EFFICIENT LEARNING BY CONSENSUS OVER REGULAR NETWORKS

Zhiyuan Weng and Petar M. Djurić

Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, NY 11790, USA
E-mails: {zhiyuan.weng, petar.djuric}@stonybrook.edu

ABSTRACT

In a network, each agent communicates with its neighbors. All the agents have initial observations, and they update their beliefs with the average of the beliefs in their neighborhoods. It is well known that in the long run, the network will reach consensus. However, the agents do not necessarily converge to the global average of the initial observations of all the agents in the network. Instead, the result is always a weighted average. Moreover, it takes infinite time for the process to converge. In this paper, we address regular networks of agents, where each agent (node) has the same number of agents. We propose a method that allows agents in these networks to learn the global average using the history of its local average in finite time.

Index Terms— Consensus, efficient learning, learning in agent networks, regular graphs.

1. INTRODUCTION

We have a network of N agents, where agent n has an initial observation s_n. The agents form a regular network, that is, each agent in the network has the same number of neighbors. We are interested in how an agent can learn the global average of the initial observations, i.e., $\frac{1}{N}\sum_{n=1}^{N} s_n$. Without a control center, this is not an easy task. If each agent updates its belief with the local average of its neighborhood at each iteration, its belief will not necessarily converge to the global average of the initial observations, although consensus is guaranteed [1, 2]. Alternatively, with the knowledge of the network topology, it is possible to design a set of weighting coefficients for each agent to compute the local weighted average so that all the agents converge to the global average in the long run as discussed in [3]. The drawbacks with this method is that the topology of the network must be known to each agent, and it takes infinite time to converge to the exact global average.

In this work, we propose a learning method that allows each agent to learn the global average within $2N$ iterations. Specifically, at each iteration, agents compute their local average in their neighborhood and keep a record of the local averages. After $2N$ iterations, the agents can use the relation between the local averages in the sequence to infer the structure of the network and recover the global average. We prove that the proposed algorithm is guaranteed to recover the global average for all regular graphs.

There is some related work in the literature. In [4] and [5], the focus is on the rate of convergence: the former tries to find the mixing matrix that leads to the highest convergence rate; the latter looks for the structure that accelerates the convergence. An optimal control scheme for achieving consensus is proposed in [6]. In [7], it is shown that an agent in a network is able to calculate any functions of the initial observations of all the agents by using any constant weights for the local computation given that some parameters are pre-set based on the network structure. In [8], the authors proposed an efficient learning algorithm which enables agents to learn the global average in finite time for any graphs. The algorithm, however, requires complete information of the topology of the graph. In [9], the authors consider the problem of designing a sequence of mixing matrices such that the agents learn the global average in minimum time. By contrast, in our work we assume that the structure and the mixing matrix are given. We focus on how and what an agent can learn in the given setting. In other words, we consider the learning from the perspective of agents.

The notation in the paper is as follows: We denote matrices by uppercase letters in bold font; scalars by plain letters; vectors by lowercase letters in bold font; all vector variables are column vectors; $\mathbf{A} = (a_{i,j})$ means that we denote by a_{ij} the entry of the ith row and jth column of the matrix $\mathbf{A}$; $(\cdot)^\top$ indicates the transpose operator; $\mathbf{I}$ is the identity matrix.

The paper is organized as follows. We formulate the problem in Section 2, and present the proposed method in Section 3. In Section 4, we provide a simple example to illustrate how the method works. In Section 5, we show that for regular networks the proposed method is guaranteed to work and how the steps of learning can be simplified. We conclude the paper with Section 6.

2. PROBLEM FORMULATION

Suppose we have a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of nodes, or agents, and $|\mathcal{V}| = N$. The symbol $\mathcal{E}$ denotes the set of edges. Two agents may communicate with each other if there is an edge between them. There is a true state μ_0. At the beginning, each agent receives an observation, which is a normal random variable with mean μ_0 and unit variance. Suppose that agent n observes s_n. Our objective is to estimate the true state μ_0 in a distributed way. Apparently, in this case, the efficient estimator is the global average:

$$\hat{\mu}_0 = \frac{1}{N}\sum_{n=1}^{N} s_n. \tag{1}$$

Therefore our objective is to achieve (1) through belief propagation among the agents. Note that the consensus algorithm does not lead to (1) for general graphs since the mixing matrices are not necessarily doubly stochastic matrices. We put our effort on the inference of the structure and the recovery of the global average within finite time. Here are our basic assumptions: (1) at every iteration, each agent broadcasts its state to its neighbors; (2) after broadcasting, each agent updates its state by taking the average of the information it received from its neighbors; (3) the topology of the network is

This work was supported by NSF under Award CCF-1320626.

a fixed undirected connected graph; (4) each agent knows the total number of agents in the network.

The adjacency matrix $\mathbf{A} = (a_{i,j})$ is defined as

$$a_{i,j} = \begin{cases} 1 & (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

We can define a stochastic matrix $\mathbf{B} = (b_{i,j})$ whose elements are given by

$$b_{i,j} = \begin{cases} \frac{a_{i,j}}{d_i + 1} & i \neq j \\ \frac{1}{d_i + 1} & i = j \end{cases}, \tag{3}$$

where d_i is the degree of node i. We denote by $x_{n,t}$ the value agent n has after t iterations. At the beginning, the state of each agent is s_n. Thus $x_{n,1} = s_n$. At each iteration, every agent performs a consensus step as follows:

$$x_{n,t} = \frac{1}{d_n + 1} \left(x_{n,t-1} + \sum_{(n,j) \in \mathcal{E}} x_{j,t-1} \right). \tag{4}$$

Let $\mathbf{x}(t) = [x_{1,t}, \cdots, x_{N,t}]^\top$, $\mathbf{s} = [s_1, \cdots, s_N]^\top$. It is not difficult to see that

$$\mathbf{x}(t) = \mathbf{B}^{t-1}\mathbf{s}. \tag{5}$$

We consider the problem from the perspective of an agent, say agent n. Our problem becomes that given $x_{n,t}$ for $t \in \{1, 2, \cdots, T\}$, we would like to find the global average $\hat{\mu}_0$. Later we will see that we require at most $2N$ records of local averages. In other words, $x_{n,t}$ for $t > 2N$ does not provide additional information about $\hat{\mu}_0$. Let $\mathbf{x}_{n,T} = [x_{n,1}, \cdots, x_{n,T}]^\top$. Let $\mathbf{c}_{n,t}$ be the nth row of $\mathbf{B}^{t-1}$. Although $\mathbf{c}_t$ comes from a row of a matrix, we still define it as a column vector for the sake of notational consistency. We define $\mathbf{C}_{n,T}$ as

$$\mathbf{C}_{n,T} = \begin{bmatrix} \mathbf{c}_{n,1}^\top \\ \mathbf{c}_{n,2}^\top \\ \vdots \\ \mathbf{c}_{n,T}^\top \end{bmatrix}. \tag{6}$$

Then we have

$$\mathbf{x}_{n,T} = \mathbf{C}_{n,T}\mathbf{s}. \tag{7}$$

Since $\mathbf{C}_{n,T}$ has only N columns, $\text{rank}(\mathbf{C}_{n,T}) \leq N$. $\mathbf{C}_{n,T}$ has an important property:

Lemma 1. *Suppose* $\text{rank}(\mathbf{C}_{n,T}) = K$, *and let* $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_N]^\top$ *be the vector such that*

$$\boldsymbol{\alpha}^\top \mathbf{C}_{n,K} = \mathbf{c}_{n,K+1}^\top. \tag{8}$$

Then we have

$$\boldsymbol{\alpha}^\top \begin{bmatrix} \mathbf{c}_{n,m}^\top \\ \mathbf{c}_{n,m+1}^\top \\ \vdots \\ \mathbf{c}_{n,m+K-1}^\top \end{bmatrix} = \mathbf{c}_{n,m+K}^\top. \tag{9}$$

The lemma can be easily proved if we notice that $\mathbf{c}_{n,m+1}^\top = \mathbf{c}_{n,m}^\top \mathbf{B}$. This property guarantees that the first K rows or more generally every K consecutive rows of $\mathbf{C}_{n,T}$ is a maximal linearly independent set of rows for $\mathbf{C}_{n,T}$. Our main result is as follows.

Theorem 1. *In a network with topology being a regular graph, given the consensus information received within* $2N$ *iterations, an agent is able to compute* $\frac{1}{N}\sum_{n=1}^{N} s_n$ *accurately with probability 1.*

3. THE PROPOSED ALGORITHM

In this section, we introduce our proposed learning algorithm. We defer the analysis and discussion to Section 5. During the consensus procedure, agent n records a sequence of local averages, say $x_{n,1}, x_{n,2}, ...$ To simplify the notation, we denote $x_{n,t}$ by x_t. Given the sequence, define a matrix $\mathbf{X}_k$ as

$$\mathbf{X}_k = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \\ x_2 & x_3 & \cdots & x_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_k & x_{k+1} & \cdots & x_{2k-1} \end{bmatrix}, \tag{10}$$

for $k = 1, 2, \cdots, T$. Note that $\mathbf{X}_N$ does not need to be of full rank. If $\mathbf{X}_N$ has rank K, then $\mathbf{X}_K$ must be a full rank matrix due to Lemma 1. Then let $\boldsymbol{\alpha}$ be

$$\boldsymbol{\alpha} = \mathbf{X}_K^{-1} \begin{bmatrix} x_{K+1} \\ \vdots \\ x_{2K} \end{bmatrix}. \tag{11}$$

Let all connected graphs with N nodes be our candidate graphs. We then apply brute-force search and check these candidates one by one, looking for the graph that *best* fits the sequence of local averages. Because a different node in a graph sees different topologies, we need to check the record with every node in every candidate graph. Specifically, given a graph $\mathcal{H}$ and its node n, we use the consensus matrix defined by $\mathcal{H}$ to construct $\mathbf{C}_{n,N}$ as defined in (6). We first compare the ranks. If the rank of $\mathbf{C}_{n,N}$ is not equal to K, we discard this case and continue to the next candidate node in $\mathcal{H}$. If they are equal, we then check whether $\boldsymbol{\alpha}^\top \mathbf{C}_{n,K} = \mathbf{c}_{n,K+1}^\top$. If it holds, we conclude that the network of the agent has a topology as graph $\mathcal{H}$. Because the objective is to find $\frac{1}{N}\sum_{j=1}^{N} s_j$, we look for a set of weighting coefficients β_j that makes

$$\sum_j \beta_j x_j = \frac{1}{N}\sum_{j=1}^{N} s_j. \tag{12}$$

Note that for an arbitrary graph, it is not always possible to find β_j that makes (12) hold. But in such case, we can instead find the coefficients that make the estimate $\hat{\mu}_0$ with minimum variance. Before we start to prove our theorem, we provide a simple but illustrative example.

4. AN EXAMPLE

We start with a simple example. Suppose there are only four nodes in a graph. The number of unlabeled connected graphs with four nodes is 6 [10], and they are shown in Fig. 1. Suppose that the actual graph is (c) and it is labeled as shown in Fig. 2. Now, given a sequence of observations $x_1, x_2, \cdots, x_8$, we construct $\mathbf{X}_K$ for $K = 1, 2, 3$ and 4. Then, (with probability 1) we will have the following:

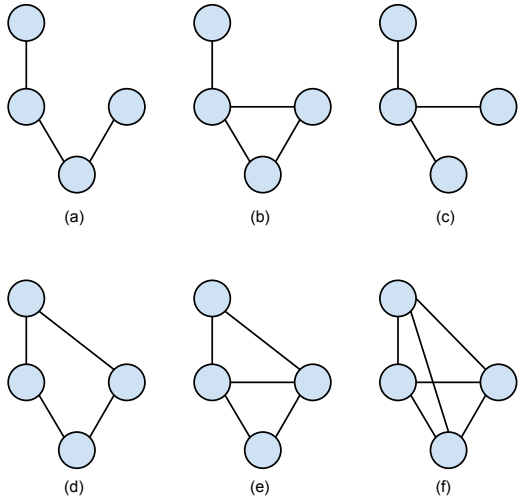

Fig. 1. All the connected graphs with four nodes.

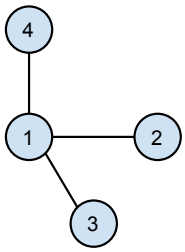

Fig. 2. The actual graph with labeled nodes.

For agent 1, rank $(\mathbf{X}_K) = 2$, $\boldsymbol{\alpha} = \left[\frac{1}{4}, \frac{3}{4}\right]^\top$. We check whether $\boldsymbol{\alpha}\mathbf{C}_{n,2} = \mathbf{c}_{n,3}^\top$ for $n = 1, 2, 3$ and 4, and for all different $\mathbf{C}_{n,T}$ of the six graphs. In this case, the only possible case is graph (c) with $n = 1$.

For agents 2, 3 and 4, rank $(\mathbf{X}_K) = 3$, $\boldsymbol{\alpha} = \left[-\frac{1}{8}, -\frac{1}{8}, \frac{5}{4}\right]^\top$. Similarly we check whether $\boldsymbol{\alpha}\mathbf{C}_{n,3} = \mathbf{c}_{n,4}^\top$ for $n = 1, 2, 3$ and 4, and for all different $\mathbf{C}_{n,T}$ of the six graphs. The possible cases are graph (c) with $n = 2, 3$ and 4. Note that agents $\{2, 3, 4\}$ are equivalent, i.e., there exists an automorphism [11] of the graph which maps i to j for $i, j \in \{2, 3, 4\}$. Intuitively speaking, these agents see the same structure. Therefore there is no way and no need to distinguish agents 2, 3 and 4.

Next we would like to recover the average. We shall find a set of coefficients $\beta_1, \beta_2, \beta_3$ and β_4 such that $\sum_{i=1}^4 x_i\beta_i$ is the average. For node 1, $\mathbf{C}_{1,4}$ is

$$\mathbf{C}_{1,4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{7}{16} & \frac{3}{16} & \frac{3}{16} & \frac{3}{16} \\ \frac{25}{64} & \frac{13}{64} & \frac{13}{64} & \frac{13}{64} \end{bmatrix}, \tag{13}$$

and we can see that $\boldsymbol{\beta} = [0, 1, 0, 0]^\top$ satisfies $\boldsymbol{\beta}^\top\mathbf{C} = \mathbf{1}^\top$. For $i = 2, 3$ and 4, $\mathbf{C}_{i,4}$ is

$$\mathbf{C}_{i,4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{13}{32} & \frac{9}{32} & \frac{5}{32} & \frac{5}{32} \end{bmatrix}. \tag{14}$$

The order of the columns might change for different i. A possible solution is $\boldsymbol{\beta} = [0, -4, 8, 0]^\top$. We note that, in general, there might be multiple solutions; there also might be no solutions.

As shown in the example, the proposed method has the following steps: we first use the observations to construct X_K, and compute $\boldsymbol{\alpha}$. We then calculate the roots of the polynomial defined by $\boldsymbol{\alpha}$. We assume those roots are the eigenvalues of the graph. Then we try to use the eigenvalues to find the topology of the graph. As long as the topology is known, the consensus matrix can be derived. Then we look for a set of weighting coefficients to combine the observations and achieve the global average. We note that for general graphs, the method does not always work. First, the roots of the polynomial defined by $\boldsymbol{\alpha}$ are not necessarily the eigenvalues of the graph. Second, the topology of a graph is not necessarily defined by the eigenvalues. Third, even if the structure is known exactly, a set of weighting coefficients that satisfy (12) does not necessarily exist.

5. RECOVERY OF GLOBAL AVERAGE

In this section, we show that the proposed algorithm achieves the objective if the network is a regular graph. For regular graphs, the easy part is that the consensus matrices become symmetric and doubly stochastic. Those matrices enjoy some nice properties that make them easy to analyze.

Given $\mathbf{x}_{n,T}$, we construct $\mathbf{X}_k$ as in (10) and find the maximum rank K. Then we calculate the vector $\boldsymbol{\alpha}$ according to (11). Note that $\boldsymbol{\alpha}$ is all we need to recover the global average. According to the definition of $\boldsymbol{\alpha}$, we have

$$\begin{bmatrix} \mathbf{X}_K & \begin{bmatrix} x_{n,K+1} \\ \vdots \\ x_{n,2K} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ -1 \end{bmatrix} = \mathbf{0}. \tag{15}$$

We only look at $x_{n,1}, \cdots, x_{n,K+1}$:

$$\begin{bmatrix} \boldsymbol{\alpha}^\top & -1 \end{bmatrix} \begin{bmatrix} x_{n,1} \\ \vdots \\ x_{n,K+1} \end{bmatrix} = 0. \tag{16}$$

Because $x_{n,t} = \mathbf{c}_{n,t}^\top\mathbf{s}$, (16) becomes

$$\begin{bmatrix} \boldsymbol{\alpha}^\top & -1 \end{bmatrix} \mathbf{C}_{n,K+1}\mathbf{s} = 0, \tag{17}$$

and it is *likely* that

$$\begin{bmatrix} \boldsymbol{\alpha}^\top & -1 \end{bmatrix} \mathbf{C}_{n,K+1} = \mathbf{0}. \tag{18}$$

This is not absolutely guaranteed because $\mathbf{s}$, as a random vector, might be orthogonal to one of the rows of $\mathbf{C}_{n,T}$. If this is the case, it means that $\mathbf{s}$ falls in a proper subspace of $\mathbb{R}^N$, which can happen with only probability zero. That is why we claim that our algorithm achieves the objective with probability 1.

Next, we show that $\boldsymbol{\alpha}$ in (18) provides information about the eigenvalues of the matrix $\mathbf{B}$. The underlying principle is closely related to Cayley−Hamilton theorem [12]. In fact, we have the following lemma:

Lemma 2. *The roots of the polynomial*

$$\lambda^K - \alpha_K\lambda^{K-1} \cdots - \alpha_2\lambda - \alpha_1 = 0 \tag{19}$$

are the eigenvalues of $\mathbf{B}$.

Proof: Define a matrix $\mathbf{M}$ as

$$\mathbf{M} = \alpha_1 \mathbf{I} + \alpha_2 \mathbf{B} + \cdots + \alpha_K \mathbf{B}^{K-1} - \mathbf{B}^K. \tag{20}$$

Note that (18) tells us that the nth row of $\mathbf{M}$ is zero. Because we only consider regular graphs, $\mathbf{B}$ is symmetric. We can decompose $\mathbf{B}$ as

$$\mathbf{B} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top, \tag{21}$$

where $\mathbf{U} = (u_{ij})$ is an orthonormal matrix with the columns being the eigenvectors of $\mathbf{B}$, and $\mathbf{\Lambda}$ is a diagonal matrix with the diagonal entries being the eigenvalues. We substitute (21) into (20), and we obtain

$$\mathbf{M} = \mathbf{U} \left(\sum_{j=1}^{K} \alpha_j \mathbf{\Lambda}^{j-1} - \mathbf{\Lambda}^K \right) \mathbf{U}^\top. \tag{22}$$

Define $\mathbf{V} = \sum_{j=1}^{K} \alpha_j \mathbf{\Lambda}^{j-1} - \mathbf{\Lambda}^K$, where $\mathbf{V}$ is a diagonal matrix with diagonal entry v_i for $i \in \{1, \cdots, N\}$. Since the nth row of $\mathbf{M}$ is zero, we have

$$\begin{bmatrix} u_{n1} & \cdots & u_{nN} \end{bmatrix} \mathbf{V} \mathbf{U}^\top = \mathbf{0}^\top. \tag{23}$$

As an eigenspace of a symmetric matrix, $\mathbf{U}$ is full rank. Therefore we have

$$\begin{bmatrix} u_{n1} & \cdots & u_{nN} \end{bmatrix} \mathbf{V} = \mathbf{0}^\top. \tag{24}$$

Consequently, $u_{ni} = 0$, or $v_i = 0$, or both of them are equal to zero. To make (23) hold, v_i must be zero for those i where $u_{ni} \neq 0$. If $u_{ni} = 0$, v_i is not required to be zero. Since each zero v_i specifies an equation of an eigenvalue, we say λ_i is *blotted out* if $u_{ni} = 0$. Those eigenvalues that have not been blotted out are said to be *visible*. Then we must have K visible eigenvalues. To see this, if the number of visible eigenvalues is smaller than K, there must exist a smaller integer, say K_2 that satisfies (23), which leads to a contradiction because K is the largest integer that makes $\mathbf{X}_K$ full rank. Suppose $v_i = 0$ for $i \in \{i_1, \cdots, i_K\}$. Then the set of equations specified by zero v_i is

$$\sum_{j=1}^{K} \alpha_j \lambda_{i_k}^{j-1} = \lambda_{i_k}^K \quad \text{for } k = 1, \cdots, K. \tag{25}$$

We rewrite them into a matrix form and have

$$\begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{K-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_N & \cdots & \lambda_N^{K-1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_K \end{bmatrix} = \begin{bmatrix} \lambda_1^K \\ \lambda_2^K \\ \vdots \\ \lambda_N^K \end{bmatrix}. \tag{26}$$

If all the eigenvalues are visible, it is clear that the value of K should be equal to the order of the minimal polynomial of $\mathbf{B}$. Also since $\mathbf{B}$ diagonalizable, K is equal to the number of distinct eigenvalues of $\mathbf{B}$ [13]. However, if some eigenvalues are blotted out, K will be smaller than the order of the minimal polynomial. But $\{\lambda_{i_k}\}_{k=1}^K$ are always distinct eigenvalues. Therefore the matrix on the left side must be full rank. Thus, $\{\lambda_{i_k}\}_{k=1}^K$ are uniquely defined by $\boldsymbol{\alpha}$. This completes the proof. ■

To recover the global average, we look for a vector $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_N]$ that makes

$$\boldsymbol{\beta}^\top \mathbf{C}_{n,K} = \mathbf{1}^\top, \tag{27}$$

which is equivalent to

$$\begin{bmatrix} u_{n1} & \cdots & u_{nN} \end{bmatrix} \left(\sum_{j=1}^{K} \beta_j \mathbf{\Lambda}^{j-1} \right) \mathbf{U}^\top = \mathbf{1}^\top. \tag{28}$$

We will show that $\boldsymbol{\beta}$ is uniquely determined by $\{\lambda_{i_k}\}_{k=1}^K$. Define a diagonal matrix $\mathbf{R}$ to be

$$\mathbf{R} = \sum_{j=1}^{K} \beta_j \mathbf{\Lambda}^{j-1}. \tag{29}$$

We move $\mathbf{U}^\top$ in (28) to the right side, and also notice that for symmetric matrices, the eigenvectors are orthogonal to each other. Moreover, $\mathbf{B}$ is a doubly stochastic matrix. The vector $\frac{1}{\sqrt{N}}\mathbf{1}$ is always an eigenvector for a doubly stochastic matrix and the corresponding eigenvalue is 1. Suppose the last column of $\mathbf{U}$ is $\mathbf{1}$. Thus we have

$$\begin{bmatrix} u_{n1} & \cdots & u_{nN} \end{bmatrix} \mathbf{R} = \begin{bmatrix} 0 & \cdots & 0 & \sqrt{N} \end{bmatrix}. \tag{30}$$

Suppose r_i is the ith diagonal entry of $\mathbf{R}$. We can ignore those r_is if $u_{ni} = 0$ for $i \in \{1, \cdots, N\}$. As a result, we can setup the following equations:

$$\sum_{j=1}^{K} \beta_j \lambda_{i_k}^{j-1} = \begin{cases} 0 & \text{if } \lambda_{i_k} \neq 1 \\ N & \text{if } \lambda_{i_k} = 1. \end{cases} \tag{31}$$

It is clear that the solution exists for the equations. The key fact is that the unknown eigenvalues, which are blotted out by zero entries in the eigenvectors, will still be blotted out in (30), and $\boldsymbol{\beta}$ is uniquely determined by $\{\lambda_i | u_{ni} \neq 0\}$; therefore, it is uniquely determined by $\boldsymbol{\alpha}$. Theorem 1 is proved.

To sum up, for regular graphs, the method can be simplified. We first find $\boldsymbol{\alpha}$ and calculate the corresponding roots λ_i. We then find $\boldsymbol{\beta}$ according to (31). Finally $[x_1, \cdots, x_K]\boldsymbol{\beta}$ would be the global average. We point out that for regular graphs, the agents need not know any information about the graphs. All they need to recover the global average is the sequence of observations $x_{n,1}, x_{n,2}, \cdots, x_{n,2K}$.

6. DISCUSSION AND CONCLUSION

In this work, we proposed an algorithm which enables an agent in a network to learn the global average through a sequence of local consensus in finite time. We proved that this algorithm always works for regular graphs.

For non-regular graphs, it may or may not work. Whether it works only depends on the structure of the graph. It would be interesting and challenging to find out the exact class of graphs for which this algorithm is guaranteed to work. To learn the global average, the agents need to know the number of total agents in the network. Besides, the agents must compare the sequence of local averages with all the graphs with N nodes to find out which graph is consistent with the sequence. The computational complexity is prohibitive for large N.

For regular graphs, it is surprising that the agents are not required to have any information. They do not even need to know the total number of agents, and as stated in Section 5, with the proposed method it is guaranteed that they can recover the global average.

7. REFERENCES

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

[2] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 1859–1864.

[3] Z. Weng and P. M. Djurić, "Distributed estimation in the presence of correlated noises," in *the Proceedings of the 21th European Signal Processing Conference (EUSIPCO)*, September 2013, pp. 2352 –2356.

[4] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[5] J. Delvenne, R. Carli, and S. Zampieri, "Optimal strategies in the average consensus problem," *Systems & Control Letters*, vol. 58, no. 10, pp. 759–765, 2009.

[6] S. Ghosh and J. Lee, "Optimal distributed consensus on unknown undirected graphs," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 2244–2249.

[7] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 4, pp. 650–660, 2008.

[8] E. Mossel and O. Tamuz, "Efficient Bayesian learning in social networks with Gaussian estimators," *arXiv preprint arXiv:1002.0747*, 2010.

[9] J. M. Hendrickx, R. M. Jungers, A. Olshevsky, and G. Vankeerberghen, "Graph diameter, eigenvalues, and minimum-time consensus," *ArXiv e-prints*, Nov. 2012.

[10] The OEIS Foundation Inc., "Number of connected graphs with n nodes. From The On-Line Encyclopedia of Integer Sequences," http://oeis.org/A001349.

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

[12] R. Bhatia, *Matrix analysis*, vol. 169, Springer, 1997.

[13] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, 1990.