{P}_i(x,k),\ Q_p(k)=\int_a^b\bar{P}(x,k)dx, \tag{25}$$

$$\bar{F}(x,k)=\frac{1}{n}\sum\nolimits_{i=1}^{n}\tilde{F}_i(x,k),\ Q_f(k)=\int_a^b\bar{F}(x,k)dx. \tag{26}$$

Then, suppose that at iteration k_1, the number of subintervals $S_i(k_1)=L_{\max}$ and a refit operation will be triggered for the first time. Denote $\bar{P}'(x,k_1)$ as the corresponding polynomial of $\bar{P}(x,k_1)$ before the refit, and we have

$$\begin{aligned}&\left|\bar{P}(x,k_1)-\bar{F}(x,k_1)\right|\\ \leq&\left|\bar{P}(x,k_1)-\bar{P}'(x,k_1)\right|+\left|\bar{P}'(x,k_1)-\bar{F}(x,k_1)\right|\leq\epsilon+\delta.\end{aligned} \tag{27}$$

Similarly, if agent i has implemented K times refit operations, there will be K intermediate functions in (27), and we obtain

$$\left|\bar{P}(x,k)-\bar{F}(x,k)\right|\leq\epsilon+K\delta. \tag{28}$$

Note that when iteration $k\geq d$, each agent is aware of all the subintervals and no more refit operations are required. Therefore, taking $\{\tilde{P}_i(x,d), i\in\mathcal{V}\}$ as the new initial state vectors and utilizing the conclusion of Theorem 1, we have

$$\lim_{k\to\infty}\hat{f}_i(x,k)=\lim_{k\to\infty}\frac{\bar{P}_i(x,k)}{\int_{\tilde{a}}^{\tilde{b}}\bar{P}_i(x,k)dx}=\hat{f}(x,d). \tag{29}$$

Since the maximum number of the refit operations among all agents is bounded by $\left\lfloor\frac{d}{S_{\max}}\right\rfloor$, we have $K\leq\lfloor\frac{d}{S_{\max}}\rfloor$. Finally, from (29) and the boundedness of K, one infers that

$$\begin{aligned}&\lim_{k\to\infty}|\hat{f}_i(x,k)-f(x)|=\left|\frac{\hat{P}(x,d)}{Q_p(d)}-\frac{\hat{F}(x,0)}{Q_f(0)}\right|\\ \leq&\frac{|\bar{P}(x,d)-\bar{F}(x,0)|}{Q_f(0)}+\frac{|Q_p(d)-Q_f(0)|\,\bar{P}(x,d)}{Q_p(d)Q_f(0)}\\ <&\frac{\epsilon+K\delta}{Q_f(0)}+\frac{|Q_p(d)-Q_f(0)|}{Q_f(0)}\leq\frac{\epsilon+K\delta}{Q_f(0)}+\frac{(\epsilon+K\delta)(\tilde{b}-\tilde{a})}{Q_f(0)}\\ <&(1+\frac{1}{\tilde{b}-\tilde{a}})(\epsilon+K\delta)\leq(1+\frac{1}{\tilde{b}-\tilde{a}})(\epsilon+\left\lfloor\frac{d}{S_{\max}}\right\rfloor\delta).\end{aligned} \tag{30}$$

The proof is completed. □

Theorem 2 illustrates that extra polynomial approximations in Algorithm 2 will be implemented at most $\lfloor\frac{d}{S_{\max}}\rfloor$ times, and the final PDF will converge to the one at iteration d instead of the initial time. Due to the lack of global information about the network structure and data intervals, the growing storage cost as n increases is inevitable. Therefore, the preset $S_{\max}$ makes a tradeoff between the storage cost and estimation performance. The approximation performance and complexity analysis will be provided in the following subsection.

D. Performance Evaluation

In this part, we evaluate the CADS algorithm performance in terms of expectation, variance and algorithm complexity.

For ease of notation, we define the l-order integral of absolute function $|x|$ on $[\tilde{a},\tilde{b}]$ as

$$Q_x^{(l)}=\int_{\tilde{a}}^{\tilde{b}}|x|^l dx. \tag{31}$$

The error bounds in Theorem 1-2 are uniformly denoted by

$$J=\begin{cases}(1+\dfrac{1}{b-a})\epsilon,\ \text{if } [a_i,b_i]=[a,b],\ \forall i\in\mathcal{V},\\ (1+\dfrac{1}{\tilde{b}-\tilde{a}})(\epsilon+\left\lfloor\dfrac{d}{S_{\max}}\right\rfloor\delta),\ \text{otherwise.}\end{cases} \tag{32}$$

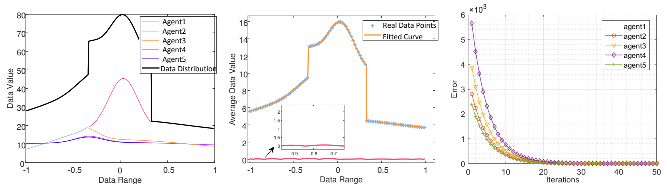
Theorem 3. *By Algorithm 1 and 2, the expectation and variance of x under $\hat{f}(x)$ satisfy*

$$\begin{cases}|\boldsymbol{E}(\hat{x})-\boldsymbol{E}(x)|<JQ_x^{(1)},\\ |\boldsymbol{D}(\hat{x})-\boldsymbol{D}(x)|<(J^2+3J)(Q_x^{(1)})^2.\end{cases} \tag{33}$$

The proof of Theorem 3 is similar with that of Theorem 2, additionally utilizing the properties $\hat{f}(x)\leq f(x)+J<1+J$ and $Q_x^{(2)}\leq(Q_x^{(1)})^2$, and the details are omitted here. Theorem 3 illustrates the approximation performance of the proposed algorithm in terms of the expectation and variance. Specifically, if the data intervals are all identical, then the error bound is only related to the approximation error ϵ, with complexity $\mathcal{O}(\epsilon)$ for $\boldsymbol{E}(\hat{x})$, and $\mathcal{O}(\epsilon^2)$ for $\boldsymbol{D}(\hat{x})$, independent on the network size n. However, when the data intervals are different, the error bounds also relate to the network structure, with complexity $\mathcal{O}(d)$ for $\boldsymbol{E}(\hat{x})$, and $\mathcal{O}(d^2)$ for $\boldsymbol{D}(\hat{x})$. Note that the complexity can be reduced to $\mathcal{O}(n\log\frac{m_{\max}}{\varepsilon})$ with additional assumptions [23].

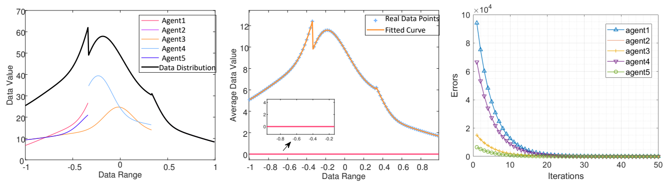
IV. Simulation

The simulations are conducted in a distributed network setup. Without loss of generality, we present the results that contain five agents and three data intervals in total. The agents are connected by a randomly generated undirected network, and thus a doubly stochastic interaction matrix is guaranteed. Due to space limit, we directly present the results of the general form of CADS Algorithm, i.e., Algorithm 2. In every group experiment, the data is generated randomly from various combination of exponential function, Gaussian function, trigonometric function and general polynomial function in different intervals, $[-1,-\frac{1}{3}]$, $[-\frac{1}{3},\frac{1}{3}]$, $[\frac{1}{3},1]$. The degree of the



(a) (b) (c)

Fig. 1. Example 1 with the Chebyshev polynomial degree of 8. (a) the real data of each agent and the overall distribution (black line). (b) fitted curve compared with real data points and its error (black line). (c) the estimation error of the global PDF for all agents during the interaction rounds.



(a) (b) (c)

Fig. 2. Example 2 with the Chebyshev polynomial degree of 15. (a) the real data of each agent and the overall distribution (black line). (b) fitted curve compared with real data points and its error (black line). (c) the estimation error of the global PDF for all agents during the interaction rounds.

approximation polynomial is also changed to verify the final PDF approximation performance.

Two representative examples are presented in Fig. 1 and Fig. 2 with the polynomial degree of 8 and 15, respectively. Fig. 1(a) and Fig. 2(a) show the real data generated by each agent in three intervals and their summation in the whole data range. Fig. 1(b) and Fig. 2(b) show the comparison of the fitted curve and the real data points. The deviation between the obtained PDF and the real one is depicted in a red line. From a global perspective, the asymptotical estimation error in terms of the coefficient vector of example 1 and 2 are presented in Fig. 1(c) and Fig. 2(c), which illustrates the convergence of the CADS algorithm. Note that the initial estimation error is large because the initial polynomial vector of each agent deviates a lot from the initial global one, and the higher polynomial orders are used, the initial estimation error will be larger.

By comparison, when the degree of the polynomial is large, the real data is fitted accurately by the CADS algorithm. Even when a low degree is used, a satisfying fitted curve is still obtained with small error. The results verify the flexibility of the CADS algorithm for various and composite data distribution.

V. Conclusion

In this paper, we investigate the PDF estimation of the data state distribution in a distributed network system. A CADS algorithm is proposed to obtain the global PDF for each agent in the network, without any prior information of the data distribution or the network structure. This algorithm is adaptive with the dynamic change of data intervals and coefficient dimensions, and guarantees the approximation performance with bounded error with low computation and space complexity. We conduct extensive simulations with different degrees of polynomials to demonstrate the effectiveness of the proposed algorithm. Future directions include extending the algorithm to more complicated data distributions, and further reduce the computation cost and estimation error bound.

References

[1] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[2] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2013.

[3] S. He, H.-S. Shin, S. Xu, and A. Tsourdos, "Distributed estimation over a low-cost sensor network: A review of state-of-the-art," *Information Fusion*, vol. 54, pp. 21–43, 2020.

[4] Y. Han, A. Özgür, and T. Weissman, "Geometric lower bounds for distributed parameter estimation under communication constraints," in *Conference on Learning Theory*. PMLR, 2018.

[5] Y. Hu, H. Chen, J.-g. Lou, and J. Li, "Distributed density estimation using non-parametric statistics," in *Proceedings of IEEE International Conference on Distributed Computing Systems*, 2007.

[6] Y. Han, P. Mukherjee, A. Ozgur, and T. Weissman, "Distributed statistical estimation of high-dimensional and nonparametric distributions," in *Proceedings of IEEE ISIT*, 2018.

[7] M. Ye and A. Barg, "Optimal schemes for discrete distribution estimation under locally differential privacy," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5662–5676, 2018.

[8] P. Kairouz, K. Bonawitz, and D. Ramage, "Discrete distribution estimation under local privacy," in *Conference on Machine Learning*. PMLR, 2016.

[9] I. Diakonikolas, E. Grigorescu, J. Li, A. Natarajan, K. Onak, and L. Schmidt, "Communication-efficient distributed learning of discrete distributions," in *Proceedings of Advances in Neural Information Processing Systems*, 2017.

[10] J. Acharya, I. Diakonikolas, C. Hegde, J. Z. Li, and L. Schmidt, "Fast and near-optimal algorithms for approximating distributions by histograms," in *Proceedings of the ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 2015.

[11] J. Acharya, Z. Sun, and H. Zhang, "Hadamard response: Estimating distributions privately, efficiently, and with little communication," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019.

[12] A. D. Sarwate and T. Javidi, "Distributed learning of distributions via social sampling," *IEEE Transactions on Automatic Control*, vol. 60, no. 1, pp. 34–45, 2014.

[13] Y. Cai, J. He, W. Yu, and X. Guan, "Consensus-based data statistics in distributed network systems," in *Proceedings of IEEE Conference on Decision and Control*, 2018.

[14] Q. Liu, X. He, and H. Fang, "Asymptotic properties of distributed social sampling algorithm," *Science China Information Sciences*, vol. 63, no. 1, pp. 1–15, 2020.

[15] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[16] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *in Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[17] K.-G. Steffens, *The history of approximation theory: From Euler to Bernstein*. Springer Science and Business Media, 2007.

[18] L. N. Trefethen, *Approximation theory and approximation practice*. SIAM, 2013.

[19] J. He, P. Cheng, L. Shi, J. Chen, and Y. Sun, "Time synchronization in WSNs: A maximum-value-based consensus approach," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 660–675, 2013.

[20] J. P. Boyd, *Solving transcendental equations: The Chebyshev polynomial proxy and other numerical rootfinders, perturbation series and oracles*. SIAM, 2014.

[21] Z. He, J. He, C. Chen, and X. Guan, "CPCA: A Chebyshev proxy and consensus based algorithm for general distributed optimization," in *Proceedings of IEEE American Control Conference*, 2020.

[22] A. Gil, J. Segura, and N. M. Temme, *Numerical methods for special functions*. SIAM, 2007.

[23] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.