ADAPTIVE PROCESSING OVER DISTRIBUTED NETWORKS

A. H. Sayed and C. G. Lopes

SUMMARY The article describes recent adaptive estimation algorithms over distributed networks. The algorithms rely on local collaborations and exploit the space-time structure of the data. Each node is allowed to communicate with its neighbors in order to exploit the spatial dimension, while it also evolves locally to account for the time dimension. Algorithms of the least-mean-squares and least-squares types are described. Both incremental and diffusion strategies are considered.

keywords: *Distributed processing, adaptive filtering, adaptive networks, incremental algorithm, diffusion, collaboration, cooperation, mean-square-error, energy conservation.*

1. INTRODUCTION

Distributed networks linking PCs, laptops, cell phones, sensors and actuators will form the backbone of future data communication and control networks. Applications will range from sensor networks to precision agriculture, environment monitoring, disaster relief management, smart spaces, target localization, as well as medical applications [1]–[4]. In all these cases, the distribution of the nodes in the field yields spatial diversity, which should be exploited alongside the temporal dimension in order to enhance the robustness of the processing tasks and improve the probability of signal and event detection [1].

Distributed processing deals with the extraction of information from data collected at nodes that are distributed over a geographic area. For example, each node in a network of nodes could collect noisy observations related to a certain parameter of interest. The nodes would then interact with each other in a certain manner, as dictated by the network topology, in order to arrive at an estimate of the parameter. The objective is to arrive at an estimate that is as accurate as the one that would be obtained if each node had access to the information across the entire network.

Obviously, the effectiveness of any distributed implementation will depend on the modes of cooperation that are allowed among the nodes. Figure 1 illustrates three such modes of cooperation.

In an incremental mode of cooperation, information flows in a sequential manner from one node to the adjacent node. This mode of operation requires a cyclic pattern of collaboration among the nodes, and it tends to require the least amount of communications and power [5]–[8], [15]. In a diffusion implementation, on the other hand, each node communicates with all its neighbors as dictated by the network topology. The amount of communication in this case is higher than in an incremental solution. Nevertheless, the nodes have access to more data from their neighbors. The communications in the diffusion implementation can be reduced by allowing each node to communicate only with a subset of its neighbors. In this mode of cooperation, the choice of which subset of neighbors to communicate with can be randomized according to some performance criterion.

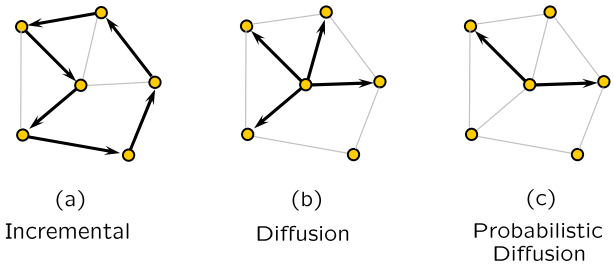


Fig. 1. Three modes of cooperation.

2. ADAPTIVE NETWORKS

In this work we describe distributed algorithms that enable a network of nodes to function as an adaptive entity following the works [8]–[13]. In order to clarify what we mean by an adaptive network, let us first review the structure of a traditional adaptive filter.

As is well known, and as shown in Fig. 2, an adaptive filter is generally a digital filter that changes its internal structure in response to an excitation and a reference signal. At each time instant, the filter compares its output to a reference signal and generates an error signal. The filter then adjusts its coefficients depending on whether the error is large or small. Thus, the key fact to note is that an adaptive filter responds in real-time to its data and to variations in the statistical properties of this data. We want to extend this ability to the network domain. By an adaptive network we therefore mean an inter-connected structure of adaptive nodes that is able to respond to data in real-time and to track

This work was supported in part by the National Science Foundation under award ECS-0601266. The authors are with the Electrical Engineering Department, University of California, Los Angeles, CA, 90095.

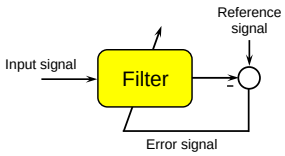


Fig. 2. An adaptive filter structure.

variations in the statistical properties of the data as well. As a result, in an adaptive network, whenever information arrives at a particular node, the information creates a ripple effect throughout the network and it influences the performance and behavior of the other nodes as dictated by the network topology — see the schematics in Fig. 3.

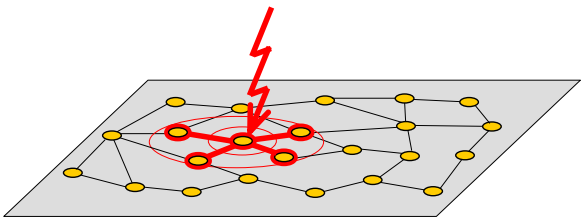


Fig. 3. A schematic representation of an adaptive network consisting of an interconnected system of adaptive nodes.

To illustrate the concept of an adaptive network, consider a collection of nodes and assume the network is required to estimate a certain parameter of interest – see Fig. 4. Each node collects local observations and at the same time interacts with its immediate neighbors. At every instant, the local observation is combined with information from the neighboring nodes in order to improve the estimate at the local node. By repeating this process of simultaneous observation and consultation, the nodes are constantly exhibiting updated estimates that respond to the observations in real time. In steady-state, after sufficient observations and cooperation, the nodes would converge to desired estimates of the unknown parameter.

We now describe several adaptation rules that exploit both temporal and spatial information and allow the nodes to estimate a parameter of interest by cooperating distributively.

2.1. Notation

In the remainder of the article we use boldface letters for random quantities and normal font for non-random (deterministic) quantities. We also use capital letters for matrices and small letters for vectors. For example, $\boldsymbol{d}$ is a random quantity and d is a realization or measurement for it, and R is a covariance matrix while w is a weight vector.

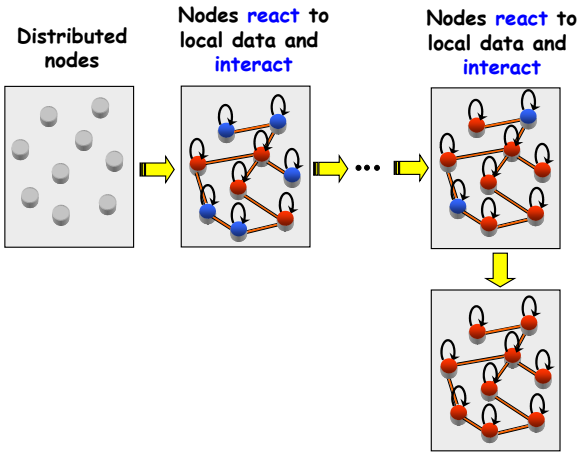


Fig. 4. An illustration of an adaptive network strategy.

The notation * denotes complex conjugation for scalars and complex-conjugate transposition for matrices.

To begin with, and following [10], consider a network with N nodes (see Fig. 5). Each node k has access to time-realizations $\{d_k(i), u_{k,i}\}$ of zero-mean spatial data $\{\boldsymbol{d}_k, \boldsymbol{u}_k\}$, $k = 1, \ldots, N$, where each $\boldsymbol{d}_k$ is a scalar measurement and each $\boldsymbol{u}_k$ is a $1 \times M$ row regression vector. We collect the regression and measurement data into two global matrices:

$$\boldsymbol{U} \triangleq \mathrm{col}\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N\} \qquad (N \times M) \tag{1}$$

$$\boldsymbol{d} \triangleq \mathrm{col}\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\} \qquad (N \times 1) \tag{2}$$

where the notation $\mathrm{col}\{\cdot\}$ denotes a column vector (or matrix) with the specified entries stacked on top of each other. These quantities collect the data across all N nodes. The objective is to estimate the $M \times 1$ vector w^o that solves

$$\min_{w} \; E\|\boldsymbol{d} - \boldsymbol{U}w\|^2 \tag{3}$$

where E is the expectation operator.

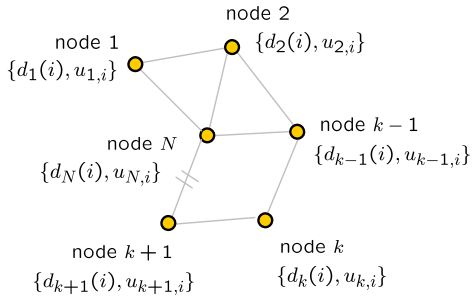


Fig. 5. A distributed network with N active nodes accessing space-time data.

The traditional iterative steepest-descent solution for determining w^o is given by

$$w_i = w_{i-1} + \mu \sum_{k=1}^{N} (R_{du,k} - R_{u,k} w_{i-1}) \tag{4}$$

where $\mu > 0$ is a step-size parameter, w_i is an estimate for w^o at iteration i, and

$$R_{u,k} \triangleq E\, \boldsymbol{u}_k^* \boldsymbol{u}_k \quad \text{and} \quad R_{du,k} \triangleq E\, \boldsymbol{d}_k \boldsymbol{u}_k^*$$

An equivalent description of (4) that is prune to a distributed implementation can be motivated as follows.

Let us define a *cycle* visiting every node only once. Let $\psi_k^{(i)}$ denote a *local estimate* of w^o at node k at time i, and assume node k has access to $\psi_{k-1}^{(i)}$, which is an estimate of w^o at node $k-1$ (see Fig. 6).

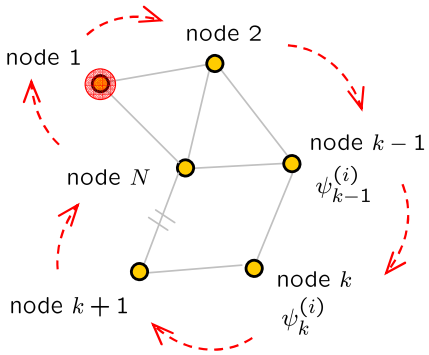


Fig. 6. A cycle covering nodes 1 through N.

If at each time instant i, we start with the initial condition $\psi_0^{(i)} = w_{i-1}$ and iterate cyclicly from node 1 to node N then, at the end of the procedure, the local estimate at node N will coincide with w_i from (4), i.e., $\psi_N^{(i)} = w_i$. In other words, the following implementation is equivalent to (4):

$$\begin{cases} \psi_0^{(i)} = w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu_k \left[R_{du,k} - R_{u,k} w_{i-1}\right]^* \\ \qquad k = 1, \ldots, N \\ w_i = \psi_N^{(i)} \end{cases} \tag{5}$$

Observe that the iteration for $\psi_k^{(i)}$ is over the spatial index k. Recursion (5) still requires the N nodes to have access to the global information w_{i-1}. In order to resolve this difficulty and arrive at a distributed implementation, we can resort to incremental techniques [5], [6], [8], [15]. If each node relies on the local estimate $\psi_{k-1}^{(i)}$ received from node $k-1$, as opposed to w_{i-1}, then an incremental version of algorithm (5) would result, namely,

$$\begin{cases} \psi_0^{(i)} = w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu_k \left[R_{du,k} - R_{u,k} \psi_{k-1}^{(i)}\right]^* \\ \qquad k = 1, \ldots, N \\ w_i = \psi_N^{(i)} \end{cases} \tag{6}$$

An adaptive implementation of (6) can be obtained by replacing the second-order moments $\{R_{du,k}, R_{u,k}\}$ by instantaneous approximations, say of the LMS type, as follows:

$$R_{du,k} \approx d_k(i) u_{k,i}^* \;, \qquad R_{u,k} \approx u_{k,i}^* u_{k,i} \tag{7}$$

by using data realizations $\{d_k(i), u_{k,i}\}$ at time i. The approximations (7) lead to a *distributed incremental LMS* algorithm of the form derived in [8]–[10]:

For each time $i \geq 0$, repeat:

$$\begin{cases} \psi_0^{(i)} = w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k u_{k,i}^* \left(d_k(i) - u_{k,i} \psi_{k-1}^{(i)}\right) \\ \qquad k = 1, \ldots, N \\ w_i = \psi_N^{(i)} \end{cases} \tag{8}$$

Obviously, different instantaneous approximations in (7) for the second-order moments lead to different adaptation rules. For example, an NLMS-type incremental solution would take the form:

For each time $i \geq 0$, repeat:

$$\begin{cases} \psi_0^{(i)} = w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{\mu_k u_{k,i}^*}{\epsilon + \|u_{k,i}\|^2} \left(d_k(i) - u_{k,i} \psi_{k-1}^{(i)}\right) \\ \qquad k = 1, \ldots, N \\ w_i = \psi_N^{(i)} \end{cases} \tag{9}$$

where $\epsilon > 0$ is a small parameter.

3. INCREMENTAL LEAST-SQUARES SOLUTIONS

We can also consider incremental RLS implementations as described in [12]. Thus assume again that each node k has access to regressor and measurement data $u_{k,i}$ and $d_k(i)$, $k = 1, \ldots, N$. At each time instant i, the entire network has access to space-time data

$$y_i = \begin{bmatrix} d_1(i) \\ d_2(i) \\ \vdots \\ d_N(i) \end{bmatrix} \text{ and } H_i = \begin{bmatrix} u_{1,i} \\ u_{2,i} \\ \vdots \\ u_{N,i} \end{bmatrix}. \tag{10}$$

Here y_i and H_i are snapshot matrices unveiling the network data status at time i. We can then formulate an exponentially weighted regularized least-squares (LS) problem [14],

where the weight vector estimate w_i is found by solving:

$$\min_{w} \left[\lambda^{i+1} w^* \Pi w + \left(\mathcal{Y}_i - \mathcal{H}_i w \right)^* \mathcal{W}_i \left(\mathcal{Y}_i - \mathcal{H}_i w \right) \right] \quad (11)$$

where $\Pi > 0$ is a regularization matrix, and the weighting matrix is given by

$$\mathcal{W}_i = \text{diag}\{\lambda^i \Gamma, \lambda^{i-1}\Gamma, \cdots, \lambda\Gamma, \Gamma\} \quad (12)$$

with a spatial weighting matrix

$$\Gamma = \text{diag}\{\gamma_1, \gamma_2, \cdots, \gamma_N\} \quad (13)$$

and a (time) forgetting factor

$$0 \ll \lambda \leq 1. \quad (14)$$

Moreover, $\mathcal{Y}_i$ and $\mathcal{H}_i$ collect all the data blocks available from the beginning of the observation period up to current time

$$\mathcal{Y}_i = \begin{bmatrix} y_0 \\ \hline y_1 \\ \hline \vdots \\ \hline y_i \end{bmatrix} \text{ and } \mathcal{H}_i = \begin{bmatrix} H_0 \\ \hline H_1 \\ \hline \vdots \\ \hline H_i \end{bmatrix}. \quad (15)$$

The global data matrices $\mathcal{Y}_i$ and $\mathcal{H}_i$ exhibit space-time structure, which naturally suggests a distributed solution. An algorithm that updates w_i recursively and in a distributed fashion is given by [12]:

$$\begin{array}{l} \psi_0^{(i)} \leftarrow w_{i-1}; \quad P_{0,i} \leftarrow \lambda^{-1} P_{i-1} \\ \text{for } k = 1 : N \\ \quad e_k(i) = d_k(i) - \boldsymbol{u}_{k,i} \psi_{k-1}^{(i)} \\ \quad \psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{P_{k-1,i}}{\gamma_k^{-1} + u_{k,i} P_{k-1,i} u_{k,i}^*} u_{k,i}^* e_k(i) \\ \quad P_{k,i} = P_{k-1,i} - \frac{P_{k-1,i} u_{k,i}^* u_{k,i} P_{k-1,i}}{\gamma_k^{-1} + u_{k,i} P_{k-1,i} u_{k,i}^*} \\ \text{end} \\ w_i \leftarrow \psi_N^{(i)}; \quad P_i \leftarrow P_{N,i} \, . \end{array} \quad (16)$$

where $P_{k,i}$ and P_i are $M \times M$ matrices updated as explained above. Note that the iterations are performed over the spatial index k. Therefore, a path is induced across the network, along which w_{i-1} is spatially updated by sequentially visiting every node once. Moreover, at each time i, the estimate $\psi_k^{(i)}$ at node k is the LS solution considering data blocks $\mathcal{Y}_{i-1}$ and $\mathcal{H}_{i-1}$ in addition to the data collected along the path. At the end of the cycle, $\psi_N^{(i)}$ will contain precisely the desired solution w_i. If we start from $i = 0$ with $w_{-1} = 0$ and $P_{-1} = \Pi^{-1}$ and repeatedly apply (16) taking into account sequentially all the data blocks up to time i then, by induction, $\psi_N^{(i)}$ (or w_i) will be the solution to the *global* LS problem (11). Figure 7 depicts the structure of the incremental implementation, in which both $\psi_k^{(i)}$ and $P_{k,i}$ are transmitted to the next node in the path.

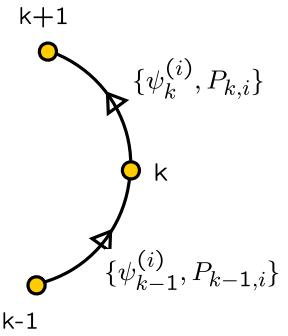


Fig. 7. The cooperation strategy of the distributed RLS algorithm (dRLS) described by (16).

A simplification that requires less communications while keeping the performance close to the exact implementation (especially for $\lambda \to 1$) can be obtained as follows. We allow collaboration for the estimates while keeping the matrices $P_{k,i}$ evolving locally and independent from the neighbor nodes. This approximation leads to the following algorithm [12]:

$$\begin{array}{l} \psi_0^{(i)} \leftarrow w_{i-1}; \quad P_{0,i} \leftarrow \lambda^{-1} P_{i-1} \\ \text{for } k = 1 : N \\ \quad e_k(i) = d_k(i) - u_{k,i} \psi_{k-1}^{(i)} \\ \quad \psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{P_{k,i-1}}{\gamma_k^{-1} + u_{k,i} P_{k,i-1} u_{k,i}^*} u_{k,i}^* e_k(i) \\ \quad P_{k,i} = P_{k,i-1} - \frac{P_{k,i-1} u_{k,i}^* u_{k,i} P_{k,i-1}}{\gamma_k^{-1} + u_{k,i} P_{k,i-1} u_{k,i}^*} \\ \text{end} \\ w_i \leftarrow \psi_N^{(i)}; \quad P_i \leftarrow P_{N,i} \, . \end{array} \quad (17)$$

Algorithm (17) iterates the estimates $\psi_k^{(i)}$ over space, while $P_{k,i}$ is iterated over time with local data only. As a consequence it requires transmission complexity $O(M)$ as opposed to $O(M^2)$ for (16). Figure 8 presents the algorithm's collaboration strategy, in which estimates are shared along the path and matrices $P_{k,i}$ evolve locally.

4. DIFFUSION LMS SOLUTION

When more communication resources are available, we may take advantage of the network connectivity and devise more sophisticated peer-to-peer cooperation rules. We describe two such LMS-based diffusion protocols here following [9]–[11] – see Figs. 9 and 10; extensions to RLS-based diffusion schemes appear in [13].

The neighborhood of a node k at time $i-1$ is denoted by $\mathcal{N}_k(i-1)$ and is defined as the set of nodes directly connected to it, including itself. Each individual node k consults peer nodes from its neighborhood and combines their past estimates

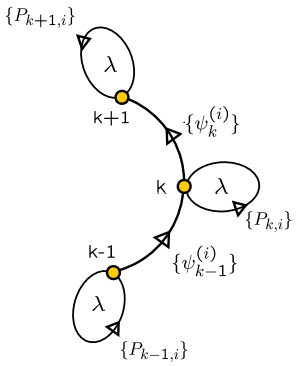


Fig. 8. The cooperation strategy of the low communications distributed RLS algorithm (LC-dRLS).

$$\{\psi_\ell^{(i-1)}; \ell \in \mathcal{N}_k(i-1)\}$$

with its own past estimate $\psi_k^{(i-1)}$. The node generates an aggregate estimate $\phi_k^{(i-1)}$ and feeds it into its local adaptive filter. The strategy can be expressed as follows for LMS-type recursions:

$$\begin{aligned} \phi_k^{(i-1)} &= f_k\left(\psi_\ell^{(i-1)}; \ell \in \mathcal{N}_k(i-1)\right) \\ \psi_k^{(i)} &= \phi_k^{(i-1)} + \mu_k u_{k,i}^* \left(d_k(i) - u_{k,i}\phi_k^{(i-1)}\right) \end{aligned} \quad (18)$$

for some local combiner $f_k(\cdot)$. The combiners $f_k(\cdot)$ can be nonlinear or even time-variant, to reflect, for instance, changing topologies or to respond to non-stationary environments.

One simple combining rule is to average the local and neighbors' previous estimates, i.e.,

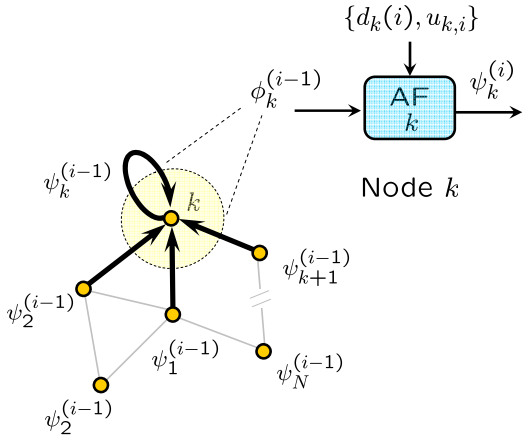


Fig. 9. A network with diffusion cooperation strategy.

$$\begin{aligned} \phi_k^{(i-1)} &= \sum_{\ell \in \mathcal{N}_k} c(k,\ell)\, \psi_\ell^{(i-1)} \\ \psi_k^{(i)} &= \phi_k^{(i-1)} + \mu_k u_{k,i}^* \left(d_k(i) - u_{k,i}\phi_k^{(i-1)}\right) \end{aligned} \quad (19)$$

where $c(k,\ell) = 1/\deg(k)$, with $\deg(k)$ denoting the degree of node k (number of incident links at this node, including itself). This scheme exploits network connectivity more fully, leading to more robust algorithms. If links or nodes eventually fail, the adaptive network can react by relying on the remaining topology. Note that the adaptive network would work even for non-connected graphs, relying on the individual agents. Furthermore, since more information is aggregated in the local adaptive filter updates, individual nodes can attain better learning behavior when compared to the non-cooperative case, provided that the combiners $f_k(\cdot)$ are well designed.

Another combination rule is motivated by the analysis results of [16]. The rule allows the network to assign convex combination weights to the local estimate and the aggregate estimate. Moreover, the weights can be adjusted adaptively so that the network can respond to node conditions and assign smaller weights to nodes that are subject to higher noise levels. One particular implementation is as follows in terms of combination weights $\{\alpha_k\}$:

$$\begin{aligned} \alpha_k &= \frac{1}{1+|\exp(-z_k)|^2} \\ \overline{\psi}_k^{(i-1)} &= \sum_{\ell \in \mathcal{N}_k/k} c_k(\ell)\psi_\ell^{(i-1)} \\ \phi_k^{(i-1)} &= \alpha_k \psi_k^{(i-1)} + (1-\alpha_k)\,\overline{\psi}_k^{(i-1)} \\ e_k(i) &= d_k(i) - u_{k,i}\phi_k^{(i-1)} \\ \psi_k^{(i)} &= \phi_k^{(i-1)} + \mu\, u_{k,i}^* e_k(i) \\ h_k &= \left(\psi_k^{(i-1)} - \overline{\psi}_k^{(i-1)}\right) e_k(i)\alpha_k\,(1-\alpha_k) \\ z_k &= z_k + \frac{\mu_z u_{k,i}}{\|u_{k,i}\|^4} h_k \end{aligned}$$

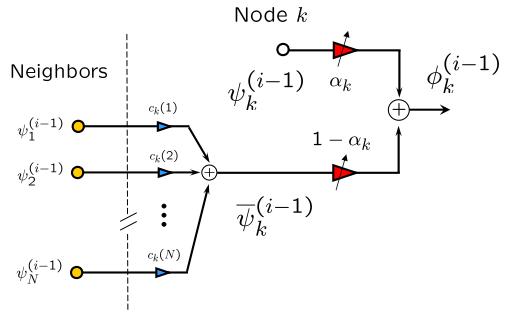


Fig. 10. A network with an *adaptive* diffusion cooperation strategy.

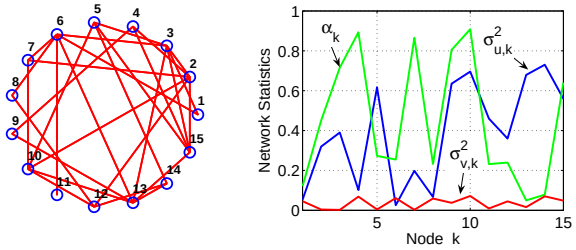


Fig. 11. Network topology and statistical profile.

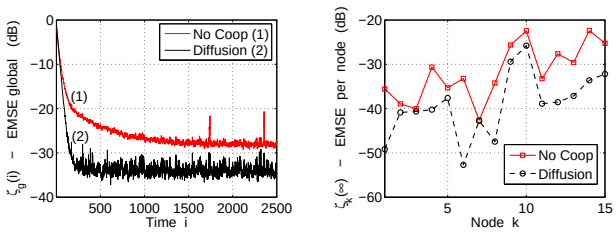


Fig. 12. Transient global EMSE and steady-state EMSE per node.

In order to illustrate the adaptive network performance, we present a simulation example in Figs. 11 and 12. Fig. 11 depicts the network topology and the network statistical profile. The regressors follow a first order Markov process with power $\sigma^2_{u,k}$ and correlation index α_k. The background noise power is denoted by $\sigma^2_{v,k}$. Note how the averaging diffusion protocol (19) outperforms the non-cooperative case (where each node runs an individual filter). Fig. 12, left plot, presents the average global excess mean-square error (EMSE), defined as

$$\zeta_g(i) = \frac{1}{N}\sum_{k=1}^{N}\zeta_k(i)$$

where the individual EMSE at node k is depicted in the right plot and is defined as (see also Sec. 5):

$$\zeta_k(i) E\left|\boldsymbol{u}_{k,i}\left(w^o - \boldsymbol{\psi}_k^{(i-1)}\right)\right|^2$$

5. MEAN-SQUARE PERFORMANCE

The prior algorithms exploit both the spatial *and* temporal dimensions of the data. In [8], [10], [12] we studied the mean-square performance of incremental LMS (8) and low-communications dRLS (17) using energy conservation arguments [14]; performance analysis for diffusion schemes of the LMS and RLS types appear in [11], [13] and are not reviewed here for brevity.

The analysis in the incremental case [8], [10], [12] relied on the following assumptions for the random variables $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$:

1. The unknown vector w^o relates $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ as
$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i) \tag{20}$$
where $\boldsymbol{v}_k(i)$ is some white noise sequence with variance $\sigma^2_{v,k}$ and independent of $\{\boldsymbol{d}_l(j), \boldsymbol{u}_{l,j}\}$ for all l, j.
2. $\boldsymbol{u}_{k,i}$ is independent of $\boldsymbol{u}_{l,i}$ for $k \neq l$ (spatial independence).
3. For every k, the sequence $\{\boldsymbol{u}_{k,i}\}$ is independent over time (time independence).
4. The regressors $\{\boldsymbol{u}_{k,i}\}$ arise from a source with circular Gaussian distribution with covariance matrix $R_{u,k}$.

Define the error signals:

$$\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} \triangleq w^o - \boldsymbol{\psi}_{k-1}^{(i)} \tag{21}$$

$$\boldsymbol{e}_{a,k}(i) \triangleq \boldsymbol{u}_{k,i}\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} \tag{22}$$

$$\boldsymbol{e}_k(i) \triangleq \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k-1}^{(i)} \tag{23}$$

where (21) denotes the weight-error vector, (22) defines the *a priori* local error, and (23) defines the output error. Mean-square analysis is interested in evaluating, in steady-state and for each node k, the mean-square deviation (MSD), the excess mean-square error (EMSE), and the mean-square error (MSE). These quantities are defined as

$$\eta_k \triangleq E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|^2 \quad \text{(MSD)} \tag{24}$$

$$\zeta_k \triangleq E|\boldsymbol{e}_{a,k}(\infty)|^2 \quad \text{(EMSE)} \tag{25}$$

$$\xi_k \triangleq E|\boldsymbol{e}_k(\infty)|^2 = \zeta_k + \sigma^2_{v,k} \quad \text{(MSE)} \tag{26}$$

Unlike traditional adaptive filter performance, in the distributed case, the weight error vectors converge to a spatial error profile and stabilize at individual error energy levels, i.e.,

$$E\|\widetilde{\boldsymbol{\psi}}_k^{(i)}\|^2 \to \varepsilon_k\,, \quad \text{as} \quad i \to \infty$$

with a value ε_k that is possibly different for each node k. Moreover, due to cooperation, the nodes are interconnected. This fact makes the analysis more challenging.

5.1. Incremental LMS

Introduce the eigen-decomposition $R_{u,k} = U_k\Lambda_k U_k^*$, where U_k is unitary and Λ_k is a diagonal matrix with the eigenvalues of $R_{u,k}$. Let

$$\overline{F}_k I - 2\mu_k\Lambda_k + \gamma\mu_k^2\Lambda_k^2 + \mu_k^2 b_k b_k^T$$

where $b_k = \mathrm{diag}\{\Lambda_k\}$ is a column vector containing the diagonal entries of Λ_k, $\gamma = 1$ for circular complex regressors, and $\gamma = 2$ for real regressors. Then it was shown in [8], [10] that the MSD and EMSE are given by

$$\eta_k = a_k\left(I - \Pi_{k,1}\right)^{-1} q \tag{27}$$

$$\zeta_k = a_k\left(I - \Pi_{k,1}\right)^{-1} b_k \tag{28}$$

where a_k is a row vector, $\Pi_{k,1}$ is a matrix, and q is a column vector:

$$\begin{aligned} q &= \mathrm{col}\{1, 1, \cdots, 1\} \\ \Pi_{k,l} &\triangleq \overline{F}_{k+l-1}\overline{F}_{k+l}\cdots\overline{F}_N\overline{F}_1\cdots\overline{F}_{k-1} \\ & \quad l = 1, \ldots, N \\ a_k &\triangleq g_k\Pi_{k,2} + g_{k+1}\Pi_{k,3} + \cdots + g_{k-2}\Pi_{k,N} + g_{k-1} \\ g_k &= \mu_k^2\sigma_{v,k}^2 b_k^T \end{aligned}$$

Analyzing these results, we find that every node individually experiences the influence of the entire network, with some emphasis given to the local statistics, as represented by b_k and $\sigma_{v,k}^2$.

For sufficiently small step-sizes, we can use the approximation

$$\overline{F}_k \approx I - 2\mu_k\Lambda_k$$

i.e., $\overline{F}_k$ becomes a diagonal matrix. As a result,

$$I - \Pi \approx 2\mu_1\Lambda_1 + 2\mu_2\Lambda_2 + \cdots + 2\mu_N\Lambda_N \triangleq D$$

and

$$a_k \approx \mu_1^2\sigma_{v,1}^2 b_1^T + \cdots + \mu_N^2\sigma_{v,N}^2 b_N^T \triangleq a$$

where D is a diagonal matrix and a is a row vector. Then

$$\eta_k \approx aD^{-1}q \quad \text{(MSD)}$$

which reveals an interesting behavior. Despite the simple cooperation mode in the incremental strategy, for small step-sizes, there is an equalization effect on the MSD throughout the network. In the same vein, the EMSE for small step-sizes is given by

$$\zeta_k \approx aD^{-1}b_k \quad \text{(EMSE)}$$

5.2. Incremental Least-Squares

In a similar vein, the mean-square-error performance of the low-communication incremental least-squares solution (17) was studied in [12]. Let

$$\overline{F}_k = (1 - 2\beta_k + \delta\beta_k^2)I + \beta_k^2 b_k r_k^T$$

where $\delta = 1$ for complex signals and $\delta = 2$ for real signals, $r_k = \mathrm{diag}\{\Lambda_k^{-1}\}$, and β_k is given by

$$\beta_k = \begin{cases} \frac{1-\lambda}{\gamma_k^{-1}}, & \text{for } \lambda \to 1 \\ \\ \frac{1-\lambda}{\gamma_k^{-1}\lambda + (1-\lambda)M}, & \text{for smaller } \lambda\,. \end{cases}$$

in terms of the forgetting factor λ. Then it was shown in [8], [10] that the MSD and EMSE are again given by

$$\eta_k = a_k\left(I - \Pi_{k,1}\right)^{-1} q \tag{29}$$

$$\zeta_k = a_k\left(I - \Pi_{k,1}\right)^{-1} b_k \tag{30}$$

where now $g_k = \beta_k^2\sigma_{v,k}^2 r_k^T$.

6. CONCLUDING REMARKS

We have described several distributed and cooperative algorithms that endow distributed networks with learning abilities. They address distributed estimation problems that arise in a variety of applications, such as environment monitoring, target localization and potential sensor network problems [1].

For low energy profile implementations, the incremental LMS algorithm performs well. As the available resources increase, more sophisticated learning rules, such as recursive least-squares, can help speed network convergence. Still, with the increase in the size of networks, setting a cycle may not be a trivial task. In order to alleviate topology constraints and exploit more fully network connectivity, diffusion protocols can be developed. They give rise to peer-to-peer estimation protocols that exploit spatial diversity, improve robustness, and benefit the network in terms of estimation performance in comparison to the non-cooperative case. Diffusion protocols may also be extended to the RLS case and other types of adaptive updates.

7. REFERENCES

[1] D. Estrin, G. Pottie and M. Srivastava, "Intrumenting the world with wireless sensor setworks," *Proc. ICASSP*, pp. 2033-2036, Salt Lake City, UT, May 2001.

[2] D. Li, K. D. Wong, Y. H. Hu and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, March 2002.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, issue 8, pp. 102–114, August, 2002.

[4] D. Culler, D. Estrin and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.

[5] D. Bertsekas, "A new class of incremental gradient methods for least squares problems," newblock *SIAM J. Optim.*, vol.7, no. 4, pp. 913-926, Nov. 1997.

[6] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol.23, no.4, pp. 798–808, April 2005.

[7] A. Nedic and D. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.

[8] C. G. Lopes and A. H. Sayed, "Distributed adaptive incremental strategies: formulation and performance analysis," *Proc. ICASSP*, vol. 3, pp. 584–587, Toulouse, France, May 2006.

[9] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," *Proc. Adaptive Sensor Array Processing Workshop*, MIT Lincoln Lab., Lexington, MA, June 2006.

[10] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," to appear in *IEEE Trans. Signal Processing*.

[11] C. G. Lopes and A. H. Sayed, "Diffusion least-mean-squares over adaptive networks," *Proc. ICASSP*, vol. III, pp. 917–920, Honolulu, Hawaii, April 2007.

[12] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," *Proc. Asilomar Conference on Signals, Systems and Computers*, pp. 233–237, Pacific Grove, CA, October 2006.

[13] F. Cattivelli, C. G. Lopes, and A. H. Sayed, "A diffusion RLS scheme for distributed estimation over adaptive networks," *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Helsinki, Finland, June 2007.

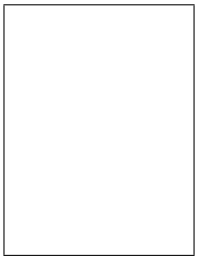
[14] A. H. Sayed, *Fundamentals of Adaptive Filtering,* Wiley, NJ, 2003.

[15] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56-69, July 2006.

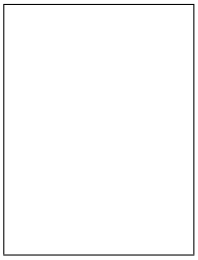
[16] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Processing*, vol. 54, no. 3, pp. 1078–1090, March 2006.

[17] B. T. Poljak and Y. Z. Tsypkin, "Pseudogradient adaptation and training algorithms" *Automatic and Remote Control*, vol. 12, pp. 83–94, 1978.

[18] M. G. Rabbat and R. D. Nowak, "Decentralized source localization and tracking," *Proc. ICASSP*, vol. III, pp. 921-924, Montreal, Canada, May 2004.



Ali H. Sayed is Professor and Chairman of electrical engineering at the University of California, Los Angeles (UCLA). He is also the Principal Investigator of the UCLA Adaptive Systems Laboratory (www.ee.ucla.edu/asl). He has more than 270 journal and conference publications, is the author of Fundamentals of Adaptive Filtering (New York: Wiley, 2003), is the coauthor of the research monograph Indefinite Quadratic Estimation and Control (Philadelphia, PA: SIAM, 1999) and of the graduate-level textbook Linear Estimation (Englewood Cliffs, NJ: Prentice-Hall, 2000). He is also coeditor of the volume Fast Reliable Algorithms for Matrices with Structure (Philadelphia, PA: SIAM, 1999). He has contributed several articles to engineering and mathematical encyclopedias and handbooks and has served on the program committees of several international meetings. His research interests span several areas, including adaptive and statistical signal processing, filtering and estimation theories, signal processing for communications, interplays between signal processing and control methodologies, system theory, and fast algorithms for large-scale problems. Dr. Sayed is a Fellow of IEEE and is the recipient of the 1996 IEEE Donald G. Fink Award. He received the Best Paper Award from the IEEE Signal Processing Society in 2002, the Kuwait Prize in Basic Science in 2003, the Frederick E. Terman Award in 2005, a Young Author Best Paper Award in 2005 from the IEEE Signal Processing Society, and two Best Student Paper awards at international meetings. He is also a member of the technical committees on Signal Processing Theory and Methods (SPTM) and on Signal Processing for Communications (SPCOM), both of the IEEE Signal Processing Society. He has served as Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2003-2005) and is now serving as Editor-in-Chief of the EURASIP Journal on Applied Signal Processing. He is serving as General Chairman of ICASSP 2008 and sits on the Board of Governors of the IEEE Signal Processing Society.



Cassio G. Lopes