DIFFUSION LEAST-MEAN SQUARES WITH ADAPTIVE COMBINERS

Noriyuki Takahashi†*, Isao Yamada*†*, and Ali H. Sayed*‡

†Department of Communications and Integrated Systems, Tokyo Institute of Technology,
S3-60, 2-12-1, Ookayama, Meguro-ku, Tokyo 152-8550, Japan
‡Department of Electrical Engineering, University of California, Los Angeles, CA 90095
Email: †{takahashi, isao}@comm.ss.titech.ac.jp, ‡sayed@ee.ucla.edu

ABSTRACT

This paper presents an efficient adaptive combination strategy for diffusion algorithms over adaptive networks in order to improve the robustness against the spatial variation of SNR over the network. The diffusion least-mean square (LMS) algorithm with the proposed combination rule and its mean transient analysis are included. Simulation results show that the diffusion LMS algorithm with our combiners outperforms those with existing static combiners and the incremental LMS algorithm.

***Index Terms*—** Adaptive filters, distributed estimation, adaptive networks, diffusion, cooperative systems, adaptive combiners

1. INTRODUCTION

We consider the problem of distributed estimation over adaptive networks [1–4] where, unlike ordinary adaptive filtering, each node of the network is allowed to cooperate with its neighbors in order to estimate some parameter of interest. Such cooperation enables us to leverage the spatial diversity obtained from the distribution of the nodes as well as the temporal diversity, and the performance of system can be improved significantly.

The performance of adaptive networks depends strongly on the mode of cooperation, e.g., incremental [1], diffusion [2], or probabilistic diffusion [3]. In this paper, we focus on the diffusion mode of cooperation because this mode is more robust to node and link failure [3]. In the diffusion mode, the nodes exchange their estimates with neighbors and then try to exploit the collected estimates via convex combination [2]. So far, several combination rules, such as the Metropolis [5] and relative-degree [4] rules, have been proposed that are based solely on the network topology, i.e., combination weights are calculated from the degree of each node and hence do not reflect node profile. Therefore, the performance of such rules deteriorates if the signal-to-noise ratio (SNR) at some nodes is lower than others; because the noisy estimate of such a node diffuses into the entire network by cooperation among the nodes.

Initial investigations on adapting the combination weights were done in [2, 6]. In this paper, we take a more systematic approach and formulate a well defined minimum variance unbiased estimation problem. We then use the problem to propose an adaptive combination rule that learns its combination weights so that the effect of noisy estimates is suppressed. In addition, simulation results show that the diffusion LMS algorithm with our adaptive combination rule outperforms existing static combination rules.

This material was based on work supported in part by the National Science Foundation under awards ECS-0601266 and ECS-0725441. The work of Mr. Takahashi was also supported in part by the Japan Society for the Promotion of Science (JSPS) while visiting the UCLA Adaptive Systems Laboratory.

2. DIFFUSION LMS ALGORITHMS

To begin with, let us introduce our notation. We use boldface letters for random variables and normal fonts for deterministic quantities. Capital letters are used for matrices and small letters for vectors. The notation $(\cdot)^T$ and $(\cdot)^*$ stand for transposition and conjugate transposition for vectors and matrices, respectively, and $(\cdot)^*$ is also used to denote complex conjugation for scalars. Expectation is denoted by $E[\,\cdot\,]$.

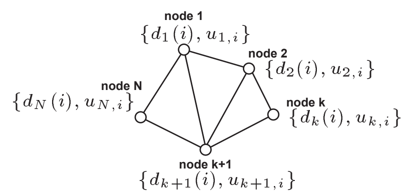


Fig. 1: Adaptive network with N nodes.

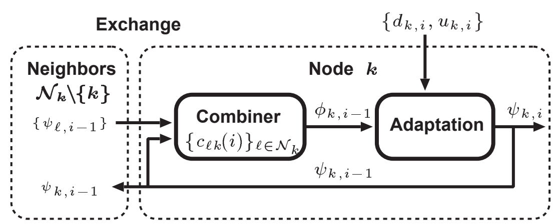


Fig. 2: Combine-then-Adapt (CTA) diffusion strategy [2, 7].

Now, consider N nodes in a predefined network topology and let $\mathcal{N}_k$ denote the neighborhood of node k including k itself; see Fig. 1. At each time i, each node k has access to a scalar measurement $d_k(i)$ and a regression *row* vector $u_{k,i}$ of length M that are related via

$$d_k(i) = u_{k,i} w^o + v_k(i), \tag{1}$$

where w^o is an unknown column vector of length M and $v_k(i)$ is noise. The objective is to generate an estimate $\psi_{k,i}$ of w^o at each node k and time i.

The diffusion strategy we consider is performed in two stages [2]: combination and adaptation (see Fig. 2). In this strategy, each node k first computes a linear combination of local estimates $\{\psi_{\ell,i-1}\}_{\ell\in\mathcal{N}_k}$ collected from its neighbors, i.e.,

$$\phi_{k,i-1} \triangleq \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(i)\, \psi_{k,i-1}, \tag{2}$$

where $\{c_{\ell k}(i)\}_{\ell\in\mathcal{N}_k}$ is a possibly time-varying combination weight calculated from information available at node k. Then, $\phi_{k,i-1}$ is used by an adaptive filter to adapt local data $\{d_k(i), u_{k,i}\}$. We refer to this strategy as *Combine-then-Adapt (CTA)* diffusion [2]. Also, it is possible to reverse the order of the two stages, i.e., adaptation followed by combination. We refer to this version as *Adapt-then-Combine (ATC)* diffusion [7]. For example, using the LMS algorithm as the core adaptive filter yields the following CTA and ATC diffusion algorithms [2]:

CTA diffusion LMS [2]:

$$\begin{cases} \phi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(i)\psi_{\ell,i-1} \\ \psi_{k,i} = \phi_{k,i-1} + \mu_k u_{k,i}^*(d_k(i) - u_{k,i}\phi_{k,i-1}) \end{cases} \tag{3}$$

ATC diffusion LMS [7]:

$$\begin{cases} \phi_{k,i} = \psi_{k,i-1} + \mu_k u_{k,i}^*(d_k(i) - u_{k,i}\psi_{k,i-1}) \\ \psi_{k,i} = \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(i)\phi_{\ell,i} \end{cases} \tag{4}$$

In the CTA and ATC strategies, the combiner weights $\{c_{\ell k}(i)\}$ play an important role. Suppose, for example, that the estimates $\psi_{\ell,i-1}$ collected from some neighbors are less reliable than others due to low SNR conditions. In such case, we should give less weight to the noisy estimates. Thus, static combination rules, which keep $c_{\ell k}(i)$ constant, are likely to result in performance deterioration (see also Sec. 5). To improve the robustness to such cases, we propose an adaptive combination strategy.

3. ADAPTIVE COMBINERS

3.1. Problem Formulation

Let us formulate the problem of controlling the combination weights $c_{\ell k}(i)$. Suppose that for each node k, the local estimates $\{\psi_{k,i}, i = 0, 1, \dots\}$ are realizations of some random vector $\boldsymbol{\psi}_k$. Then, we would like to find a vector of coefficients $c_k = \mathrm{col}\{c_{1k}, c_{2k}, \dots, c_{Nk}\}$ that solves the following problem:

$$\begin{aligned} &\underset{c_k\in\mathbb{R}^N}{\text{minimize}} \quad J(c_k) \triangleq E\|\boldsymbol{\Psi} c_k - w^o\|^2 \\ &\text{subject to} \quad c_{\ell k} = 0 \quad \text{for } \ell \notin \mathcal{N}_k, \end{aligned} \tag{5}$$

where $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_N]$ is an $M \times N$ random matrix. Note that the constraint must be satisfied because node k has no access to realizations of $\{\boldsymbol{\psi}_\ell\}_{\ell\notin\mathcal{N}_k}$. Unfortunately, problem (5) cannot be solved directly due to the presence of the unknown quantity w^o. However, if we assume that every estimate $\boldsymbol{\psi}_k$ is unbiased, i.e., $E\boldsymbol{\psi}_k = w^o$ for all $k = 1, \dots, N$ and hence $E\boldsymbol{\Psi} = w^o\mathbb{1}^T$, we have by the bias-variance decomposition that [8]

$$J(c_k) = \underbrace{c_k^T Q_\Psi c_k}_{\text{variance}} + \underbrace{\|(\mathbb{1}_N^T c_k - 1)w^o\|^2}_{\text{bias}^2},$$

where Q_Ψ is an $N \times N$ matrix defined as

$$Q_\Psi \triangleq E\big[\mathrm{Re}\big\{(\boldsymbol{\Psi} - E\boldsymbol{\Psi})^*(\boldsymbol{\Psi} - E\boldsymbol{\Psi})\big\}\big], \tag{6}$$

$\mathrm{Re}\{\cdot\}$ denotes the real part, and $\mathbb{1}_N$ is the $N \times 1$ vector whose components are all unity. Therefore, if we impose $\mathbb{1}_N^T c_k = 1$, the second term involving the unknown quantity w^o is eliminated and we arrive at the following minimum variance unbiased estimation problem:

$$\begin{aligned} &\underset{c_k\in\mathbb{R}^N}{\text{minimize}} \quad c_k^T Q_\Psi c_k \\ &\text{subject to} \quad \mathbb{1}_N^T c_k = 1 \quad \text{and} \quad c_{\ell k} = 0 \text{ for } \ell \notin \mathcal{N}_k. \end{aligned} \tag{7}$$

The dimension of problem (7) can be reduced from N unknowns to the cardinality of $\mathcal{N}_k$, say n_k, by introducing an auxiliary variable. Let P_k be the $N \times n_k$ matrix whose columns are given by

$$P_k = [\,\text{the } \ell\text{-th column of } I_N\,]_{\ell\in\mathcal{N}_k}.$$

Then, any vector c_k that satisfies $c_{\ell k} = 0$ for $\ell \in \mathcal{N}_k$ can be represented as

$$c_k = P_k a_k \quad \text{with some } a_k \in \mathbb{R}^{n_k}. \tag{8}$$

Therefore, substituting into (7), we get

$$\begin{aligned} &\underset{a_k\in\mathbb{R}^{n_k}}{\text{minimize}} \quad f_k(a_k) \triangleq a_k^T Q_{\Psi,k} a_k \\ &\text{subject to} \quad a_k \in V_k \triangleq \{\, a \in \mathbb{R}^{n_k} \mid \mathbb{1}_{n_k}^T a = 1\,\}, \end{aligned} \tag{9}$$

where $Q_{\Psi,k}$ is the $n_k \times n_k$ matrix defined as

$$Q_{\Psi,k} \triangleq P_k^T Q_\Psi P_k \tag{10}$$

and $\mathbb{1}_{n_k} \triangleq P_k^T \mathbb{1}_N$ is the $n_k \times 1$ vector whose components are all unity. Problem (9) is much simpler than (7) and the solution to (9) is well-known to be [8]:

$$a_k^o \triangleq \frac{Q_{\Psi,k}^{-1}\mathbb{1}_{n_k}}{\mathbb{1}_{n_k}^T Q_{\Psi,k}^{-1}\mathbb{1}_{n_k}},$$

provided that $Q_{\Psi,k}$ is positive definite ($Q_{\Psi,k}$ is at least nonnegative definite). Noting that all components of $Q_{\Psi,k}$ are quadratic moments of the random vectors $\{\boldsymbol{\psi}_\ell\}_{\ell\in\mathcal{N}_k}$, we can compute a_k^o from local information available at node k, if we collect a number of realizations of $\{\boldsymbol{\psi}_\ell\}_{\ell\in\mathcal{N}_k}$ sufficient to estimate the moment $Q_{\Psi,k}$. Then, the solution of (7) can be recovered from a_k^o as $c_k^o = P_k a_k^o$. However, for the purpose of an adaptive implementation, we introduce a steepest-descent solution.

3.2. Steepest-Descent Solution

In order to apply the standard steepest-descent method to (9), we need to eliminate the linear constraint V_k. We apply a similar technique introduced in [9, Example 5]. Let $\mathcal{P}_{V_k}$ be the projection from $\mathbb{R}^{n_k}$ onto V_k, which is given by

$$\mathcal{P}_{V_k}(a) = \left(I_{n_k} - \frac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right)a + \frac{\mathbb{1}_{n_k}}{n_k} \quad \text{for all } a \in \mathbb{R}^{n_k}.$$

We introduce a second auxiliary variable. Since any $a_k \in V_k$ is a projection of some point $b_k \in \mathbb{R}^{n_k}$ onto V_k, we can write

$$a_k = \mathcal{P}_{V_k}(b_k) \quad \text{for some } b_k \in \mathbb{R}^{n_k}. \tag{11}$$

Therefore, substituting this into (9), we arrive at the following unconstrained problem:

$$\underset{b_k\in\mathbb{R}^{n_k}}{\text{minimize}} \quad \mathcal{P}_{V_k}(b_k)^T Q_{\Psi,k}\,\mathcal{P}_{V_k}(b_k). \tag{12}$$

Note that this is just a quadratic function because $\mathcal{P}_{V_k}$ is affine in b_k. Our algorithm is derived by applying the standard steepest-descent method to (12) and then recovering c_k from (8) and (11), i.e.,

$$\begin{cases} b_{k,i} = b_{k,i-1} - \nu_k(i)\left(I_{n_k} - \frac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right)Q_{\Psi,k}\mathcal{P}_{V_k}(b_{k,i-1}), \\ a_{k,i} = \mathcal{P}_{V_k}(b_{k,i}), \qquad c_{k,i} = P_k a_{k,i}, \end{cases}$$

where $\nu_k(i) > 0$ is a stepsize. In this iteration, if we choose an initial point $b_{k,-1}$ from V_k, we can verify that $b_{k,i} \in V_k$ is ensured for all i, i.e., $\mathcal{P}_{V_k}(b_{k,i}) = b_{k,i}$ for all i. Hence, the iteration can be simplified to the following:

$$\begin{cases} b_{k,i} = b_{k,i-1} - \nu_k(i)\left(I_{n_k} - \dfrac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right) Q_{\Psi,k} b_{k,i-1}, \\ c_{k,i} = P_k b_{k,i}, \end{cases} \tag{13}$$

where $b_{k,-1}$ must be chosen such that $\mathbb{1}_{n_k}^T b_{k,-1} = 1$.

3.3. Adaptive Solution

We now replace $Q_{\Psi,k}$ by an instantaneous approximation to derive an adaptive version for (13). In view of (10) and (6), we use the following approximation:

$$Q_{\Psi,k} \approx \mathrm{Re}\{(\Delta\Psi_{k,i-1})^* \Delta\Psi_{k,i-1}\},$$

where $\Delta\Psi_{k,i-1}$ is an $M \times n_k$ matrix defined as

$$\Delta\Psi_{k,i-1} \triangleq [\psi_{\ell,i-1} - \psi_{\ell,i-2}]_{\ell \in \mathcal{N}_k}.$$

Replacing $Q_{\Psi,k}$ in (13) by its instantaneous approximation, we arrive at the following adaptive combination algorithm:

Diffusion with Adaptive Combiners (DAC): At each node k, do the following:

1. Choose $b_{k,-1} > 0$ so that $\mathbb{1}_{n_k}^T b_{k,-1} = 1$ and set $\psi_{k,-1} = \psi_{k,-2} = 0$, where the inequality is componentwise.
2. At each time i, update $b_{k,i-1}$ and compute $c_{k,i}$ via

$$\begin{cases} g_{k,i} = \left(I_{n_k} - \dfrac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right)\mathrm{Re}\{(\Delta\Psi_{k,i-1})^* \Delta\Psi_{k,i-1}\} b_{k,i-1} \\ b_{k,i} = b_{k,i-1} - \nu_k(i)\, g_{k,i}, \qquad c_{k,i} = P_k b_{k,i}. \end{cases} \tag{14}$$

Then, combine $\{\psi_{\ell,i-1}\}_{\ell \in \mathcal{N}_k}$ with $c_{k,i}$ as in (3) or (4).

Replacing $\psi_{k,i}$ by $\phi_{k,i}$ yields the combiners for the ATC version.

A possible choice for $\nu_k(i)$ is the following normalized stepsize:

$$\nu_k(i) = \gamma \frac{\min\{b_{k,i-1}(m) \mid 1 \le m \le n_k\}}{\|g_{k,i}\|_\infty + \varepsilon}, \tag{15}$$

where $\gamma \in (0,1)$ and $\varepsilon > 0$ are constants, $\|\cdot\|_\infty$ denotes the maximum norm, and $b_{k,i-1}(m)$ is the m-th component of $b_{k,i-1}$. This rule keeps $b_{k,i} > 0$ for all i and the weight vector $c_{k,i} = P_k b_{k,i}$ enforces a convex combination. Hence, the Gerschgorin Circle Theorem ensures $|\lambda_{\max}(C_i)| \le 1$, where $|\lambda_{\max}(C_i)|$ denotes the spectral radius of $C_i = [c_{1,i}, \ldots, c_{N,i}]$. As we will see in the next section, this is not sufficient but at least a necessary condition for the stability in the mean sense.

4. MEAN TRANSIENT ANALYSIS

Let us comment briefly on the mean transient analysis of the CTA diffusion LMS algorithm with (14), by following the arguments of [3]. Let

$$w^{(o)} \triangleq \mathbb{1}_N \otimes w^o, \qquad \boldsymbol{d}_i \triangleq \mathrm{col}\{\boldsymbol{d}_1(i), \ldots, \boldsymbol{d}_N(i)\},$$
$$\boldsymbol{U}_i \triangleq \mathrm{diag}\{\boldsymbol{u}_{1,i}, \ldots, \boldsymbol{u}_{N,i}\}, \qquad \boldsymbol{v}_i \triangleq \mathrm{col}\{\boldsymbol{v}_1(i), \ldots, \boldsymbol{v}_N(i)\},$$

where $\otimes$ denotes the Kronecker product and $\boldsymbol{d}_k(i)$, $\boldsymbol{u}_k(i)$, and $\boldsymbol{v}_k(i)$ are random variables. Then, by the model (1), we have $\boldsymbol{d}_i = \boldsymbol{U}_i w^{(o)} + \boldsymbol{v}_i$. Moreover, let us introduce

$$\boldsymbol{C}_i \triangleq [\boldsymbol{c}_{1,i}, \ldots, \boldsymbol{c}_{N,i}], \quad D \triangleq \mathrm{diag}\{\mu_1 I_M, \ldots, \mu_N I_M\},$$
$$\boldsymbol{G}_i \triangleq \boldsymbol{C}_i^T \otimes I_M, \qquad \boldsymbol{\psi}^{i-1} \triangleq \mathrm{col}\{\boldsymbol{\psi}_{1,i-1}, \ldots, \boldsymbol{\psi}_{N,i-1}\},$$
$$\widetilde{\boldsymbol{\psi}}^{i-1} \triangleq w^{(o)} - \boldsymbol{\psi}^{i-1},$$

where realizations of $\boldsymbol{c}_{k,i}$ appear in (14). Then, by (3) and the fact that $\boldsymbol{G}_i w^{(o)} = w^{(o)}$, we have

$$\widetilde{\boldsymbol{\psi}}^i = (I_{NM} - D\boldsymbol{U}_i^*\boldsymbol{U}_i)\boldsymbol{G}_i\widetilde{\boldsymbol{\psi}}^{i-1} - D\boldsymbol{U}_i^*\boldsymbol{v}_i.$$

Assuming (i) temporal and spatial independence of the regressors, (ii) that the noise is zero-mean and independent of the regressors, and (iii) the independence of $\boldsymbol{G}_i$ and $\widetilde{\boldsymbol{\psi}}^{i-1}$, we get

$$\boxed{E\widetilde{\boldsymbol{\psi}}^i = \left(I_{NM} - DE[\boldsymbol{U}_i^*\boldsymbol{U}_i]\right) E[\boldsymbol{G}_i]\, E\widetilde{\boldsymbol{\psi}}^{i-1}}$$

Therefore, letting $B \triangleq I_{NM} - DE[\boldsymbol{U}_i^*\boldsymbol{U}_i]$, we find that the stability in the mean is ensured if the spectral radius of $BE[\boldsymbol{G}_i]$ is strictly less than one for all i. Noting that B is Hermitian and $\boldsymbol{G}_i = \boldsymbol{C}_i^T \otimes I_M$, we have $|\lambda_{\max}(BE[\boldsymbol{G}_i])| \le |\lambda_{\max}(B)| \cdot \|E\boldsymbol{C}_i\|_2$, where $\|\cdot\|_2$ is the matrix 2-norm. Hence, if every realization C_i of $\boldsymbol{C}_i$ is controlled so that $\|C_i\|_2 \le 1$, our combiner has a stabilizing effect over the noncooperative case, which corresponds to $\boldsymbol{C}_i = I_N$. Note that $|\lambda_{\max}(C_i)| \le 1$ is a necessary condition for $\|C_i\|_2 \le 1$.

5. SIMULATION EXAMPLES

We present simulation examples to illustrate the performance of our adaptive combiner. We consider two scenarios over the network topology with $N = 15$ nodes of Fig. 3. The network statistical profile for scenario 1 is also depicted in Fig. 3 and that of scenario 2 is the same as scenario 1, except for the noise power at node 5. In scenario 2, $\sigma_{v,5}^2$ is set to 10^{-2} (17.58 dB in SNR) and hence the weights for node 5 are critical for the network performance. Through these examples, we would like to illustrate the importance of the adaptive combiners as well as the performance of our algorithm. The unknown vector is set to $w^o = \mathbb{1}_5/\sqrt{5}$ ($M = 5$). The regressors are zero-mean Gaussian, independent in time and space. We compare the ATC diffusion LMS algorithms (4) equipped with our combiners and several other existing combiners listed in Table 1, together with the LMS algorithm without cooperation (i.e., $c_{kk} = 1$ and $c_{\ell k} = 0$ for $\ell \ne k$). For all algorithms, the stepsize of the LMS is set to $\mu_k = 0.01$, i.e., all algorithms use the same LMS algorithm and only the combiners are different. We also compare the above algorithms with the incremental LMS algorithm [1], for which the stepsize is set to $\mu_k = 0.01/N$ because the incremental LMS algorithm uses the LMS-type iterations N times for every i. For the proposed combination rule, a uniform weight is chosen as the initial weight and the stepsize rule (15) with $\gamma = 0.01$ and $\varepsilon = 10^{-6}$ is used. Results of CTA versions are omitted due to lack of space but we observed that ATC versions outperform CTA versions for all combiners.

Figure 4 shows the learning behavior of each algorithm in terms of the network mean-square deviation (MSD):

$$\eta^{\text{network}}(i) \triangleq \frac{1}{N}\sum_{k=1}^{N} \eta_k(i), \quad \eta_k(i) \triangleq E\|w^o - \boldsymbol{\psi}_{k,i}\|^2,$$

where $\eta_k(i)$ is the MSD at node k and the expectation is calculated by averaging 200 independent experiments. We observe that the proposed algorithm outperforms the other algorithms especially in scenario 2. On the other hand, Fig. 5 shows the steady-state MSD at

Table 1: Values of $c_{\ell k}$ for $\ell \in \mathcal{N}_k \setminus \{k\}$ for several combination rules. All rules choose $c_{kk} = 1 - \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} c_{\ell k}$ and $c_{\ell k} = 0$ for $\ell \notin \mathcal{N}_k$.

Rule	$c_{\ell k}$ for $\ell \in \mathcal{N}_k \setminus \{k\}$
Uniform	$1/n_k$
Maximum degree	$1/\max\{n_m \mid 1 \le m \le N\}$
Metropolis [5]	$1/\max(n_k, n_\ell)$
Relative degree [4]	$n_\ell / \sum_{m \in \mathcal{N}_k} n_m$

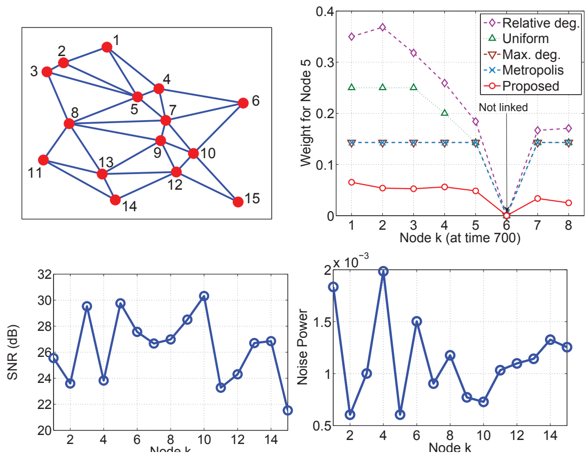
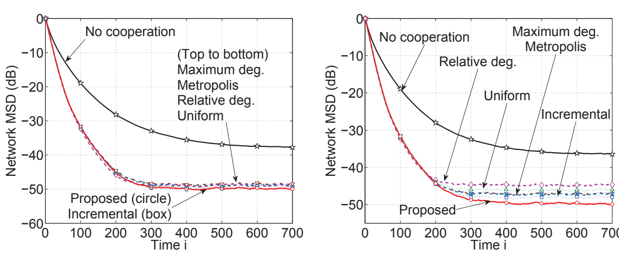


Fig. 3: Network topology (top), SNR (bottom left), and noise variance $\sigma_{v,k}^2$ (bottom right) for $N = 15$ nodes. The SNR at node 5 in scenario 2 is 17.58 dB. The top right figure shows combination weights for node 5 in scenario 2 at time 700, i.e., $c_{5k}(700)$ $(k = 1, \ldots, N)$.



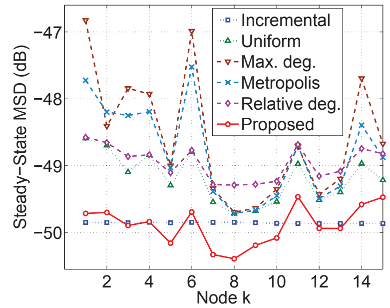
(a) Scenario 1 (b) Scenario 2

Fig. 4: Learning behavior of network MSDs.

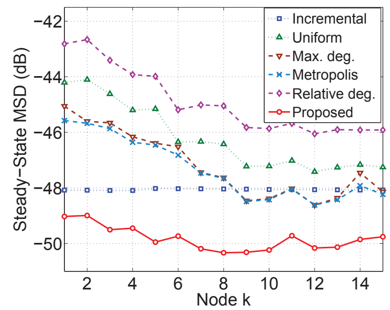
each node, which is obtained by averaging the last 100 samples after convergence. The performance of our algorithm in scenario 2 is still as good as scenario 1 in spite of the presence of the noisy node 5, while significant deterioration is observed in the other algorithms. To see why our algorithm is robust, we show in the top right plot in Fig. 3 the weights $c_{\ell k}(i)$ for node 5. Compared with the other static rules, our adaptive combiner successfully puts small weights for node 5, which leads to robustness. In contrast, the performance deterioration of the relative-degree rule is significant because it puts weight on node 5.

6. CONCLUDING REMARKS

An efficient adaptive combination rule for diffusion algorithms over adaptive networks has been proposed to improve robustness to the spatial variation in SNR conditions. Although we focused on the LMS algorithm as the adaptive filter module, combinations with other adaptive filters are possible. Furthermore, our combiners enable us to develop a self-maintaining network, where we can remove or detect noisy nodes in a distributed manner and maintain the network at a good condition.



(a) Scenario 1



(b) Scenario 2

Fig. 5: Comparison of steady-state MSDs.

7. REFERENCES

[1] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

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

[3] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proc. IEEE ICASSP*, Las Vegas, NV, Apr. 2008, pp. 3285–3288.

[4] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[5] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, Sept. 2004.

[6] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[7] A. H. Sayed and F. S. Cattivelli, "Distributed adaptive learning mechanisms," to appear in the *Handbook on Array Processing and Sensor Networks,* S. Haykin and K. J. Ray Liu, Editors, Wiley, NJ, 2009.

[8] A. H. Sayed, *Adaptive Filters*, John Wiley & Sons, NJ, 2008.

[9] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of a sequence of nonnegative convex functions," *Numer. Funct. Anal. Optim.*, vol. 25, no. 7, pp. 593–617, Jan. 2004.