On the Effects of Topology and Node Distribution on Learning over Complex Adaptive 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*—Adaptive networks consist of a collection of nodes with adaptation and learning abilities. The nodes interact with each other on a local level and diffuse information across the network through their collaborations, as dictated by the network topology and by the spatial distribution of the nodes. In this work, we consider two types of nodes: informed and uninformed. The former collect data and perform processing, while the latter only participate in the processing tasks. We examine the performance of adaptive networks as a function of the fraction of informed nodes. The results reveal an interesting trade-off between convergence and performance. The analysis indicates that the larger the proportion of informed nodes in a network, the faster the convergence rate is at the expense of a deterioration in the mean-square-error performance. The conclusion suggests an important interplay relating the number of informed nodes, the desired convergence rate, and the desired estimation accuracy.**

***Index Terms*—Adaptive networks, diffusion adaptation, learning, topology, Erdos-Renyi network, scale-free network, power law, small world phenomenon, informed nodes.**

I. Introduction

Adaptive networks consist of a collection of spatially distributed nodes that are linked together through a connection topology and that cooperate with each other through local interactions. Adaptive networks are well-suited to perform decentralized information processing and inference tasks [1], [2] and to model complex and self-organized behavior encountered in biological systems [3], such as fish joining together in schools [4], birds flying in formation [5], and bees swarming towards a new hive [6].

In the previous works [1], [2], [4], the nodes in the network were assumed to be homogeneous in that all nodes had similar capabilities and were able to have continuous access to measurements. However, it is often observed in biological networks that the behavior of the network tends to be dictated more heavily by a small fraction of the agents, as happens with bees [7] and fish [8]. This observation motivates us to study what we shall refer to as *heterogeneous* adaptive networks, where a fraction of the nodes are assumed to be informed while the remaining nodes are assumed to be uninformed. Informed nodes collect data and perform in-network processing, while uninformed nodes only participate in the processing tasks.

This work was supported in part by NSF grants CCF-1011918 and CCF-0942936.

We shall examine how the transient and steady-state behavior of the network is dependent on its topology and on the distribution of informed nodes. The results reveal an interesting trade-off between convergence and performance. In particular, the analysis shows that the larger the proportion of informed nodes in a network, the faster the convergence rate is at the expense of a deterioration in the mean-square-error performance. We apply the results to two types of topology models widely used in the complex networks literature [9]: Erdos-Renyi models and scale-free models.

II. Diffusion Adaptation Algorithm

Consider a collection of N nodes distributed over a spatial domain. Two nodes are said to be neighbors if they can share information. The set of neighbors of node k, including k itself, is called the neighborhood of k and is denoted by $\mathcal{N}_k$. The nodes would like to estimate an unknown column vector, w°, of size M. At every time instant, i, each node k is able to observe realizations $\{d_k(i), u_{k,i}\}$ of a scalar random process $\boldsymbol{d}_k(i)$ and a $1\times M$ vector random process $\boldsymbol{u}_{k,i}$ with a positive-definite covariance matrix, $R_{u,k} = E\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}$. All vectors in our treatment are column vectors with the exception of the regression vector, $\boldsymbol{u}_{k,i}$. We also denote random quantities by boldface letters. The random processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ are assumed to be related to w° via a linear regression model of the form [10]:

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

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

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

in terms of the Kronecker delta function. The noise $\boldsymbol{v}_k(i)$ is also assumed to be independent of $\boldsymbol{u}_{l,j}$ for all l and j. All random processes are assumed to be zero mean.

The objective of the network is to estimate w° in a distributed manner and in real-time through an online learning process, where each node is allowed to interact only with its neighbors. The nodes estimate w° by seeking to minimize the following global cost function:

$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} E|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2 \tag{3}$$

Several diffusion adaptation schemes for solving (3) in a distributed manner were proposed in [1], [2]. One such scheme is the Adapt-then-Combine (ATC) diffusion algorithm [2]. It operates as follows. We assign an $N \times N$ matrix A with nonnegative entries $\{a_{l,k}\}$ satisfying:

$$A^T\mathbf{1} = \mathbf{1} \text{ and } a_{l,k} = 0 \text{ if, and only if, } l \notin \mathcal{N}_k \tag{4}$$

where $\mathbf{1}$ is a vector of size N with all entries equal to one. The entry $a_{l,k}$ denotes the weight on the link connecting node l to node k. The ATC algorithm consists of two steps. The first step (5a) involves local adaptation, where node k uses its own data $\{d_k(i), u_{k,i}\}$. This step updates the weight estimate at node k from $w_{k,i-1}$ to an intermediate value $\psi_{k,i}$. The second step (5b) is a combination step where the intermediate estimates $\{\psi_{l,i}\}$ from the neighborhood are combined through the coefficients $\{a_{l,k}\}$ to obtain the updated weight estimate $w_{k,i}$. The algorithm is described as follows:

$$\begin{cases} \psi_{k,i} = w_{k,i-1} + \mu_k u_{k,i}^*[d_k(i) - u_{k,i}w_{k,i-1}] & \text{(5a)} \\ w_{k,i} = \sum_{l\in\mathcal{N}_k} a_{l,k}\psi_{l,i} & \text{(5b)} \end{cases}$$

where μ_k is the positive step-size used by node k. To model heterogeneity over the network, we set $\mu_k = 0$ if node k is uninformed. In this model, uninformed nodes do not perform the adaptation step (5a) but continue to perform (5b).

III. Performance Analysis

The mean-square performance of the ATC algorithm was studied in detail in [2] by applying the energy conservation approach of [10], [11].

A. Mean Stability

Let the error vector for any node k be $\tilde{\boldsymbol{w}}_{k,i} = w^\circ - \boldsymbol{w}_{k,i}$. We collect all weight error vectors and step-sizes across the network into a block vector and block matrix: $\tilde{\boldsymbol{w}}_i = \text{col}\{\tilde{\boldsymbol{w}}_{k,i}\}$ and $\mathcal{M} = \text{diag}\{\mu_k I_M\}$, and introduce the extended combination matrix $\mathcal{A} = A \otimes I_M$ where the symbol $\otimes$ denotes the Kronecker product operation of two matrices. Then, starting from (5a)-(5b) and using model (1), we can verify that the global error vector evolves according to the relation:

$$\boxed{\tilde{\boldsymbol{w}}_i = \mathcal{A}^T(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}^T\mathcal{M}\boldsymbol{g}_i} \tag{6}$$

where $\boldsymbol{\mathcal{R}}_i = \text{diag}\{\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}\}$ and $\boldsymbol{g}_i = \text{col}\{\boldsymbol{u}_{k,i}^*\boldsymbol{v}_{k,i}\}$. Suppose that the regressors $\{\boldsymbol{u}_{k,i}\}$ are spatially and temporally independent. This assumption implies that $\boldsymbol{u}_{k,i}$ is independent of $\tilde{\boldsymbol{w}}_{i-1}$. Taking expectation of both sides of (6), we find that the mean relation of $\tilde{\boldsymbol{w}}_i$ evolves in time according to the recursion:

$$\boxed{E\tilde{\boldsymbol{w}}_i = \mathcal{X} \cdot E\tilde{\boldsymbol{w}}_{i-1}} \tag{7}$$

where $\mathcal{X} = \mathcal{A}^T(I - \mathcal{M}\mathcal{R})$ with $\mathcal{R} = E\boldsymbol{\mathcal{R}}_i = \text{diag}\{R_{u,k}\}$.

In the following, we give conditions to ensure mean stability, i.e., $E\tilde{\boldsymbol{w}}_i \to 0$ as $i \to \infty$, even in the presence of uninformed nodes.

Theorem 1 (Mean stability)**.** *For heterogeneous networks, the ATC algorithm converges in the mean if the step-sizes $\{\mu_k\}$ and the combination matrix A satisfy the following conditions:*

1) *For every informed node l, the step-size μ_l satisfies:*

$$0 < \mu_l \cdot \rho(R_{u,l}) < 2 \tag{8}$$

where $\rho(\cdot)$ denotes the spectral radius.

2) *For every node k, there exists an informed node l and a finite integer j such that*

$$\left[A^j\right]_{l,k} > 0 \tag{9}$$

That is, the (l,k)th entry of A^j is positive.

Result (9) states that as long as there exists a path from informed nodes to every other node, the mean stability holds. Usually, we are interested in connected networks where a path always exists between any two arbitrary nodes. This in turn implies that there exists a finite integer j such that all entries in the matrix $(A^T)^j$ are positive [12] so that condition (9) is automatically satisfied. In this case, the ATC algorithm converges in the mean if there exists at least one informed node with its step-size satisfying condition (8). In the following, we show that conditions (8)-(9) also guarantee mean-square convergence when the step-size is sufficiently small.

B. Mean-Square Performance

The network mean-square-deviation (MSD) is used to assess how well the network estimates the weight vector, w°. The MSD is defined as follows:

$$\text{MSD} \triangleq \lim_{i\to\infty} \frac{1}{N}\sum_{k=1}^{N} E\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \tag{10}$$

Let Σ denote a nonnegative-definite Hermitian matrix that we are free to choose and let $\sigma = \text{vec}(\Sigma)$ denote the vector that is obtained by stacking the columns of Σ on top of each other. We shall interchangeably use the notation $\|x\|_\sigma^2$ and $\|x\|_\Sigma^2$ to denote the same weighted norm of a vector x. Following the energy conservation approach of [10], we can derive the following weighted variance relation:

$$\begin{aligned} E\|\tilde{\boldsymbol{w}}_i\|_\Sigma^2 =& E\left(\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{(I-\boldsymbol{\mathcal{R}}_i\mathcal{M})\mathcal{A}\Sigma\mathcal{A}^T(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)}\right) \\ &+ \text{Tr}(\Sigma\mathcal{A}^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}) \end{aligned} \tag{11}$$

where $\mathcal{G} = E\boldsymbol{g}_i\boldsymbol{g}_i^* = \text{diag}\{\sigma_{v,k}^2 R_{u,k}\}$. Some algebra (see [2]) shows that under a sufficiently small step-size assumption, expression (11) can be approximated and rewritten as:

$$\boxed{E\|\tilde{\boldsymbol{w}}_i\|_\sigma^2 = E\|\tilde{\boldsymbol{w}}_{i-1}\|_{F\sigma}^2 + [\text{vec}(\mathcal{Y}^T)]^T\sigma} \tag{12}$$

where $F \approx \mathcal{X}^T \otimes \mathcal{X}^*$ and $\mathcal{Y} = \mathcal{A}^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}$. The following result ensures that $E\|\tilde{\boldsymbol{w}}_i\|_\sigma^2$ remains bounded and converges as i goes to infinity.

Theorem 2 (Mean-square stability)**.** *For sufficiently small step-sizes, the ATC algorithm is mean-square stable if the step-sizes $\{\mu_k\}$ and the matrix A satisfy conditions (8)-(9).*

Now, let the time index i tend to infinity. From (12), we obtain the steady-state relation $E\|\tilde{\boldsymbol{w}}_\infty\|^2_{(I-F)\sigma} = [\text{vec}(\mathcal{Y}^T)]^T\sigma$. Since the eigenvalues of the matrix F are within the unit disc, the matrix $(I-F)$ is invertible. Thus, the network MSD can be obtained by choosing $\sigma = (I-F)^{-1}\,\text{vec}(I_{NM})/N$, which gives

$$\boxed{\text{MSD} = \frac{1}{N}[\text{vec}(\mathcal{Y}^T)]^T\,(I-F)^{-1}\,\text{vec}(I_{NM})} \tag{13}$$

IV. Network Topology Models

In this section, we consider two popular models used in the study of complex networks. In both models, we let n_k denote the degree (number of neighbors) of node k. Note that since node k is a neighbor of itself, we have $n_k \geq 1$. In addition, we assume the topology is symmetric. That is, if node l is a neighbor of node k, then node k is also a neighbor of node l.

A. Erdos-Renyi Networks

The Erdos-Renyi model [13] is widely used to model network behavior over complex networks. In the model, there is a single parameter, called *edge probability*, and denoted by $p \in [0,1]$. The edge probability specifies the probability that two distinct nodes are connected. In this way, the degree distribution of any node k becomes a binomial distribution:

$$f(n_k) = \binom{N-1}{n_k-1} p^{n_k-1}(1-p)^{N-n_k} \tag{14}$$

and the expected degree, $\bar{n}_k$, for node k is

$$\bar{n}_k = (N-1)p + 1 \tag{15}$$

By adjusting the parameter p, we are able to control the distribution of the degrees of the nodes.

B. Scale-Free Networks

The Erdos-Renyi model does not capture several prominent features of real networks such as the *small world phenomenon* and the *power-law degree* phenomenon [9]. The small world phenomenon refers to the fact that the number of edges between two arbitrary nodes is small on average. The power-law degree effect refers to the fact that the number of nodes with degree n_k falls off as an inverse power of n_k, namely,

$$f(n_k) \sim c n_k^{-\gamma} \tag{16}$$

with two positive constants c and γ. Networks with degree distributions of the form (16) are called scale-free networks [14] and can be generated using preferential attachment models. We briefly describe the model proposed by [15]. The model starts with a small network with N_0 nodes. We assume the network is connected. At every iteration, we add a new node, which will connect to $m \leq N_0$ distinct nodes besides itself. The probability of connecting to node k is proportional to the degree of node k. As time evolves, nodes with higher degree are more likely to be connected to new nodes. Eventually, there are a few nodes that connect to most of the network. This phenomenon is observed in real networks. If $N \gg N_0$, the expected degree of the network, $\bar{\eta}$, is

$$\bar{\eta} \triangleq \frac{1}{N}\sum_{k=1}^{N}\bar{n}_k \approx 2m+1 \tag{17}$$

because every new arrival node contributes $2m+1$ degree.

V. Effect of Topology and Node Distribution

From (12), we note that the convergence behavior of the network depends on the eigenstructure of the matrix F. This matrix reflects two kinds of influences: the effect of the network topology through its dependence on $\mathcal{A}$, and the effect of the spatial distribution of informed nodes through $\mathcal{M}$. In this section, we use the theoretical expressions to examine the effect of the topology and node distribution on the convergence rate and mean-square performance of the network.

A. Convergence Rate

The convergence rate relates to the transient behavior of the algorithm and shows how fast the network converges to steady-state. We define the convergence factor, f, as the rate at which $E\|\tilde{\boldsymbol{w}}_i\|^2$ decays during the transient phase, so that the smaller the value of f is, the faster the rate of decay of $E\|\tilde{\boldsymbol{w}}_i\|^2$ is. As shown in [2], [10] and as indicated by (12), the convergence factor is determined by the spectral radius of the matrix F in (12), i.e.,

$$f = \rho(F) = [\rho(\mathcal{X})]^2 \tag{18}$$

Let us assume from now on that $\mu_k = \mu$ for all informed nodes and that $R_{u,k} = R_u$ for all k. In addition, we assume that the step-size is small enough such that $\mu \cdot \rho(R_u) < 1$. Let $\mathcal{N}_I$ denote the set of informed nodes. Then, we have the following result.

Lemma 1 (Monotonicity)**.** *Consider two configurations of the network: one with $\mathcal{N}_{I,1}$ informed nodes and another with $\mathcal{N}_{I,2}$ informed nodes. If $\mathcal{N}_{I,1} \subseteq \mathcal{N}_{I,2}$, then $f_1 \geq f_2$. In other words, configurations with a larger proportion of informed nodes, converge faster.*

The following result provides bounds for the convergence factor.

Lemma 2 (Bounds)**.** *The convergence factor is bounded by*

$$[1-\mu\cdot\lambda_M(R_u)]^2 \leq f < 1 \tag{19}$$

where $\lambda_M(R_u)$ denotes the smallest eigenvalue of R_u.

B. Network MSD

Using the equalities for arbitrary matrices $\{U, W, \Sigma\}$: $\text{vec}(U\Sigma W) = (W^T \otimes U)\sigma$ and $\text{Tr}(\Sigma W) = \text{vec}(W^T)^T\sigma$, we can obtain an alternative expression for the network MSD from (13):

$$\text{MSD} = \frac{1}{N}\sum_{j=0}^{\infty}\text{Tr}[\mathcal{X}^j\mathcal{Y}(\mathcal{X}^*)^j] \tag{20}$$

We observe that the MSD depends on $\mathcal{X}$ in a nontrivial manner. In an effort to gain insight into the behavior of the

network, we further assume that the nodes use the uniform combination rule in step (5b), i.e.,

$$a_{l,k} = \begin{cases} 1/n_k, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

It can be verified that the matrix A defined by (21) is diagonalizable and has real eigenvalues. Let r_k and s_k^* denote an arbitrary right and left eigenvector pair for A^T corresponding to eigenvalue $\lambda_k(A)$. Without loss of generality, we order the eigenvalues of A^T in decreasing order, i.e., $|\lambda_1(A)| \geq |\lambda_2(A)| \geq \cdots \geq |\lambda_N(A)|$. We also normalize r_k to satisfy $\|r_k\|^2 = 1$. Then, the eigen-decomposition of A can be written as:

$$\boxed{A^T = \sum_{k=1}^{N} \lambda_k(A) r_k s_k^*} \tag{22}$$

Note that $r_k^* s_l = \delta_{kl}$. For Erdos-Renyi networks, since, on average, all nodes have the same degree value, then the corresponding matrix A^T is approximately symmetric. As such, its eigenvectors $\{r_k, s_k\}$ are approximately orthonormal, i.e., $r_k^* r_l \approx \delta_{kl}$ and $s_k^* s_l \approx \delta_{kl}$. Even though this approximation is not generally valid for scale-free networks, simulations indicate that the approximation leads to good match between theory and practice. Since R_u is assumed to be positive-definite, we let z_m ($m = 1, \ldots, M$) be an eigenvector of R_u associated with the eigenvalue $\lambda_m(R_u)$. Then, the eigen-decomposition of R_u is given by:

$$\boxed{R_u = \sum_{m=1}^{M} \lambda_m(R_u) z_m z_m^*} \tag{23}$$

where the $\{z_m\}$ have unit magnitude, i.e., $\|z_m\|^2 = 1$, and $\{\lambda_m(R_u)\}$ are positive. In the sequel, for any vector x, we shall use the notation $x_{k:l}$ to denote a sub-vector of x formed from the kth up to the lth entries of x. Let N_I denote the number of informed nodes in the network. Without loss of generality, we assume that $\{1, 2, \ldots, N_I\} \in \mathcal{N}_I$.

Lemma 3 (Eigenstructure of $\mathcal{X}$)**.** *The matrix $\mathcal{X} = \mathcal{A}^T(I - \mathcal{M}\mathcal{R})$ has approximate right and left eigenvector pairs $\{r_{k,m}^x, s_{k,m}^x\}$ of the form:*

$$r_{k,m}^x \approx r_k \otimes z_m \tag{24}$$

$$s_{k,m}^x \approx \frac{\lambda_k(A)}{\lambda_{k,m}^*(\mathcal{X})} \begin{bmatrix} [1 - \mu\lambda_m(R_u)]\, s_{k,1:N_I} \otimes z_m \\ s_{k,N_I+1:N} \otimes z_m \end{bmatrix} \tag{25}$$

where $\lambda_{k,m}(\mathcal{X})$ denotes the eigenvalue of the eigenvector pair $\{r_{k,m}^x, s_{k,m}^x\}$ and has the approximate value

$$\lambda_{k,m}(\mathcal{X}) \approx [1 - s_{k,1:N_I}^* r_{k,1:N_I} \mu\lambda_m(R_u)]\lambda_k(A) \tag{26}$$

Therefore, the eigen-decomposition of the matrix $\mathcal{X}^j$ in (20) has the approximate form:

$$\boxed{\mathcal{X}^j \approx \sum_{k=1}^{N} \sum_{m=1}^{M} \lambda_{k,m}^j(\mathcal{X}) r_{k,m}^x s_{k,m}^{x*}} \tag{27}$$

In this way, the network MSD (24) simplifies to:

$$\text{MSD} \approx \sum_{k=1}^{N} \sum_{m=1}^{M} \frac{s_{k,m}^{x*} \mathcal{Y} s_{k,m}^x}{N[1 - |\lambda_{k,m}(\mathcal{X})|^2]} \tag{28}$$

where we used (24) to get $r_{l,n}^{x*} r_{k,m}^x \approx \delta_{kl} \cdot \delta_{mn}$. Next, we focus on the term $s_{k,m}^{x*} \mathcal{Y} s_{k,m}^x$ in (28). We rewrite the matrix $\mathcal{Y}$ from (12) as:

$$\mathcal{Y} = \mathcal{Z}\Omega^{-1}\mathcal{Z}^* \tag{29}$$

where $\mathcal{Z} = \mathcal{A}^T\mathcal{M}\mathcal{R} = \mathcal{A}^T - \mathcal{X}$ and $\Omega = \text{diag}\{\sigma_{v,k}^{-2} R_u\}$. Then, some algebra shows that

$$s_{k,m}^{x*} \mathcal{Y} s_{k,m}^x \approx \mu^2 \lambda_m(R_u) \lambda_k^2(A) \sum_{l=1}^{N_I} \sigma_{v,l}^2 |s_{k,l}|^2 \tag{30}$$

Substituting (26) and (30) into (28), the MSD expression becomes

$$\boxed{\text{MSD} \approx \sum_{k=1}^{N} \sum_{m=1}^{M} \frac{\mu^2 \lambda_m(R_u) \lambda_k^2(A) \sum_{l=1}^{N_I} \sigma_{v,l}^2 |s_{k,l}|^2}{N[1 - \lambda_k^2(A)|1 - s_{k,1:N_I}^* r_{k,1:N_I} \mu\lambda_m(R_u)|^2]}} \tag{31}$$

Expression (31) only depends on the eigenvalues of A and R_u and on the eigenvectors of A. In the following, we use properties of the eigenvalues of A to provide insights into the behavior of the network MSD.

In Fig. 1, we show the averaged distribution of $|\lambda_k(A)|$ for the Erdos-Renyi and scale-free models and observe that there is an eigenvalue (i.e., $\lambda_1(A)$) greater than the remaining eigenvalues, especially for highly connected networks (i.e., high values of p or m). We then decompose the MSD in (31) into two parts. The first part is determined by $\lambda_1(A)$, i.e., $k = 1$ in (31), and is denoted by $\text{MSD}_{k=1}$. The second part is contributed by the remaining eigenvalues of A, i.e., $k > 1$ in (31), and is denoted by $\text{MSD}_{k>1}$. Since $\lambda_1(A) = 1$, the term $\text{MSD}_{k=1}$ becomes

$$\text{MSD}_{k=1} \approx \frac{M\mu}{2N} \cdot \frac{\sum_{l=1}^{N_I} \sigma_{v,l}^2 |s_{k,l}|^2}{\text{Re}(s_{1,1:N_I}^* r_{1,1:N_I})} \tag{32}$$

where the notation $\text{Re}(\cdot)$ denotes the real part of its argument. From condition (4), we know that the vector $\mathbf{1}$ is a right eigenvector of A^T corresponding to the eigenvalue $\lambda_1(A) = 1$. To satisfy the normalization condition that $\|r_k\| = 1$, the vector r_1 is taken as $r_1 = \mathbf{1}/\sqrt{N}$. In addition, for the uniform combination weights (21), it can be verified that s_1 has the following form:

$$s_1 = \frac{\sqrt{N}}{\sum_{l=1}^{N} n_l} \text{col}\{n_1, \ldots, n_N\} \tag{33}$$

Therefore, expression (32) becomes

$$\boxed{\text{MSD}_{k=1} \approx \frac{M\mu}{2N} \cdot \frac{\sum_{l=1}^{N_I} \sigma_{v,l}^2 n_l^2}{\eta \sum_{l=1}^{N_I} n_l}} \tag{34}$$

Expression (34) reveals several interesting properties. First, we observe that the term $\text{MSD}_{k=1}$ does not depend on the matrix R_u, which is a property for the MSD expression of

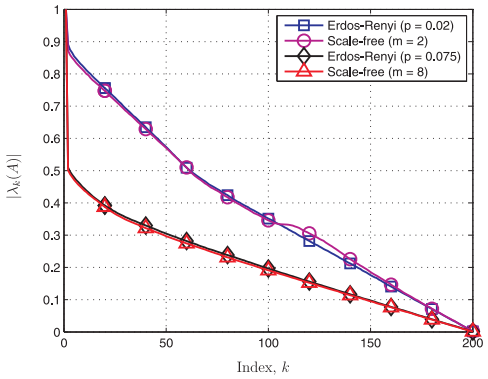

Fig. 1. Eigenvalue distribution $\{|\lambda_k(A)|\}$ for the combination matrix A defined by (21).

stand-alone adaptive filters [10]. Second, expression (34) is inversely proportional to the average degree of the network. That is, when the network is more connected, the network will have lower $\text{MSD}_{k=1}$. Third, expression (34) depends on the distribution of informed nodes through its dependence on the degree of the informed nodes. If the number of informed nodes increases by one, the value of $\text{MSD}_{k=1}$ may increase or decrease. More precisely, from (34) we see that $\text{MSD}_{k=1}$ will increase if the degree of the added node satisfies:

$$\sigma^2_{v,N_I+1} n_{N_I+1} > \frac{\sum_{l=1}^{N_I} \sigma^2_{v,k} n_l^2}{\sum_{l=1}^{N_I} n_l} \tag{35}$$

Finally, we note that since we have expressions for r_1 and s_1, we can obtain from (26) an expression for $\rho(\mathcal{X})$:

$$\boxed{\begin{aligned}\rho(\mathcal{X}) &= \lambda_{1,M}(\mathcal{X}) \\ &\approx 1 - \mu\lambda_M(R_u)\left(\frac{\sum_{l\in\mathcal{N}_I} n_l}{\sum_{l=1}^{N} n_l}\right)\end{aligned}} \tag{36}$$

Expression (36) can be motivated by noting that the decay of $\rho(\mathcal{X})$ will be larger if the informed nodes have higher degrees.

For the second part, $\text{MSD}_{k>1}$, since $\{\lambda_k^2(A)\}$ for $2 \le k \le N$ are generally much smaller than $\lambda_1(A) = 1$, for sufficiently small step-sizes, the denominator in (31) can be approximated to $1 - \lambda_k^2(A)$. Then, $\text{MSD}_{k>1}$ becomes

$$\text{MSD}_{k>1} \approx \frac{\mu^2 \text{Tr}(R_u)}{N} \sum_{k=2}^{N} \left[\frac{\lambda_k^2(A)}{1-\lambda_k^2(A)} \sum_{l=1}^{N_I} \sigma^2_{v,l} |s_{k,l}|^2\right] \tag{37}$$

Expression (37) requires the eigenvectors of the matrix A^T. To simplify the expression, we introduce the following assumption:

$$|s_{k,l}|^2 \approx \frac{n_l}{\sum_{j=1}^{N} n_j} \tag{38}$$

That is, we assume that the magnitude of $s_{k,l}$ is proportional to its degree n_l and that the eigenvectors $\{s_k\}$ for $2 \le k \le N$ have unit magnitude. Then, expression (37) simplifies to:

$$\text{MSD}_{k>1} \approx \frac{\mu^2 \text{Tr}(R_u)}{N} \frac{\sum_{l=1}^{N_I} \sigma^2_{v,l} n_l}{\eta} \frac{1}{N} \sum_{k=2}^{N} \frac{\lambda_k^2(A)}{1-\lambda_k^2(A)} \tag{39}$$

Furthermore, from Fig. 1, we observe that the magnitude of the remaining eigenvalues of matrix A decreases almost in a linear manner. This suggests the following approximation:

$$|\lambda_k(A)| \approx \frac{N-k}{N-2} |\lambda_2(A)| \tag{40}$$

Using an integral approximation, the summation in (39) can be written as:

$$\begin{aligned}\frac{1}{N}\sum_{k=2}^{N} \frac{\lambda_k^2(A)}{1-\lambda_k^2(A)} &\approx \int_0^1 \frac{\lambda_2^2(A)\cdot(1-x)^2}{1-\lambda_2^2(A)\cdot(1-x)^2} dx \\ &= h(|\lambda_2(A)|)\end{aligned} \tag{41}$$

where $h(\alpha)$ is defined as

$$h(\alpha) \triangleq \left[\frac{1}{2\alpha}\log\left(\frac{1+\alpha}{1-\alpha}\right) - 1\right] \tag{42}$$

Substituting expression (41) into (39), we find that the MSD contributed by the remaining terms ($k > 1$) has the form:

$$\boxed{\text{MSD}_{k>1} \approx \frac{\mu^2 \text{Tr}(R_u)}{N} \frac{\sum_{l=1}^{N_I} \sigma^2_{v,l} n_l}{\eta} h\left(|\lambda_2(A)|\right)} \tag{43}$$

Note that, in contrast to (34), expression (43) always increases when the number of informed nodes increases. In addition, $|\lambda_2(A)|$ can be approximated to [16]:

$$\lambda_2(A) \approx \frac{2}{\sqrt{\eta}} \tag{44}$$

Since the function $h(\alpha)$ can be shown to be strictly increasing and convex in α when $\alpha \in (0,1)$, when η (or, p or m) increases, $\text{MSD}_{k>1}$ in (43) decreases. That is, similar to $\text{MSD}_{k=1}$ in (34), the value of $\text{MSD}_{k>1}$ is lower if the network is more connected.

Combining expressions (34) and (43), we obtain an expression for the MSD:

$$\boxed{\begin{aligned}\text{MSD} \approx &\frac{M\mu}{2N\eta} \cdot \frac{\sum_{l\in\mathcal{N}_I} \sigma^2_{v,l} n_l^2}{\sum_{l\in\mathcal{N}_I} n_l} \\ &+ \frac{\mu^2 \text{Tr}(R_u) h(2/\sqrt{\eta})}{N\eta} \sum_{l\in\mathcal{N}_I} \sigma^2_{v,l} n_l\end{aligned}} \tag{45}$$

We observe that the MSD in (45) depends on the network topology only through the average degree of the network η. Additionally, the MSD in (45) depends on the distribution of informed nodes through their degrees, n_l, and noise variances, $\sigma^2_{v,l}$. That is, the effect of different types of network models only depends on the degree distribution of the nodes.

VI. Simulation Results

We consider networks with 200 nodes. The weight vector is set to $w^\circ = [8; 5]$ ($M = 2$). All nodes have the same covariance matrices ($R_{u,k} = I_2$) and noise variances ($\sigma^2_{v,k} = 0.01$). Also, $\mu_k = 0.05$ for informed nodes. Without loss of generality, we assume that the nodes are indexed in decreasing order of degree, i.e., $n_1 \ge n_2 \ge \cdots \ge n_N$.

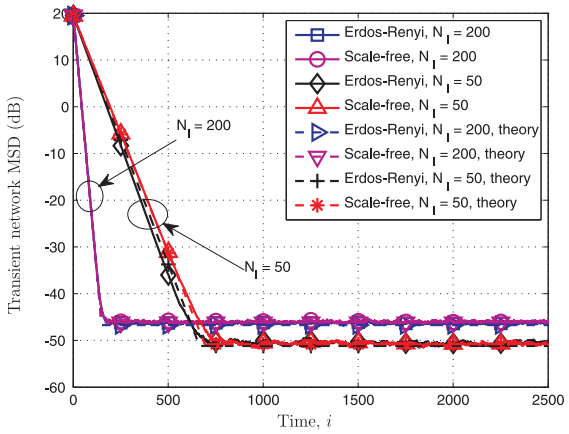

Fig. 2. Transient network MSD in Erdos-Renyi and scale-free models. The dash lines represent theoretical results (18) and (20).

We first verify the theoretical results in (18) and (20). Figure 2 shows the MSD over time for two network models with parameters $p = 0.02$, $m = 2$, and $N_0 = 10$. In each network model, we consider two cases: all and 50 (randomly selected) informed nodes. We observe that when there are 50 informed nodes, the convergence factor increases, as expected, but interestingly, the MSD decreases. For the theoretical results, the MSD decays at rate f in (18) during the transient stage. When the value of the MSD is lower than the steady-state MSD value from (20), the MSD stays constant and equals (20). We observe that the results match well with simulations.

In Fig. 3, We show the effect of the number and distribution of informed nodes on the convergence factor and the MSD of the network. We increase the number of informed nodes, N_I, from the node with the highest degree, i.e., from node 1 to node N. For each model, we consider two possible values of parameters: $p = 0.02$ and 0.075 in the Erdos-Renyi model and $m = 2$ and 8 in the scale-free model. As expected, the convergence factors is smaller for larger N_I and expression (36) matches well with the simulation results. Interesting patterns are seen in the MSD behavior. From (35), $\text{MSD}_{k=1}$ in (34) decreases in N_I in this case, whereas $\text{MSD}_{k>1}$ in (43) always increase in N_I. Therefore, the change of the MSD depends on the values of these two terms. We observe from Fig. 3 that in Erdos-Reyni model, the MSD in (45) increases in N_I. However, in scale-free model, especially for large value of m, we observe that the MSD for scale-free network decreases first and then increases after a certain value of N_I. We also see that the MSD in (45) matches well with expression (20).

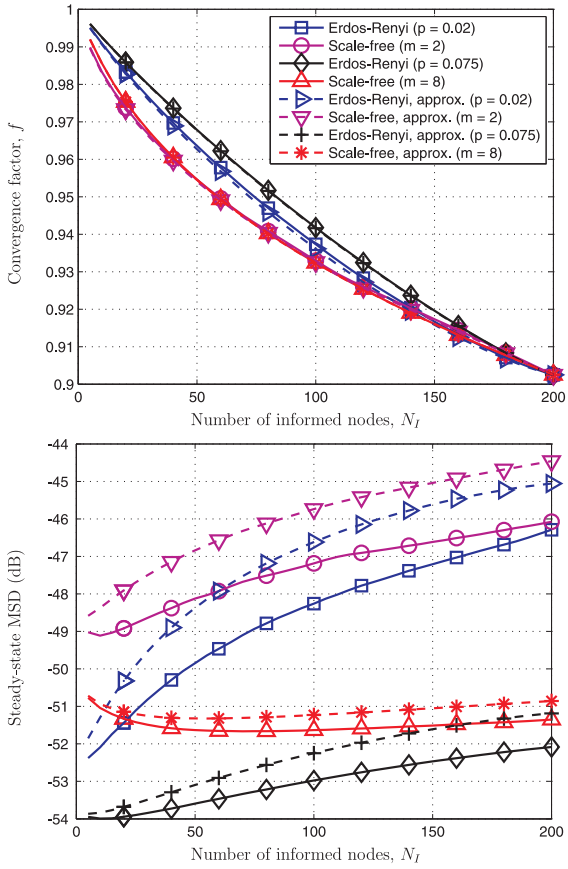

Fig. 3. Convergence factor (top) and steady-state MSD (bottom) in Erdos-Renyi and scale-free models. The dash lines denote results in (36) and (45).

REFERENCES

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] J. Li and A. H. Sayed, "Modeling bee swarming behavior through diffusion adaptation with asymmetric information sharing," *EURASIP Journal on Advances in Signal Processing*, 2011.

[7] T. D. Seeley, R. A. Morse, and P. K. Visscher, "The natural history of the flight of honey bee swarms," *Psyche*, vol. 86, pp. 103–114, 1979.

[8] S. G. Reebs, "Can a minority of informed leaders determine the foraging movements of a fish shoal?" *Anim. Behav.*, vol. 59, pp. 403–409, 2000.

[9] M. E. J. Newman, D. J. Watts, and A. L. Barabasi, *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[10] A. H. Sayed, *Adaptive Filters*. NJ. Wiley, 2008.

[11] T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Trans. on Signal Processing*, vol. 51, no. 3, pp. 639–652, Mar. 2003.

[12] R. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[13] P. Erdos and A. Renyi, "On the evolution of random graphs," *Magyar Tud. Akad. Mat. Kutato Int. Kozl.*, vol. 5, pp. 17–61, 1960.

[14] A. L. Barabasi and E. Bonabeau, "Scale-free networks," *Scientific American*, vol. 288, pp. 50–59, 2003.

[15] A. L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[16] F. Chung, L. Li, and V. Vu, "Spectra of random graphs with given expected degrees," *Proc. National Academy of Sciences*, vol. 100, no. 11, pp. 6313–6318, 2003.