Towards Spatially Universal Adaptive Diffusion Networks

Cássio G. Lopes, Luiz F. O. Chamon, and Vítor H. Nascimento
Dept. of Electronic Systems Engineering, University of São Paulo – Brazil
cassio@lps.usp.br, vitor@lps.usp.br, chamon@usp.br

***Abstract*—Adaptive networks (ANs) rely on local adaptive filters (AFs) and a cooperation protocol to achieve a common goal, e.g., estimating a set of parameters. This protocol fuses the information from the rest of the network based on local combiners whose design impacts directly the network performance. Indeed, although diffusion schemes improve network performance on average, heterogeneity in signal statistics implies that indiscriminate cooperation may not be the best policy for good nodes. In this work, these observations lead to the introduction of different concepts of spatial universality which motivate a new adaptive combiner structure. The goal of the new combiner is to enforce that the cooperative AFs perform at least as well as the best individual non-cooperative AF, without discarding information from other nodes. The new structure has lower complexity and outperforms existing techniques, as illustrated by simulations. Network learning analysis is also provided.**

***Index Terms*—Adaptive network, Diffusion, Adaptive filtering, Spatial universality**

I. Introduction

Adaptive Networks (ANs) consist of a pool of agents or experts—called *nodes*—that estimate a set of parameters using cooperative adaptive filters (AFs) operating on local data. Information from neighboring nodes is aggregated by means of combination weights and the result is explored by the local AF [1]–[7]. Originally, these combiners were fixed and weighting rules such as the uniform and the relative-variance [2], [5] were used to explore spatial diversity.

Since the introduction of the diffusion AN, several adaptive combiners have been proposed attempting to improve its performance, such as the adaptive diffusion [2], the MSD combiner [8], the least-squares (LS) adaptive combiner [9], the switching combiner [10], and the neighbor-selective combiner [11]. At the expense of additional computational complexity, network global performance improves, but not necessarily at every individual node.

Inspired by concepts from the context of universal prediction [12] and combination of adaptive filters [13], this work addresses the idea of universal estimation in the AN scenario by proposing a new adaptive combiner that promotes spatial universality, enforcing that individual nodes do not lose when cooperating with their peers.

II. Adaptive Networks

In an N-node AN, the n-th node attempts to estimate an unknown $M \times 1$ vector of parameters w^o using an AF and a local data pair $\{d_n(i), u_{n,i}\}$, where $u_{n,i}$ is an $1 \times M$ regressor vector and $d_n(i)$ is a scalar measurement. The network topology is captured by an undirected graph whose adjacency matrix is $A = [a_{mn}]$, where $a_{mn} = 1$ if node m and n are connected to each other and $a_{mn} = 0$ otherwise.

In standard diffusion ANs, a combining matrix C is defined over matrix A which explores spatial diversity by fusing estimates from neighboring nodes [2]. Node n's neighborhood is the set of nodes it is connected to *excluding* itself, i.e., $\mathcal{N}_n = \{\ell \in 1, \dots, N : a_{n\ell} = 1\}$, and its cardinality $\mathcal{D}_n = |\mathcal{N}_n|$ is known as node n's degree. Each node in a standard diffusion LMS AN [2] updates a previous local estimate $w_{n,i-1}$ of w^o by

$$\psi_{n,i-1} = c_{nn}\, w_{n,i-1} + \sum_{\ell \in \mathcal{N}_n} c_{\ell n}\, w_{\ell,i-1} \tag{1a}$$

$$w_{n,i} = \psi_{n,i-1} + \mu_n u_{n,i}^* [d_n(i) - u_{n,i}\psi_{n,i-1}], \tag{1b}$$

where μ_n is a scalar step size and c_{mn} is an element from C representing the diffusion weight attributed by node n to node m. This formulation is sometimes referred to as CTA (Combine Then Adapt). An alternative formulation, the ATC (Adapt Then Combine), can be obtained by simply exchanging $\psi \leftrightarrow w$ and changing the output variable. Given that both formulations are equivalent [7], the original CTA from [2] is used in this work.

The performance of ANs can be improved by changing the adaptive rules at the nodes [3], [9]. Another possibility is to enhance the combination step (1a), for instance by using time-varying combiners $c_{mn}(i)$ (resulting in a time-varying C_i). The first of these techniques was the adaptive diffusion network (ADN), introduced together with the diffusion AN in [2]. Inspired by combinations of AFs, only one adaptive convex combiner $\eta_n(i)$ per node is used

$$\psi_{n,i-1} = \eta_n(i)\, w_{n,i-1} + [1 - \eta_n(i)]\phi_{n,i-1}, \tag{2}$$

with $\phi_{n,i-1} = \sum_{\ell \in \mathcal{N}_n} c_{\ell n}\, w_{\ell,i-1}$.

Pursuing a set of $\mathcal{D}_n$ time-varying combiners per node, [8] used a local approximation of the node's Mean-Square Deviation (MSD)—$\text{MSD}_n(i) = \text{E}\,\|w_{n,i-1} - w_o\|^2$—to drive the combination weights (MSD combiner). Also adapting the $c_{mn}(i)$ individually, an LS adaptive combiner (LS combiner) was introduced in [9], however adapting the AFs indepen-

978-1-4799-7088-9/14/$31.00 ©2014 IEEE

dently of the rest of the network. Explicitly,

$$\begin{aligned}\psi_{n,i} &= \psi_{n,i-1} + \mu u_{n,i}^{*}[d_n(i) - u_{n,i}\psi_{n,i-1}] \\ P_{n,i} &= \sum_{j=i-\mathcal{L}}^{i} \beta_n(i-j)\,\Delta y_n(j)\Delta y_n^T(j) \\ z_{n,i} &= \sum_{j=i-\mathcal{L}}^{i} \beta_n(i-j)\,\Delta y_n(j) e_n(j),\end{aligned} \tag{3}$$

where $\Delta y_n(i) = [\,u_{n,i}(w_{\ell,i-1} - \psi_{n,i-1})\,]_{\ell\in\mathcal{N}_n}$ is a $\mathcal{D}_n \times 1$ vector, $\beta_n(k)$ is a length $\mathcal{L}$ windowing function—rectangular in [9]—, and the elements of C_i are evaluated from $g_{n,i} = P_{n,i}^{-1}\, z_{n,i}$ as $c_{\ell n}(i) = g_{n,i}[\ell]$, for $\ell \in \mathcal{N}_n$, $c_{nn}(i) = 1 - \mathbb{1}^T g_{n,i}$, and 0 otherwise.

Most of these works improve performance over the original diffusion AN, though not without issues. The original ADN, despite its simplicity, only provides small gains in a handful of scenarios [2]. The MSD combiner is effective at reducing the global network misadjustment, albeit at slower convergence rates and with extra computational complexity, requiring $\mathcal{O}(\mathcal{D}_n M^2)$ multiplications per node if $\mathcal{D}_n \ll M$. The LS combiner requires $\mathcal{O}(\mathcal{D}_n^3 \mathcal{L})$ operations and memory for $\mathcal{O}(\mathcal{D}_n \mathcal{L})$ elements to compute and invert $P_{n,i}$, not scaling well with network size and connectivity; besides, $P_{n,i}$ is ill-conditioned (nearly singular) during most of the AN operation, compromising robustness, a fact already noticed in [9].

III. Towards spatial universality

Three fundamental properties are desired for an AN: (i) ability to shield itself from poor estimates, or "rejecting" a poor node whose direct use may degrade its peers; (ii) ability to exploit exceptional nodes, so that their local estimates enhance their neighborhoods; and (iii) node performance homogeneity, as having better network (global) average performance, with some nodes much worse than others, may compromise the targeted application. In summary, a node should not lose from cooperating. In many ways, the concepts from universal estimation and AFs combinations, when interpreted in the (spatial) distributed case, capture such *desiderata*, as explained in the sequel.

To begin with, in the AN literature the network average performance is commonly used as a figure of merit to study cooperation strategies. As such, when the SNR is heterogeneous across the network, with direct cooperation eventually deteriorating the individual performance of some nodes [2], [6], *local* steady-state mean-square performance is more adequate, and in this work the aforementioned local MSD_n is adopted as figure of merit (We consider here only steady-state or *asymptotic universality*).

In combination of AFs, a pool of experts (the AFs) observe the same data pair $\{d(i), u_i\}$ and present their estimates to a supervisor in charge of generating good estimates. The overall system, or the supervisor output, is said to be mean-square universal within the class spanned by the AFs if it performs at least as well as the best individual AF in the mean-square sense.

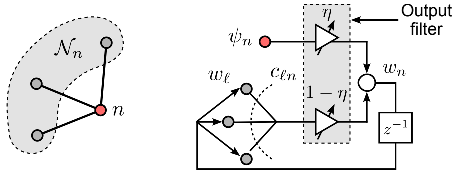


Figure 1. The feedback output filter combiner

In the space-time AN context, considering the N nodes as experts observing N local data pairs $\{d_n(i), u_{n,i}\}$, the AN would be declared *spatially universal*, with respect to its non-cooperative counterpart, if every cooperative node performed at least as well as the best non-cooperative node in the entire network. If the AFs reported to a central supervising node, or in a fully connected network, it would be straightforward to promote spatial universality. In a distributed scenario, with nodes accessing only their neighbors, it is more intricate. The following definitions are relevant to study the problem.

Definition 1 (Local universality)**.** A node n of an AN is said to be locally universal when is at least as good as the best node $m \in \mathcal{N}_n$ in its neighborhood.

Definition 2 (Universality with respect to the non-cooperative strategy)**.** An AN is said to be universal with respect to the non-cooperative strategy when for all nodes $n \in \{1, \ldots, N\}$ it holds that n performs at least as well the best non-cooperative AF in the network.

IV. A spatial universality promoting combiner

The proposed combiner is constructed in two fundamental steps. First, a simple and robust cooperation protocol that *decouples* network learning from node learning, enforcing local universality within each neighborhood (Def. 1). Second, a simple multihop procedure to propagate good estimates throughout the AN (Def. 2). The new algorithm is collected in (4) and explained in the sequel.

$$\phi_{n,i-1} = \sum_{\ell\in\mathcal{N}_n} c_{\ell n}\, w_{\ell,i-1}, \tag{4a}$$

$$w_{n,i} = \eta_n(i)\,\psi_{n,i-1} + [1 - \eta_n(i)]\phi_{n,i-1}, \tag{4b}$$

$$\begin{aligned}a_n(i) = a_n(i-1) &+ \bar{\mu}_{a,n} u_{n,i}(\psi_{n,i-1} - \phi_{n,i-1}) \\ &\times e_n(i)\eta_n(i)[1 - \eta_n(i)],\end{aligned} \tag{4c}$$

$$\psi_{n,a} = \delta_{L_n}(i)\, w_{n,i} + [1 - \delta_{L_n}(i)]\,\psi_{n,i-1}, \tag{4d}$$

$$\psi_{n,i} = \psi_{n,a} + \mu_n u_{n,i}^{*}[d_n(i) - u_{n,i}\psi_{n,a}], \tag{4e}$$

where $a_n(i)$ is limited to $[-4, 4]$; $\eta_n(i) = [1 + e^{-a_n(i-1)}]^{-1}$; $e_n(i) = d_n(i) - u_{n,i}w_{n,i}$; $\bar{\mu}_{a,n} = \mu_{a,n}\,/\,[p_n(i) + \epsilon]$, with a scalar step size $\mu_{a,n}$ and $p_n(i) = \gamma_n p_n(i-1) + [1 - \gamma_n]\,|u_{n,i}(\psi_{n,i-1} - \phi_{n,i-1})|^2$, $0 \ll \gamma_n < 1$; $\psi_{n,a}$ denotes the *a priori* coefficients; L_n is the cycle length of node n; and $\delta_{L_n}(i) = 1$ if $i = rL_n$ for some $r \in \mathbb{N}$ and $\delta_{L_n}(i) = 0$ otherwise. Note that L_n is deterministic and constant, allowing for efficient implementations of the feedback strategy [13].

Equation (4b) introduces the concept of *output filter* (Fig. 1), or *local supervisor*, which is an evolution of [2]; it enforces local universality (Def. 1) as each node protects its estimation process from an underperforming neighborhood. The local estimate is taken as the output filter's, instead of the local AF's. The combination parameter η_n is chosen to minimize some function of $e_n(i) = d_n(i) - u_{n,i}w_{n,i}$—typically the Mean-Square Error (MSE) as in (4c).

Equation (4a) combines the neighborhoods output filters from the previous iteration, injecting the result in the local output filter. This implements a network-level feedback, so that the local universal node is combined with locally universal nodes form adjacent neighborhoods, distributing the estimates of exceptional nodes to the entire network (Def. 2), instead of confining them into their neighborhoods.

The spatial universality promoted by Algorithm (4) is an important improvement over standard diffusion ANs, whose individual cooperative nodes are sensitive to the signals' spatial diversity in arbitrary scenarios [2], [6]. Compared to the algorithms in Section II, (4) has considerably lower complexity, with the advantage of scaling linearly with the network connection density—i.e., with $\mathcal{D}_n$.

Finally, although Alg. (4) decouples network learning and node adaptation, the node-level cyclic feedback in Eq. (4d) allows local AFs to improve their convergence using the estimates from other filters in the network, as in the original diffusion schemes—see Section VI. The feedback is cyclical so as not to deteriorate the steady-state performance of the AN, or stall the combiner adaptation—as argued in [13]. Note that, for $L_n \to \infty$, the node learning is fully decoupled from the network—as in [9]—and for $L_n = 1$, a solution similar to the adaptive diffusion (1b) is recovered.

V. Network learning

A complete mean-square analysis of the ANs in (4) is demanding, so this section analyzes the network learning dynamics assuming the nodes have learned, i.e., the AFs have converged. If the nodes adapt independently ($L_n \to \infty$ in (4d)), this is reasonable and allows the combiner impact to be studied unaffected by the AFs transient behavior.

To do so, recall that $\text{MSD}_n(i) = \mathrm{E}\,\|w_{n,i-1} - w_o\|^2$ and assume that $M = 1$ and $w^o = 0$. These assumptions simplify the derivations and clarify the effects of interest with no loss of generality. Consider then that every node of the network is operating at steady-state. Under these conditions, the local node estimates $\psi_{n,i}$ can be modeled as i.i.d. Gaussian random variables whose variances represent the steady-state MSD of each node. Explicitly, $\Psi_i = [\,\psi_{1,i} \;\cdots\; \psi_{N,i}\,]^T \sim \text{Normal}(b, R_\Psi)$, where b is an $N \times 1$ vector of node biases and $R_\Psi = \mathrm{E}\,\Psi\Psi^T$ is the $N \times N$ covariance matrix of the node estimates. Note that R_Ψ is not assumed to be diagonal, so that it can account for correlation between the nodes estimates.

Under this model, a global network recursion for (4) reads

$$W_i = H_i\Psi_{i-1} + (I - H_i)C^T W_{i-1} \tag{5}$$

$$A_i = A_{i-1} + \bar{\mathcal{M}}_{a,i}H_i(I - H_i)\mathcal{Y}_{i-1}W_i, \tag{6}$$

where $W_i = \text{col}\{w_{n,i}\}$, $\Psi_i = \text{col}\{\psi_{n,i}\}$, $A_i = \text{col}\{a_n(i)\}$, and $\bar{\mathcal{M}}_{a,i} = \text{diag}\{\bar{\mu}_{a,n}\}$ are $N \times 1$ vectors; $\mathcal{Y}_{i-1} = \text{diag}\left\{C^T W_{i-1} - \Psi_{i-1}\right\}$ and $H_i = \text{diag}\{[1 + e^{-a_n(i-1)}]^{-1}\}$ are $N \times N$ matrices; and I is the identity matrix. Notice that, to be consistent with (4a), the diagonal of C in (5) and (6) is 0—i.e., $c_{nn} = 0$.

A. The effect of network-level feedback

To better understand the importance of the network-level feedback in (4), this section compares the effect of the global output filter with and without feedback. The case without feedback is obtained by replacing W_{i-1} by Ψ_{i-1} in (5)

$$W_i = H_i\Psi_{i-1} + [(I - H_i)C^T]\Psi_{i-1}. \tag{7}$$

This is as if the nodes distributed their local estimates $\psi_{n,i}$, as in the adaptive diffusion (2). This resulting network is equivalent to a standard diffusion AN (1) with combination matrix $\check{C}_i = H_i + (I - H_i)C^T$, i.e., with adaptive c_{nn}.

In contrast, using network-level feedback yields

$$W_i = H_i\Psi_{i-1} + \sum_{k=1}^{i-1}\prod_{j=0}^{i-k-1}\left[(I - H_{i-j})C^T\right]H_k\Psi_{k-1} + \prod_{k=0}^{i-1}\left[(I - H_{i-k})C^T\right]\Psi_0, \tag{8}$$

which depends on local estimates and combiners from the entire network. By propagating $w_{n,i}$, the nodes can take advantage of information beyond their neighborhoods. As indicated by the second term in (8), this information traverses the network in a multihop fashion. By comparing (7) and (8) the limitations of the adaptive diffusion from [2] and the advantages of network-level feedback become clear.

B. Network learning behavior

The following derivations provide insights into the effect of the combiner adaptation on network learning. To do so, assume that **(A.1)** *the variance of* $\{\eta_n(i)\}$ *is small enough that* $\mathrm{E}\,H_i \approx \text{diag}\{[1 + e^{-\mathrm{E}\,a_n(i-1)}]^{-1}\} = \bar{H}_i$ *and* $\mathrm{E}[H_i^2] \approx \bar{H}_i^2$; and **(A.2)** $\{\eta_n(i)\}$ *varies slowly enough compared to* $w_{n,i}$ *so that* $\mathrm{E}[H_iW_i] \approx \bar{H}_i\,\mathrm{E}[W_i]$. These assumptions are commonly used in the analysis of combinations of AFs and give good agreement simulations—see Section VI [13].

Defining $K_i = \mathrm{E}\,W_iW_i^T$ and $\bar{W}_i = \mathrm{E}\,W_i$, one gets from (5)

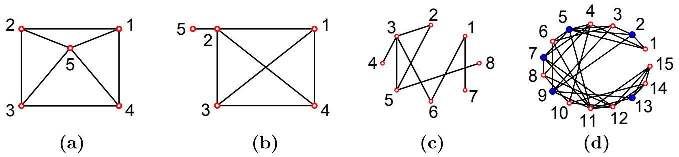
$$\bar{W}_i = \bar{H}_i b + (I - \bar{H}_i)C^T\bar{W}_{i-1} \tag{9a}$$

$$\begin{aligned} K_i = {} & \bar{H}_i(R_\Psi + bb^T)\bar{H}_i + (I - \bar{H}_i)C^T\bar{W}_{i-1}b^T\bar{H}_i \\ & + \bar{H}_i\,b\bar{W}_{i-1}^T C(I - \bar{H}_i) \\ & + (I - \bar{H}_i)C^T K_{i-1}C(I - \bar{H}_i) \end{aligned} \tag{9b}$$

where A.1, A.2, and the fact that $\{\Psi_{i-1}, H_i\}$ are independent were used. Under A.1 and A.2, only $\bar{A}_{i-1}$ is required to evaluate $\bar{H}_i$ in (9). Thus, taking the expected value of (6),

$$\bar{A}_i = \bar{A}_{i-1} + \mathrm{E}\,\bar{\mathcal{M}}_{a,i}\bar{H}_i(I - \bar{H}_i)\,\mathcal{K}_i, \tag{10}$$

where $\mathcal{K}_i$ is an $N \times 1$ vector with elements $\mathcal{K}_i[n] = [1 - \eta_n(i)]c_n^T K_{i-1}c_n - \eta_n(i)(\sigma_n^2 + b_n^2) + [2\eta_n(i) - 1]c_n^T\,(\bar{W}_{i-1} \circ b)$, c_n is the n-th column of C, b_n is the n-th element of the



(a) (b) (c) (d)

Figure 2. Networks used in the simulations

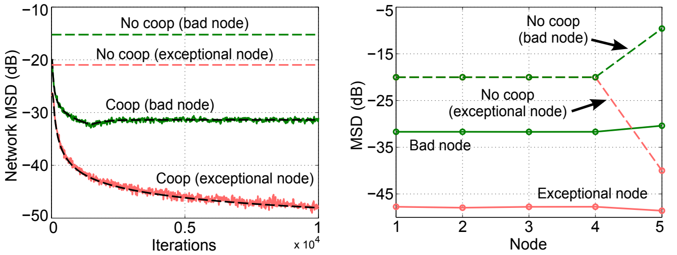


Figure 3. Network learning dynamics. **Rejecting a bad node:** Network in Fig. 2a, $R_\Psi = 10^{-2}I$, $b = [0\ 0\ 0\ 0\ \sqrt{0.1}]^T$, $\mu_{a,n} = 0.4$, $\epsilon = 10^{-2}$; **Exploiting an exceptional node:** Network in Fig. 2b, $R_\Psi = \operatorname{diag}\{[10^{-2}\ 10^{-2}\ 10^{-2}\ 10^{-2}\ 10^{-4}\ ]\}$, $b = 0$, $\mu_{a,n} = 0.2$, $\epsilon = 10^{-3}$.

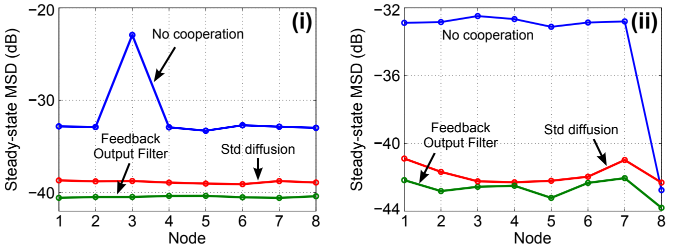


Figure 4. New combiner without node-level feedback. **For all simulations:** Network in Fig. 2c, $\sigma^2_{x,n} = [0\ -3\ -12\ -10\ -9\ -10\ -4\ -4]$ (dB), $\alpha_n = [0.4\ 0.1\ 0.2\ 0.7\ 0.9\ 0.5\ 0.5\ 0.7]$, $M = 5$, $\mu_n = 0.1/M$, $\gamma_n = 0.9$. **Rejecting a bad node (i):** $\sigma^2_{v,n} = 10^{-2}$ for all nodes except $\sigma^2_{v,3} = 10^{-1}$, $\mu_{a,n} = 0.01$, $\epsilon = 10^{-3}$; **Exploiting an exceptional node (ii):** $\sigma^2_{v,n} = 10^{-2}$ for all nodes except $\sigma^2_{v,8} = 10^{-3}$, $\mu_{a,n} = 0.1$, $\epsilon = 10^{-3}$.

bias vector b, σ^2_n is the n-th element of the diagonal of R_Ψ, and $\circ$ denotes the Hadamard product. Recall that the elements of $\bar{A}_i$ are limited to the interval $[-a^+, a^+]$. Also, $\mathrm{E}\,\bar{\mathcal{M}}_{a,i} \approx \mathcal{M}_{a,i} \operatorname{diag}\{(\bar{p}_{n,i} + \epsilon)^{-1}\}$ with $\bar{p}_{n,i} = \nu_n \bar{p}_{n,i-1} + (1-\nu_n)[\, c_n K_{i-1} c_n^T - 2c_n(\bar{W}_{i-1} \circ b) + \sigma^2_n + b^2_n\,]$.

The first thing to notice from (9) and (10) is that when C follows the uniform rule [14], (9a) displays a consensus step $C^T \bar{W}_{i-1}$ that distributively averages estimates from the entire network. The $\{\eta_n(i)\}$ are in fact selecting between pure node learning and pure network learning ANs. However, instead of hard switching between these two states—similar to [10]—or using the information from other nodes selectively—as in [11]—it combines both learning methods in a weighted fashion. Finally, notice that the adaptation (10) depends on $\mathcal{K}_i$, whose n-th element compares the aggregated performance of $\mathcal{N}_n$ with the performance of node n, namely $\sigma^2_n + b^2_n$.

VI. Simulations

Data for all simulations are taken from the zero mean Gaussian i.i.d. sequences $\{x_n(i)\}$ and $\{v_n(i)\}$ with variances $\sigma^2_{x,n}$ and $\sigma^2_{v,n}$. The input regressor $u_{n,i}$ collects the realizations $u_n(i) = \alpha_n u(i-1) + \sqrt{1-\alpha^2_n}\,x(i)$. Network MSD curves are averaged across the nodes MSDs. All curves are ensemble averages of 100 realizations.

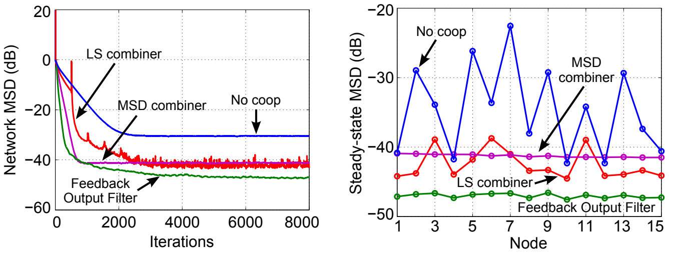


Figure 5. Feedback output filter with node-level feedback. **For all simulations:** Network in Fig. 2d, NLMS, $M = 50$, $\sigma^2_{x,n} = 1$, $\alpha_n = 0$, SNR $\sim$ Uniform(20, 30) (dB); Cluster 1 (1, 3, 4, 6, 8, 10, 11, 12, 14, 15): $\mu_n = 0.1$; Cluster 2 (2, 5, 7, 9, 13): $\mu_n = 1$. **MSD supervisor:** $\nu_n = 0.8$, $\epsilon = 10^{-3}$ [8]; **LS combiner:** $\mathcal{L} = 500$ [9]; **Feedback output filter:** $\mu_{a,n} = 0.8$, $\epsilon = 10^{-3}$, $\gamma_n = 0.9$, $L_n = 200$.

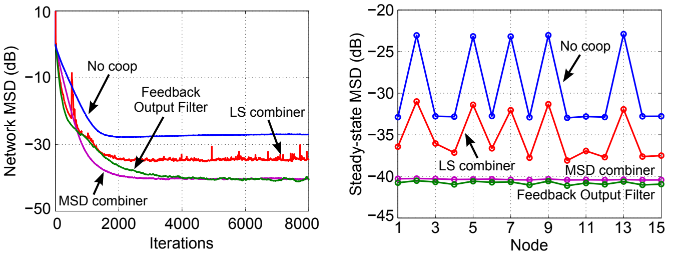


Figure 6. Feedback output filter with node-level feedback. **For all simulations:** Network in Fig. 2d, $\sigma^2_{x,n} = 1$; Cluster 1 (1, 3, 4, 6, 8, 10, 11, 12, 14, 15): $\sigma^2_{v,n} = 10^{-2}$, $\alpha_n = 0.3$; Cluster 2 (2, 5, 7, 9, 13): $\sigma^2_{v,n} = 10^{-1}$, $\alpha_n = 0.8$; $M = 50$, $\mu_n = 0.1/M$. **MSD supervisor:** $\nu_n = 0.8$, $\epsilon = 10^{-3}$ [8]; **LS combiner:** $\mathcal{L} = 500$ [9]; **Feedback output filter:** $\mu_{a,n} = 0.9$, $\epsilon = 10^{-2}$, $\gamma_n = 0.9$, $L_n = 400$.

The network learning analysis from Section V is illustrated in Fig. 3 (Left: network average MSD, simulations and theory (9); right: steady-state MSD per node), which shows the network (i) rejecting a highly connected biased node and (ii) exploiting an exceptional node with low degree—node 5 in Figs. 2a and 2b, respectively. The new protocol does fulfill the *desiderata* (Section III) in such challenging scenarios.

In Fig. 4, again the bad and good nodes have high and low degree, respectively, and AFs are zero-started with no node-level feedback ($L_n \to \infty$). Note that the standard diffusion is not universal by Def. 2, although the AN with the new combiner is. With node-level feedback ($L_n = 200$), Fig. 5 compares the new AN with the MSD combiner and the LS combiner in a scenario from [9]. The advantage is clear.

The scenario in Fig. 6 also compares (4) to the MSD combiner [8] and the LS combiner [9], and it is similar to the complex network from [9], in which there are two clusters of nodes with different performances. Here, however, the node statistics are not homogeneous and there is no difference in the AFs step sizes. The network used is the same as in [8] (Fig. 2d). Due to node-level feedback, the network under the new combiner displays a fast initial convergence with a steady-state error comparable to that under the MSD combiner. What is more, it has significantly lower complexity. Finally, the robustness issues of the LS combiner are clear from its learning curve, which is also observed in Fig. 5.

References

[1] C.G. Lopes and A.H. Sayed, “Incremental adaptive strategies over distributed networks,” *IEEE Trans. Signal Process.*, vol. 55[8], pp. 4064–4077, 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[7], pp. 4064–4077, 2008.

[3] 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[5], pp. 1865–1877, 2008.

[4] F.S. Cattivelli and A.H. Sayed, “Distributed detection over adaptive networks using diffusion adaptation,” *IEEE Trans. Signal Process.*, vol. 59[5], pp. 1917–1932, 2011.

[5] Sheng-Yuan Tu and A.H. Sayed, “Optimal combination rules for adaptation and learning over networks,” in *International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*. IEEE, 2011, pp. 317–320.

[6] A.H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z.J. Towfic, “Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior,” *IEEE Signal Process. Mag.*, vol. 30[3], pp. 155–171, 2013.

[7] A.H. Sayed, *Adaptation, Learning, and Optimization over Networks*, NOW Publishers, 2014.

[8] N. Takahashi, I. Yamada, and A.H. Sayed, “Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis,” *IEEE Trans. Signal Process.*, vol. 58[9], pp. 4795–4810, 2010.

[9] J. Fernandez-Bes, L.A. Azpicueta-Ruiz, M.T.M. Silva, and J. Arenas-Garcia, “A novel scheme for diffusion networks with least-squares adaptive combiners,” in *International Workshop on Machine Learning for Signal Processing*. IEEE, 2012, pp. 1–6.

[10] Chung-Kai Yu and A.H. Sayed, “A strategy for adjusting combination weights over adaptive networks,” in *International Conference in Acoustic, Speech, and Signal Processing*. IEEE, 2013, pp. 4579–4583.

[11] Songcen Xu, R.C. de Lamare, and H.V. Poor, “Adaptive link selection strategies for distributed estimation in diffusion wireless networks,” in *International Conference in Acoustic, Speech, and Signal Processing*. IEEE, 2013, pp. 5402–5405.

[12] N. Merhav and M. Feder, “Universal prediction,” *IEEE Trans. Inf. Theory*, vol. 44[6], pp. 2124–2147, 1998.

[13] L.F.O. Chamon, W.B. Lopes, and C.G. Lopes, “Combination of adaptive filters with coefficients feedback,” in *ICASSP*, 2012, pp. 3785–3788.

[14] X. Zhao and A.H. Sayed, “Performance limits of LMS-based adaptive networks,” in *International Conference in Acoustic, Speech, and Signal Processing*. IEEE, 2011, pp. 3768–3771.