Adaptive Decision-Making over Complex Networks

Sheng-Yuan Tu and Ali H. Sayed

Department of Electrical Engineering
University of California, Los Angeles, CA 90095
E-mail: {shinetu, sayed}@ee.ucla.edu

***Abstract*—It is common for biological networks to encounter situations where agents need to decide between multiple options, such as deciding between moving towards one food source or another or between moving towards a new hive or another. In previous works, we developed several powerful diffusion strategies that allow agents to estimate a model of interest in an adaptive and distributed manner through a process of in-network collaboration and learning. In this work, we consider the situation in which the data observed by the agents may arise from two different distributions or models. We develop and study a procedure by which the entire network can be made to follow one objective or the other through a distributed and collaborative decision process.**

***Index Terms*—Adaptive networks, diffusion adaptation, learning, decision-making process, biological networks.**

I. Introduction

Self-organized behavior is a remarkable property of biological networks [1], [2]. The behavior is attained in a distributed manner where agents interact with their immediate neighbors. It is a remarkable feature of biological networks that sophisticated behavior can emerge from localized interactions among agents with limited capabilities. One example of such sophisticated behavior is the group decision-making process by animals [3]. Examples include fish deciding between following one food source or another [4], and bees or ants moving towards a new hive or another [5], [6]. Even though several options may be available, the agents are able to achieve agreement and move towards a common target of interest.

Motivated by these examples, we study in this work the decision-making process over adaptive networks. These networks consist of a collection of agents with adaptation and learning abilities. The agents interact with each other on a local level to perform estimation and inference tasks in a distributed manner [7]–[9]. Adaptive networks are suitable to model collective motion in biological networks [10], [11]. They are also suitable to solve estimation and optimization problems in a distributed manner [9], [12]. In this article, we study the situation in which the data collected by the agents are influenced by one of two underlying models. At every time instant, the data arriving at any particular node could have originated from one model or the other. The objective of the network is to achieve agreement among the agents about which model to estimate as a group. A good analogy is the behavior of a fish school sensing two separate food sources. Through a process of in-network decision making, the entire fish school ends up moving towards one source in lieu of the other. We are interested in showing how adaptive networks can be designed to generally mimic this useful behavior; in this work, we focus on the case of *static* agents.

This work was supported in part by NSF grant CCF-1011918.

This objective is more challenging than earlier works on distributed estimation because each agent now needs to distinguish between which model each of its neighbors is collecting data from (this is called the *observed* model) and which model the network is evolving to (this is called the *desired* model). Therefore, in addition to the traditional learning and adaptation process, the agents should also be equipped with a decision-making process to distinguish between the observed and desired models and to help reach agreement on the desired model. The estimation and decision processes need to be implemented in a fully distributed manner and in real-time. Furthermore, the learning and decision-making processes are intertwined in that the decisions by the agents depend on their estimates and, conversely, the decisions affect the evolution of the estimates.

II. Distributed Learning and Adaptation Process

Consider a collection of N agents distributed over a spatial domain. Two agents are said to be neighbors if they can share information. The set of neighbors of agent k is called the neighborhood of k and is denoted by $\mathcal{N}_k$. At every time instant, i, each agent (or node) k is able to observe realizations $\{d_k(i), u_{k,i}\}$ of a scalar random process $\boldsymbol{d}_k(i)$ and a $1\times M$ *row* vector random process $\boldsymbol{u}_{k,i}$ with a positive-definite covariance matrix, $R_{u,k} = E\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i} > 0$. We denote random quantities by boldface letters and their realizations or deterministic quantities by normal letters. The data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ collected at node k originate from one of two unknown *column* vectors $\{w_0^\circ, w_1^\circ\}$ of size M. The data at node k are related to the observed model $z_k^\circ \in \{w_0^\circ, w_1^\circ\}$ via a linear regression model of the form [13]:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} z_k^\circ + \boldsymbol{v}_k(i) \quad (1)$$

where $\boldsymbol{v}_k(i)$ is measurement noise with variance $\sigma_{v,k}^2$ and assumed to be temporally white and spatially independent, i.e.,

$$E\boldsymbol{v}_k^*(i)\boldsymbol{v}_l(j) = \sigma_{v,k}^2 \cdot \delta_{kl} \cdot \delta_{ij} \quad (2)$$

in terms of the Kronecker delta function. The regression data $\boldsymbol{u}_{k,i}$ is likewise assumed to be temporally white and spatially independent. The noise $\boldsymbol{v}_k(i)$ is assumed to be independent of

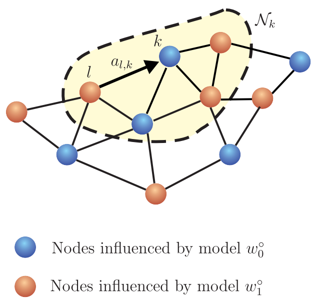

Fig. 1. A connected network where data collected by the agents are influenced by one of two models. The weight $a_{l,k}$ scales the data transmitted from node l to node k over the edge linking them.

$\boldsymbol{u}_{l,j}$ for all l and j. All random processes are assumed to be zero mean.

Although the agents are subjected to data arising from different models, the objective of the network is to have *all* agents converge to an estimate for one of the models. For example, if the model happens to represent the location of a food source [11], then this agreement will make all agents move towards one particular food source in lieu of the other. More specifically, let $w_{k,i}$ denote the estimate at node k at time i. The network would like to achieve

$$w_{k,i} \rightarrow w_q^\circ \text{ for } q=0 \text{ or } q=1 \text{ and for all } k \text{ as } i \rightarrow \infty \quad (3)$$

where convergence is in some desirable sense (such as the mean-square-error sense). To achieve agreement, it is reasonable to assume that the network topology is connected where a path always exists between any two nodes.

A. Diffusion Strategy

Several diffusion adaptation schemes for distributed estimation under a common model were proposed and studied in [7]–[9]. One such scheme is the Adapt-then-Combine (ATC) diffusion algorithm [8], which has been shown in [14] to outperform other variations of diffusion strategies as well as consensus-based strategies [15]–[18]. ATC diffusion operates as follows. We select an $N \times N$ matrix A with nonnegative entries $\{a_{l,k}\}$ satisfying:

$$A^T \mathbb{1} = \mathbb{1} \text{ and } a_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k \quad (4)$$

where $\mathbb{1}$ is the vector of size N with all entries equal to one. The entry $a_{l,k}$ denotes the weight that node k assigns to data arriving from node l (see Fig. 1). The larger the value of $a_{l,k}$ is, the higher the confidence of node k is on the information provided by node l. The ATC strategy consists of two steps and is described as follows:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu \boldsymbol{u}_{k,i}^* [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \quad (5)$$

$$\boldsymbol{w}_{k,i} = \sum_{l \in \mathcal{N}_k} a_{l,k} \boldsymbol{\psi}_{l,i} \quad (6)$$

where μ is the positive step-size. The first step (5) involves local adaptation, where node k uses its own data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ to update the weight estimate at node k from $\boldsymbol{w}_{k,i-1}$ to an intermediate value $\boldsymbol{\psi}_{k,i}$. The second step (6) is a combination step where the intermediate estimates $\{\boldsymbol{\psi}_{l,i}\}$ from the neighborhood of node k are combined through the weights $\{a_{l,k}\}$ to obtain the updated weight estimate $\boldsymbol{w}_{k,i}$.

When the data arriving at the nodes could have risen from one model or another, the diffusion strategy (5)-(6) will not be able to achieve agreement among the nodes and the resulting weight estimates will include a bias term. We first explain how this degradation arises and subsequently explain how it can be remedied.

B. Mean Convergence

Let us assume for the time being that the agents in the network have agreed on converging towards one of the models, say, w_0° or w_1°. We denote the desired model generically by w_q°. In the next section we explain how this agreement process can be attained. Let us first explain that even when agreement is present, the diffusion strategy (5)-(6) leads to biased estimates unless it is modified in a proper way. To see this, we introduce the following error vectors for any node k:

$$\tilde{\boldsymbol{w}}_{k,i} \triangleq w_q^\circ - \boldsymbol{w}_{k,i} \quad \text{and} \quad \tilde{z}_k^\circ \triangleq w_q^\circ - z_k^\circ. \quad (7)$$

Observe that these quantities measure the error relative to the desired objective, w_q°. Moreover, this desired model may or may not be the model that is influencing the data received by node k. We collect all error vectors across the network into block vectors:

$$\tilde{\boldsymbol{w}}_i \triangleq \text{col}\{\tilde{\boldsymbol{w}}_{k,i}\} \quad \text{and} \quad \tilde{z}^\circ \triangleq \text{col}\{\tilde{z}_k^\circ\} \quad (8)$$

where the notation col$\{\cdot\}$ denotes the vector that is obtained by stacking its arguments on top of each other. We also introduce the extended combination matrix:

$$\mathcal{A} \triangleq A \otimes I_M \quad (9)$$

where the symbol $\otimes$ denotes the Kronecker product of two matrices. Then, starting from (5)-(6) and using model (1), we can verify that the global error vector $\tilde{\boldsymbol{w}}_i$ evolves over time according to the recursions:

$$\tilde{\boldsymbol{w}}_i = \mathcal{A}^T(I_{NM} - \mu \boldsymbol{\mathcal{R}}_i)\tilde{\boldsymbol{w}}_{i-1} + \mu \mathcal{A}^T \boldsymbol{\mathcal{R}}_i \tilde{z}^\circ - \mu \mathcal{A}^T \boldsymbol{s}_i \quad (10)$$

where $\boldsymbol{\mathcal{R}}_i \triangleq \text{diag}\{\boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i}\}_{k=1}^N$ and $\boldsymbol{s}_i \triangleq \text{col}\{\boldsymbol{u}_{k,i}^* \boldsymbol{v}_{k,i}\}_{k=1}^N$, and where the notation diag$\{\cdot\}$ constructs a diagonal matrix from its arguments. Since the regressors $\{\boldsymbol{u}_{k,i}\}$ are temporally white and spatially independent, then the $\{\boldsymbol{u}_{k,i}\}$ are independent of $\tilde{\boldsymbol{w}}_{i-1}$. In addition, since $\boldsymbol{u}_{k,i}$ is independent of $\boldsymbol{v}_k(i)$, the vector $\boldsymbol{s}_i$ in (10) has zero mean. Taking expectation of both sides of (10), we find that the mean of $\tilde{\boldsymbol{w}}_i$ evolves over time according to the recursion:

$$\boxed{\mathbb{E}\tilde{\boldsymbol{w}}_i = \mathcal{B} \cdot \mathbb{E}\tilde{\boldsymbol{w}}_{i-1} + y} \quad (11)$$

where

$$\mathcal{B} \triangleq \mathcal{A}^T(I_{NM} - \mu \mathcal{R}) \quad (12)$$

$$\mathcal{R} \triangleq \mathbb{E}\boldsymbol{\mathcal{R}}_i = \text{diag}\{R_{u,1}, \cdots, R_{u,N}\} \quad (13)$$

$$y \triangleq \mu \mathcal{A}^T \mathcal{R} \tilde{z}^\circ. \quad (14)$$

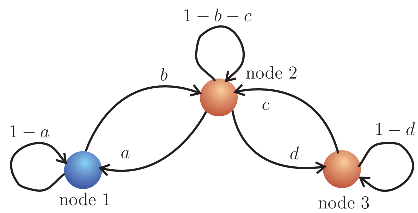

Fig. 2. A three-node network. Node 1 observes data from w_0° while nodes 2 and 3 observe data from w_1°.

The following result provides conditions to ensure the mean convergence of the network, namely, that $\mathbb{E}\tilde{\boldsymbol{w}}_i \to 0$ as $i \to \infty$.

Theorem 1. *Recursion (11) for $\mathbb{E}\tilde{\boldsymbol{w}}_i$ converges to zero if, and only if,*

$$\rho(\mathcal{B}) < 1 \text{ and } y = 0 \tag{15}$$

where $\rho(\cdot)$ denotes the spectral radius of its argument. ■

Therefore, to guarantee mean convergence, the nodes need to select the step-size μ and the combination matrix A so that conditions (15) are satisfied. It can be verified that the spectral radius of $\mathcal{B}$ is less than one as long as [14]:

$$0 < \mu < 2/\max_k \rho(R_{u,k}) \tag{16}$$

This conclusion is independent of A. However, for the second condition in (15), we note that in general, the vector y cannot be zero no matter how the nodes select the combination matrix A. When this happens, the weight estimate will be biased. Let us consider an example with three nodes in Fig. 2 where node 1 observes data from model w_0°, while nodes 2 and 3 observe data from model w_1°. The combination matrix in this case is given by

$$A^T = \begin{bmatrix} 1-a & a & 0 \\ b & 1-b-c & c \\ 0 & d & 1-d \end{bmatrix} \tag{17}$$

with the parameters $\{a, b, c, d\}$ lying in the interval $[0, 1]$ and $b+c \leq 1$. For simplicity, we assume that the vectors $\{w_0^\circ, w_1^\circ\}$ are scalars and the regressors have the same variance, i.e., $R_{u,k} = \sigma_u^2$ for all k. If the desired model of the network is $w_q^\circ = w_0^\circ$, then the vector y from (14) becomes

$$y = \mu\sigma_u^2(w_0^\circ - w_1^\circ) \begin{bmatrix} a \\ 1-b \\ 1 \end{bmatrix} \tag{18}$$

We observe that no matter how we select the parameters $\{a, b, c, d\}$, the third entry of the vector y cannot become zero. To deal with this problem, we show how to modify the diffusion strategy (5)-(6).

C. Modified Diffusion Strategy

We observe from (18) that the vector y cannot be zero because of node 3 whose neighbors observe data arising from a model that is different from the desired model. In addition, note that the bias comes from the intermediate estimates in (5). Therefore, to ensure mean convergence, a node should not combine intermediate estimates from neighbors whose observed model is different from the desired model. Instead, we replace the intermediate estimates from these neighbors by their previous estimates $\{\boldsymbol{w}_{l,i-1}\}$ in the combination step (6). Specifically, we adjust the diffusion strategy (5)-(6) in the following manner:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^* [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \tag{19}$$

$$\boldsymbol{w}_{k,i} = \sum_{l \in \mathcal{N}_k} \left(a_{l,k}^{(1)} \boldsymbol{\psi}_{l,i} + a_{l,k}^{(2)} \boldsymbol{w}_{l,i-1} \right) \tag{20}$$

where $a_{l,k}^{(1)}$ and $a_{l,k}^{(2)}$ are nonnegative entries of two matrices A_1 and A_2 that satisfy $A_1 + A_2 = A$, with A being left-stochastic. More specifically, as the discussion in the sequel will reveal, starting from the same combination matrix A used in (6), we are going to split its entries into two sets: some entries will be assigned to the matrix A_1 and the remaining entries will be assigned to the matrix A_2. The choice of which entries of A go into A_1 or A_2 will depend on which of the neighbors of node k are observing data arising from a model that agrees with the desired objective for node k. Nodes that observe data arising from the same model that node k wishes to converge to will be reinforced and their intermediate estimates $\{\psi_{l,i}\}$ will be used (their combination weights enter into A_1), while nodes that observe data arising from a different model than the objective of node k will be de-emphasized and their prior estimates $\{w_{l,i-1}\}$ will be used (their combination weights enter into A_2). Note that the first step (19) is the same as step (5). However, in the second step (20), a node is able to aggregate the $\{\boldsymbol{\psi}_{l,i}, \boldsymbol{w}_{l,i-1}\}$ from its neighborhood. With such adjustment, we will verify that by properly selecting $\{a_{l,k}^{(1)}, a_{l,k}^{(2)}\}$, mean convergence can be guaranteed for any connected network.

To begin with, the recursion for the global error vector $\tilde{\boldsymbol{w}}_i$ of the modified diffusion strategy (19)-(20) is given by:

$$\tilde{\boldsymbol{w}}_i = \left[\mathcal{A}_1^T(I_{NM} - \mu\boldsymbol{\mathcal{R}}_i) + \mathcal{A}_2^T\right]\tilde{\boldsymbol{w}}_{i-1} + \mu\mathcal{A}_1^T\boldsymbol{\mathcal{R}}_i\tilde{z}^\circ - \mu\mathcal{A}_1^T\boldsymbol{s}_i \tag{21}$$

where $\mathcal{A}_1$ and $\mathcal{A}_2$ are defined in a manner similar to $\mathcal{A}$ in (9). Taking the expectation of the both sides of (21), we get (compare with (11)):

$$\boxed{\mathbb{E}\tilde{\boldsymbol{w}}_i = \left[\mathcal{A}_1^T(I_{NM} - \mu\mathcal{R}) + \mathcal{A}_2^T\right] \cdot \mathbb{E}\tilde{\boldsymbol{w}}_{i-1} + \mu\mathcal{A}_1^T\mathcal{R}\tilde{z}^\circ} \tag{22}$$

Now we specify a way to select the combination weights $\{A_1, A_2\}$ to guarantee the convergence of $\mathbb{E}\tilde{\boldsymbol{w}}_i$ in (22) to zero. We assume the matrix $A = A_1 + A_2$ is primitive, i.e., there exists an integer power $j > 0$ such that

$$[A^j]_{l,k} > 0 \text{ for all } l \text{ and } k. \tag{23}$$

For example, it can be verified that for any connected network, if the matrix A satisfies condition (4) and $a_{k,k} > 0$ for at least one k, then condition (23) holds. Next, let us introduce two network vectors $\{f, g_i\}$ of size N each with the kth entries $\{f(k), g_i(k)\}$ indicating the indices of the observed and desired models for node k, respectively. The value of $f(k)$ is set to $f(k) = 0$ if the observed model of node k is w_0°; otherwise, $f(k) = 1$. Likewise for $g_i(k)$, whose value will be zero or one depending on whether the desired model of node k is w_0° or w_1°. Note that we assume that the observed model of a node is fixed (f is independent of i), whereas the $\{g_i(k)\}$ may change over time since the decision by each node about what the desired model should be is an evolving decision that changes over time. Since we are assuming for the time being that the nodes have achieved agreement on the desired model, which we are denoting by w_q°, we have that:

$$g_i(1) = g_i(2) = \cdots = g_i(N) = q, \quad \text{for all } i. \tag{24}$$

Then, we set the entries of A_1 and A_2 according to the following rules:

$$a_{l,k}^{(1)} = \begin{cases} a_{l,k}, & \text{if } l \in \mathcal{N}_k \text{ and } f(l) = g_i(k) \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

$$a_{l,k}^{(2)} = \begin{cases} a_{l,k}, & \text{if } l \in \mathcal{N}_k \text{ and } f(l) \neq g_i(k) \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

That is, the weights to the nodes whose observed model is the same as the desired model are collected into the matrix A_1, while the remaining weights are collected into the matrix A_2. The following result states that by selecting the combination weights according to (25)-(26), the modified diffusion strategy (19)-(20) converges in the mean.

Theorem 2. *Assume that the network is connected and A is primitive. The mean recursion in (22) converges to zero if the step size μ satisfies (16) and the matrices A_1 and A_2 are chosen according to (25)-(26).*

Proof: It suffices to show that the vector $\mu \mathcal{A}_1^T \mathcal{R} \tilde{z}^\circ$ is zero and the matrix $\mathcal{A}_1^T(I_{NM} - \mu\mathcal{R}) + \mathcal{A}_2^T$ has spectral radius strictly less than one. Without loss of generality, let w_0° be the desired model for the network (i.e., $q = 0$) and assume there are N_0 nodes with indices $\{1, 2, \ldots, N_0\}$ observing data arising from the model w_0°, while the remaining $N - N_0$ nodes observe data arising from the model w_1°. Then, we obtain from (7), (25), and (26) that

$$\tilde{z}_k^\circ = \begin{cases} 0, & \text{if } k \leq N_0 \\ w_0^\circ - w_1^\circ, & \text{if } k > N_0 \end{cases} \tag{27}$$

$$a_{l,k}^{(1)} = 0 \text{ if } l > N_0 \tag{28}$$

$$a_{l,k}^{(2)} = 0 \text{ if } l \leq N_0 \tag{29}$$

Therefore, we conclude that the vector $\mu \mathcal{A}_1^T \mathcal{R} \tilde{z}^\circ$ is zero. Moreover, in view of (28)-(29), we can write:

$$\mathcal{A}_1^T(I_{NM} - \mu\mathcal{R}) + \mathcal{A}_2^T = \mathcal{A}^T(I_{NM} - \mathcal{M}\mathcal{R}) \tag{30}$$

where $\mathcal{M}$ is an $N \times N$ block diagonal matrix with each block of size $M \times M$ and the kth diagonal block of $\mathcal{M}$, denoted by $\mathcal{M}_k$, has the form

$$\mathcal{M}_k = \begin{cases} \mu I_M, & \text{if } k \leq N_0 \\ 0, & \text{otherwise} \end{cases} \tag{31}$$

Therefore, recursion (22) is equivalent to the mean recursion of the network with N_0 informed nodes (nodes 1 to N_0) and $N - N_0$ uninformed nodes [19]. If the step-size is set to satisfy (16), then according to Theorem 1 of [19], the spectral radius of $\mathcal{A}_1^T(I_{NM} - \mu\mathcal{R}) + \mathcal{A}_2^T$ will be strictly less than one. ■

III. Distributed Decision-Making Process

The result of Theorem 2 establishes that it is possible for connected networks to converge on average to a common desired model by using (19)-(20). However, the analysis so far is based on the assumption that the nodes know what are the observed models influencing their neighbors (i.e., they need to know $f(l)$ for their neighbors); they also need to know how to update their target in $g_i(k)$. This information is then used in (25)-(26) to construct the combination weights. In this section, we describe a procedure by which this information can be estimated through local cooperation. The procedure is motivated by the process used by animal groups to reach agreement, and which is known as quorum sensing [3], [5], [6].

Assume that at time i, node k has access to the desired models of its neighborhood from the previous time instant, i.e., $\{g_{i-1}(l)\}$ for $l \in \mathcal{N}_k$. Then, one way for node k to participate in the quorum setting process is to update its target value in $g_i(k)$ according to the following rule:

$$g_i(k) = q \text{ with probability } \frac{n_k(q)^K}{\sum_r n_k(r)^K} \tag{32}$$

where $n_k(q)$ denotes the number of neighbors of node k whose desired model is w_q° and the exponent K is a positive constant ($k = 1$ in simulations). That is, node k determines its desired model in a probabilistic manner, and the probability that node k set its desired target to w_q° is proportional to the Kth power of the number of neighbors whose desired model is w_q° as well. Using such stochastic approach, we are able to verify the agreement on the desired model among the nodes.

Lemma 1. *Starting from an initial arbitrary selection of targets, $\{g_{-1}(l)\}$ for all nodes l in the network, and applying the update rule (32), then all nodes will eventually achieve agreement on the desired model, i.e., $g_\infty(1) = g_\infty(2) = \ldots = g_\infty(N)$.*

Proof: The result follows from the fact that the $\{g_i(1), g_i(2), \ldots, g_i(N)\}$ form an absorbing Markov chain with 2^N possible states and two absorbing states, namely, $g_i(1) = g_i(2) = \cdots = g_i(N) = q$ for $q = 0$ or $q = 1$. ■

However, rule (32) is still not a distributed solution for one subtle reason: nodes need to agree on which index (0 or 1) to use to refer to either model $\{w_0^\circ, w_1^\circ\}$. This task would require the nodes to share some global information. To avoid this difficulty, we shall associate with each node k two local vectors $\{f_k, g_{k,i}\}$. Each node will then assign the index one to its observed model, i.e., each node k sets $f_k(k) = 1$. Then, $f_k(l)$ and $g_{k,i-1}(l)$ are set to one if they represent the same model as the one observed by node k; otherwise, $f_k(l)$ and

$g_{k,i-1}(l)$ are set to zero. Specifically, node k needs to adjust $g_{l,i-1}(l)$ from node l according to the rule:

$$g_{k,i-1}(l) = \begin{cases} g_{l,i-1}(l), & \text{if } f_k(l) = f_k(k) \\ 1 - g_{l,i-1}(l), & \text{otherwise} \end{cases} \quad (33)$$

That is, if node l has the same observed model as node k, node k simply assigns the value of $g_{l,i-1}(l)$ to $g_{k,i-1}(l)$; otherwise, node k needs to change the value of $g_{l,i-1}(l)$ (from 0 to 1 or from 1 to 0) from node l. It can be verified that the local vectors $\{f_k, g_{k,i}\}$ are equivalent to the network vectors $\{f, g_i\}$ in the sense that for $l \in \mathcal{N}_k$, it holds that

$$f_k(k) \oplus f_k(l) = f(k) \oplus f(l) \quad (34)$$

$$f_k(k) \oplus g_{k,i}(l) = f(k) \oplus g_i(l) \quad (35)$$

where the symbol $\oplus$ denotes the exclusive-OR operation. To implement (33), node k still needs to determine f_k, i.e., the ability to differentiate between observed models. We propose a procedure to determine f_k using the available data $\{\boldsymbol{w}_{l,i-1}, \boldsymbol{\psi}_{l,i}\}$.

A. Information Classification

To determine the vector f_k, we introduce another vector $b_{k,i}$, whose lth entry, $b_{k,i}(l)$, will be a measure of the belief by node k that node l has the same observed model. The value of $b_{k,i}(l)$ lies in the range $[0, 1]$. The higher the value of $b_{k,i}(l)$ is, the more confidence node k has that node l is subject to the same model as its own model. In this construction, the vector $b_{k,i}$ changes over time according to the data $\{\boldsymbol{w}_{l,i-1}, \boldsymbol{\psi}_{l,i}\}$. Node k adjusts $\{b_{k,i}(l)\}$ according to the rule:

$$b_{k,i}(l) = \begin{cases} (1-\alpha) \cdot b_{k,i-1}(l) + \alpha, & \text{if belief is increased} \\ (1-\alpha) \cdot b_{k,i-1}(l), & \text{if belief is decreased} \end{cases} \quad (36)$$

for some positive step-size $\alpha \in [0, 1]$. That is, node k increases the belief by linearly combining the belief from the previous time instant with one. In contrast, node k linearly combines $b_{k,i-1}(l)$ and zero to decrease the belief. Node k then set $f_k(l)$ according to the rule:

$$f_k(l) = \begin{cases} 1, & \text{if } b_{k,i}(l) \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (37)$$

To update $b_{k,i}(l)$, we use model (1) and obtain from the adaptation step (19) that

$$\boldsymbol{\psi}_{l,i} - \boldsymbol{w}_{l,i-1} = \mu \boldsymbol{u}_{l,i}^* \boldsymbol{u}_{l,i}(z_l^\circ - \boldsymbol{w}_{l,i-1}) + \mu \boldsymbol{u}_{l,i}^* \boldsymbol{v}_l(i) \quad (38)$$

Taking expectation of the both sides, we have that

$$\mathbb{E}[\boldsymbol{\psi}_{l,i} - \boldsymbol{w}_{l,i-1}] = \mu R_{u,l} \mathbb{E}[z_l^\circ - \boldsymbol{w}_{l,i-1}] \quad (39)$$

That is, on average, the adaptation term is a scaled vector pointing from $\boldsymbol{w}_{l,i-1}$ towards z_l° with scaling matrix $\mu R_{u,l}$. Note that since $R_{u,l}$ is a positive-definite matrix, the adaptation term lies in the same half plane of the vector $\mathbb{E}[z_l^\circ - \boldsymbol{w}_{l,i-1}]$. Therefore, the term $\mathbb{E}[\boldsymbol{\psi}_{l,i} - \boldsymbol{w}_{l,i-1}]$ provides useful information about the observed model at node l. In addition, this term also tells us how close the estimate at node l is to its observed model. We know that if the magnitude of $\mathbb{E}[\boldsymbol{\psi}_{l,i} - \boldsymbol{w}_{l,i-1}]$ is large, or the estimate at node l is far from z_l°, then node l is in the transient state. On the contrary, if the magnitude is small, then the estimate $\boldsymbol{w}_{l,i-1}$ at node l is close to z_l° and the node is operating close to steady-state. The expected value of $(\boldsymbol{\psi}_{l,i} - \boldsymbol{w}_{l,i-1})$ can be estimated by the first-order recursion:

$$\phi_{l,i} = (1-\nu) \cdot \phi_{l,i-1} + \nu \cdot (\psi_{l,i} - w_{l,i-1}) \quad (40)$$

where ν is a positive step-size smaller than one. Note that recursion (40) is able to track the variation of $\mathbb{E}[\boldsymbol{\psi}_{l,i} - \boldsymbol{w}_{l,i-1}]$ over time. There are four scenarios to consider depending on whether nodes k or l are in the transient phase or in steady-state operation.

We first assume that both nodes k and l are in transient state, i.e., $\|\phi_{k,i}\| > \eta_1$ and $\|\phi_{l,i}\| > \eta_1$ for some threshold η_1. Node k will increase the belief value $b_{k,i}(l)$ using (36) if

$$\phi_{k,i}^T \phi_{l,i} > 0 \quad (41)$$

Otherwise, node k decreases the belief $b_{k,i}(l)$. That is, if both nodes k and l are in transient state, then node k increases its belief of node l when their adaptation terms $\{\phi_{k,i}, \phi_{l,i}\}$ lie in the same half plan. Next we assume that both nodes k and l are in steady-state, i.e., $\|\phi_{k,i}\| \leq \eta_1$ and $\|\phi_{l,i}\| \leq \eta_1$. In this case, node k increases the belief $b_{k,i}(l)$ if

$$\|w_{k,i-1} - w_{l,i-1}\| \leq \eta_2 \quad (42)$$

for some threshold η_2. Otherwise, node k decreases the belief $b_{k,i}(l)$. That is, when both nodes k and l converge to their observed models (namely, $w_{k,i-1} \approx z_k^\circ$ and $w_{l,i-1} \approx z_l^\circ$), node k increases the belief to node l if they converge to the same model. Finally, when one of nodes k or l is in transient state and the other is in steady-state, node k decreases the belief $b_{k,i}(l)$ if one of the following conditions holds:

$$\begin{cases} \|\phi_{k,i}\| \leq \eta_1 \text{ and } \phi_{l,i}^T(w_{k,i-1} - w_{l,i-1}) \leq 0 \\ \|\phi_{l,i}\| \leq \eta_1 \text{ and } \phi_{k,i}^T(w_{l,i-1} - w_{k,i-1}) \leq 0 \end{cases} \quad (43)$$

That is, when node k is in steady-state and node l is in transient state, then we have $w_{k,i-1} \approx z_k^\circ$ and node k decreases the belief to node l if the adaptation term of node l points towards the opposite direction of the observed model of node k. Similar explanation applies to the other case.

We now explain how to determine the thresholds η_1 and η_2. First, it is reasonable to assume that two model vectors are sufficiently away from each other, i.e., $\|w_0^\circ - w_1^\circ\| \gg 1$. In addition, in steady-state, the error vector $(z_k^\circ - w_{k,i})$ has magnitude much less than 1. To determine η_1, we observe from (39) that the magnitude of $\phi_{l,i}$ is of the order of μ, i.e., $\phi_{l,i} = c \cdot \mu$ and that the term c is much smaller than 1 if node l is in steady-state. Hence, we set η_1 to $\eta_1 = \mu$. In terms of η_2, by the triangular inequality, we have that

$$\begin{aligned} &\|w_{k,i-1} - w_{l,i-1}\| \\ &\leq \|z_k^\circ - w_{k,i-1}\| + \|z_l^\circ - w_{l,i-1}\| + \|z_k^\circ - z_l^\circ\| \end{aligned} \quad (44)$$

If both nodes k and l are in steady-state, then the first two terms on the right-hand side of (44) are much less than one.

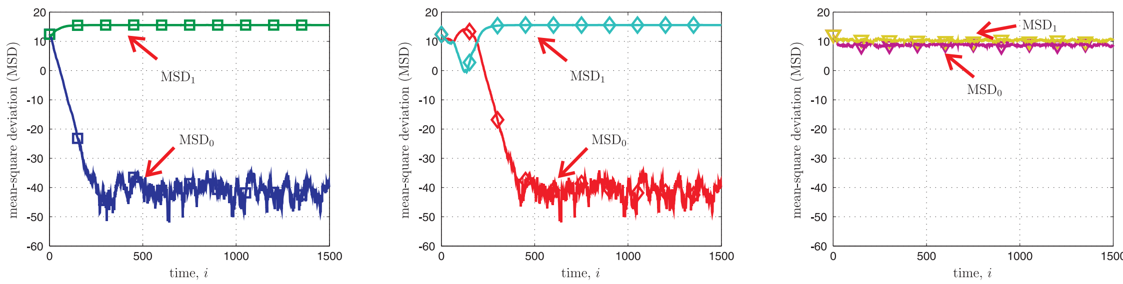

Fig. 3. Transient network MSD over a network with 40 nodes in the case with perfect information (left), in the case with imperfect information (middle), and in the case with the conventional diffusion strategy (5)-(6) (right).

Therefore, the term on the left-hand side of (44) is much less than one if $z_k^\circ = z_l^\circ$; otherwise, it is much greater than one. Hence, we set $\eta_2 = 1$.

IV. Simulation Results

We consider a connected networks with 40 static nodes randomly connected. The model vectors are set to $w_0^\circ = [3;3]$ and $w_1^\circ = [3;-3]$ ($M = 2$). Assume that the first 20 nodes (nodes 1 through 20) observe data arising from the model w_0°, while the remaining nodes observe data originating from the model w_1°. The step-sizes are set to $\mu = 0.05$, $\nu = 0.2$, and $\alpha = 0.05$. The network employs the uniform combination rule: $a_{l,k} = 1/n_k$ if $l \in \mathcal{N}_k$, where n_k denotes the number of neighbors of node k.

In Fig. 3, we illustrate the network mean-square deviation (MSD) with respect to the two model vectors over time, i.e.,

$$\mathrm{MSD}_q(i) = \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|w_q^\circ - \boldsymbol{w}_{k,i}\|^2 \tag{45}$$

for $q = 0$ and $q = 1$. We consider three situations: the case with known vectors $\{f, g_i\}$ (perfect information), the case with estimated vectors $\{f_k, g_{k,i}\}$ (imperfect information), and the case with the conventional diffusion strategy (5)-(6). We observe that our proposed algorithm successfully converges to one of the models, although it takes about 200 iterations to achieve agreement. Moreover, the algorithm with imperfect information converges to the perfect case. However, for the conventional diffusion strategy, the nodes converge to a common vector that does not coincide with either of the model vectors.

References

[1] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-Organization in Biological Systems*. Princeton University Press, 2003.

[2] I. D. Couzin, "Collective cognition in animal groups," *Trends in Cognitive Sciences*, vol. 13, pp. 36–43, Jan. 2009.

[3] D. J. T. Sumpter and S. C. Pratt, "Quorum responses and consensus decision making," *Phil. Trans. R. Soc. B*, vol. 364, pp. 743–753, Dec. 2009.

[4] I. D. Couzin, C. C. Ioannou, G. Demirel, T. Gross, C. J. Torney, A. Hartnett, L. Conradt, S. A. Levin, and N. E. Leonard, "Uninformed individuals promote democratic consensus in animal groups," *Science*, vol. 334, pp. 1578–1580, Dec. 2011.

[5] N. F. Britton, N. R. Franks, S. C. Pratt, and T. D. Seeley, "Deciding on a new home: How do honeybees agree?" *Proc. R. Soc. Lond. B*, vol. 269, pp. 1383–1388, May 2002.

[6] S. C. Pratt, E. B. Mallon, D. J. T. Sumpter, and N. R. Franks, "Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant Leptothorax albipennis," *Behav. Ecol. Sociobiol.*, vol. 52, pp. 117–127, May 2002.

[7] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[8] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[9] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, Aug 2012.

[10] F. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Trans. on Signal Processing*, vol. 59, no. 5, pp. 2038–2051, May 2011.

[11] S. Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Selected Topics on Signal Processing*, vol. 5, no. 4, pp. 649–664, Aug. 2011.

[12] A. H. Sayed, "Diffusion adaptation over networks," to appear in *E-Reference Signal Processing,* R. Chellapa and S. Theodoridis, editors, Elsevier, 2013. Also available online at http://arxiv.org/abs/1205.4220, May 2012.

[13] ——, *Adaptive Filters*. NJ. Wiley, 2008.

[14] S. Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," to appear in *IEEE Trans. on Signal Processing*, Dec. 2012. Also available online at http://arxiv.org/abs/1205.3993v2, Aug. 2012.

[15] J. N. Tsitsiklis, J. N. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. on Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[16] A. Nedic and A. Ozdaglar, "Cooperative distributed multi-agent optimization," in the book *Convex Optimization in Signal Processing and Communications,* Y. Eldar and D. Palomar *(Eds.),* Cambridge University Press, pp. 340–386, 2009.

[17] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[18] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE J. Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 674–690, Aug. 2011.

[19] S. Y. Tu and A. H. Sayed, "On the influence of informed agents on learning and adaptation over networks," to appear in *IEEE Trans. on Signal Processing*, 2013. Also available online at http://arxiv.org/abs/1203.1524, Mar. 2012.