Multitask Diffusion Adaptation Over Networks

Jie Chen, *Member, IEEE*, Cédric Richard, *Senior Member, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

***Abstract*—Adaptive networks are suitable for decentralized inference tasks. Recent works have intensively studied distributed optimization problems in the case where the nodes have to estimate a single optimum parameter vector collaboratively. However, there are many important applications that are multitask-oriented in the sense that there are multiple optimum parameter vectors to be inferred simultaneously, in a collaborative manner, over the area covered by the network. In this paper, we employ diffusion strategies to develop distributed algorithms that address multitask problems by minimizing an appropriate mean-square error criterion with ℓ_2-regularization. The stability and performance of the algorithm in the mean and mean-square error sense are analyzed. Simulations are conducted to verify the theoretical findings, and to illustrate how the distributed strategy can be used in several useful applications related to target localization and hyperspectral data unmixing.**

***Index Terms*—Asymmetric regularization, collaborative processing, data unmixing, diffusion strategy, distributed optimization, multitask learning, target localization.**

I. Introduction

DISTRIBUTED adaptation over networks has emerged as an attractive and challenging research area with the advent of multi-agent (wireless or wireline) networks. Accessible overviews of recent results in the field can be found in [2]–[4]. In adaptive networks, the interconnected nodes continually learn and adapt, as well as perform assigned tasks such as parameter estimation from observations collected by the dispersed agents.

There are several useful distributed strategies for sequential data processing over networks including consensus strategies [5]–[10], incremental strategies [11]–[15], and diffusion strategies [2], [3], [16]–[19]. Incremental techniques require the determination of a cyclic path that runs across the nodes, which is generally a challenging (NP-hard) task to perform. Besides, incremental solutions can be problematic for adaptation over networks because they are sensitive to link failures. On the other hand, diffusion strategies are attractive since they are scalable, robust, and enable continuous adaptation and learning. In addition, for data processing over adaptive networks, diffusion strategies have been shown to have superior stability and performance ranges [20] than consensus-based implementations. Consequently, we focus on diffusion-type implementations in the sequel. The diffusion LMS strategy was proposed and studied in [16], [17]. Its performance was further examined under various scenarios and the algorithms applied to a variety of inference and estimation problems in [18], [19], [21]–[30].

An inspection of the existing literature on distributed algorithms shows that most works focus primarily, though not exclusively [31]–[33], on the case where the nodes have to estimate a single optimum parameter vector collaboratively. We shall refer to problems of this type as *single-task* problems. However, many problems of interest happen to be *multitask*-oriented in the sense that there are multiple optimum parameter vectors to be inferred simultaneously and in a collaborative manner. The multitask learning problem is relevant in several machine learning formulations and has found applications in web page categorization [34], web-search ranking [35], and disease progression modeling [36], among other areas. Clearly, this concept is also relevant in the context of distributed estimation and adaptation over networks. Initial investigations along these lines for the traditional diffusion strategy appear in [32], [37]. In this article, we consider the general situation where there are connected clusters of nodes, and each cluster has a parameter vector to estimate. The estimation still needs to be performed cooperatively across the network because the data across the clusters may be correlated and, therefore, cooperation across clusters can be beneficial. The aim of this paper is to solve this general multitask estimation problem with distributed strategies based on diffusion adaptation and information exchange between neighboring agents only, and to analyze their performance in terms of mean-square error and convergence rate.

There are at least three main contributions in this work. To begin with real-time adaptation and learning from streaming data is a key aspect of the proposed strategies, which differentiates them from traditional multi-task formulations in the machine learning literature where solutions tend to be application-specific and rely heavily on offline or batch implementations. Second, the available literature on distributed estimation over networks, including studies on incremental strategies, consensus strategies and diffusion strategies, focuses largely on estimating a single parameter vector by the entire network. In this paper, we extend this setting to the problem of estimating multiple parameter vectors by interconnected agents. Third, we introduce a game-theoretic formulation for the multi-task scenario whereas [38] focuses on the single-task case. Moreover, our framework endows the network with the ability to promote

Manuscript received April 10, 2014; accepted June 05, 2014. Date of publication June 27, 2014; date of current version July 18, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tongtong Li. This work was supported in part by the Agence Nationale pour la Recherche, France (Hypanema project: ANR-12-BS03; Odissee project: ANR-13-ASTR-0030). The work of A. H. Sayed was supported in part by the NSF under Grant CCF-1011918. This work appeared in part in the *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), Florence, Italy, May 2014, pp. 5542–5528 [1]. An extended report is also available at http://arxiv.org/abs/1311.4894.

J. Chen and C. Richard are with the Université de Nice Sophia-Antipolis, UMR CNRS 7293, Observatoire de la Côte d'Azur, Laboratoire Lagrange, Parc Valrose, 06102 Nice, France (e-mail: jie.chen@unice.fr; cedric.richard@unice.fr).

A. H. Sayed is with the Department of Electrical Engineering, University of California, Los Angeles, CA 9005-1594 USA (email: sayed@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2014.2333560

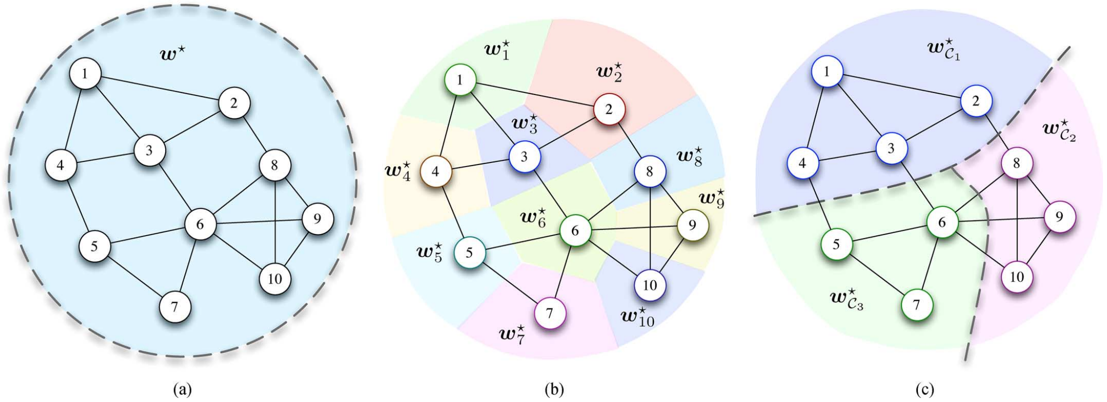


Fig. 1. Three types of networks. Through direct links, nodes can communicate with each other in one hop. The single-task and multitask networks can be viewed as special cases of the clustered multitask network. (a) Single-task network. (a) Single-task network. (c) Clustered multitask network.

asymmetric collaboration between nodes via proper selection of regularization parameters. We carry out a detailed performance analysis and derive expressions that explain the effect of various design parameters on network performance. We remark that the multi-task formulation of this work can be extended to other useful scenarios, such as applications involving Kernel-based learning along the lines studied in [39]–[46].

II. Network Models and Multitask Learning

We start with a summary of main symbols and notation:

x	Normal font denotes scalars.
$\boldsymbol{x}$	Boldface small letters denote vectors. All vectors are column vectors.
$\boldsymbol{R}$	Boldface capital letters denote matrices.
$(\cdot)^\top$	Matrix transpose.
$\boldsymbol{I}_N$	Identity matrix of size $N \times N$.
$\mathcal{N}_k$	The index set of nodes that are in the neighborhood of node k, including k.
$\mathcal{N}_k^-$	The index set of nodes that are in the neighborhood of node k, excluding k.
$\mathcal{C}_i$	Cluster i, i.e., index set of nodes in the i-th cluster.
$\mathcal{C}(k)$	The cluster to which node k belongs, i.e., $\mathcal{C}(k) = \{\mathcal{C}_i : k \in \mathcal{C}_i\}$.
$J(\cdot)$, $\overline{J}(\cdot)$	Cost functions without/with regularization.
$\boldsymbol{w}^\star$, $\boldsymbol{w}^o$	Optimum parameter vectors without/with regularization.
$\boldsymbol{w}_k^\star$, $\boldsymbol{w}_{\mathcal{C}_i}^\star$, $\boldsymbol{w}_{\mathcal{C}(k)}^\star$	Optimum parameter vectors at node k, at cluster $\mathcal{C}_i$, and at cluster $\mathcal{C}(k)$.

We consider a connected network consisting of N nodes. The problem is to estimate an $L \times 1$ unknown vector at each node k from collected measurements. Node k has access to temporal measurement sequences $\{d_k(n), \boldsymbol{x}_k(n)\}$, with $d_k(n)$ denoting a scalar zero-mean reference signal, and $\boldsymbol{x}_k(n)$ denoting an $L \times 1$ regression vector with a positive-definite covariance matrix, $\boldsymbol{R}_{x,k} = E\{\boldsymbol{x}_k(n)\boldsymbol{x}_k^\top(n)\} > 0$. The data at node k are assumed to be related via the linear regression model:

$$d_k(n) = \boldsymbol{x}_k^\top(n)\,\boldsymbol{w}_k^\star + z_k(n) \tag{1}$$

where $\boldsymbol{w}_k^\star$ is an unknown parameter vector at node k, and $z_k(n)$ is a zero-mean white noise that is independent of any other signal and has variance $\sigma_{z,k}^2$. Considering the number of parameter vectors to estimate, which we shall refer to as the number of tasks, the distributed learning problem can be single-task or multi-task oriented. We therefore distinguish among the following three types of networks, as illustrated by Fig. 1, depending on how the parameter vectors $\boldsymbol{w}_k^\star$ across the nodes are related:

- *Single-task networks*: All nodes have to estimate the same parameter vector $\boldsymbol{w}^\star$. That is, in this case we have that

$$\boldsymbol{w}_k^\star = \boldsymbol{w}^\star, \quad \forall k \in \{1, \ldots, N\} \tag{2}$$

- *Multitask networks*: Each node k has to determine its own optimum parameter vector, $\boldsymbol{w}_k^\star$. However, it is assumed that similarities and relationships exist among the parameters of neighboring nodes, which we denote by writing

$$\boldsymbol{w}_k^\star \sim \boldsymbol{w}_\ell^\star, \quad \text{if} \quad \ell \in \mathcal{N}_k \tag{3}$$

The sign $\sim$ represents a similarity relationship in some sense, and its meaning will become clear soon once we introduce expressions (8) and (9) further ahead. There are many situations in practice where the objective parameters are not identical across clusters but have inherent relationships. It is therefore beneficial to exploit these relationships to enhance performance. A number of application problems can be addressed using this model. For instance, consider an image sensor array and the problem of image restoration. In this case, links in Fig. 1(b) can represent neighboring relationships between adjacent pixels. We will consider this application in greater detail in the simulation section. Promoting similarity relationships can be performed in several ways. Within the area of machine learning, for instance, this has been performed through mean regularization [47], low rank regularization [48], or clustered regularization [49]. In this paper, in the spirit of

[47], we focus on promoting the similarity of objective parameter vectors via their distance to each other.

- *Clustered multitask networks*: Nodes are grouped into Q clusters, and there is one task per cluster. The optimum parameter vectors are only constrained to be equal within each cluster, but similarities between neighboring clusters are allowed to exist, namely,

$$\boldsymbol{w}_k^\star = \boldsymbol{w}_{\mathcal{C}_q}^\star, \quad \text{whenever } k \in \mathcal{C}_q \tag{4}$$

$$\boldsymbol{w}_{\mathcal{C}_p}^\star \sim \boldsymbol{w}_{\mathcal{C}_q}^\star, \quad \text{if } \mathcal{C}_p,\, \mathcal{C}_q \text{ are connected} \tag{5}$$

  where p and q denote two cluster indexes. We say that two clusters $\mathcal{C}_p$ and $\mathcal{C}_q$ are connected if there exists at least one edge linking a node from one cluster to a node in the other cluster.

One can observe that the single-task and multitask networks are particular cases of the clustered multitask network. In the case where all nodes are clustered together, the clustered multitask network reduces to the single-task network. On the other hand, in the case where each cluster only involves one node, the clustered multitask network becomes a multitask network. Building on the literature on diffusion strategies for single-task networks, we shall now generalize its use and analysis for distributed learning over clustered multitask networks. The results will be applicable to multitask networks by setting the number of clusters equal to the number of nodes.

III. Problem Formulation

A. Global Cost Function and Optimization

Clustered multitask networks require that nodes that are grouped in the same cluster estimate the same coefficient vector. Thus, consider the cluster $\mathcal{C}(k)$ to which node k belongs. A local cost function, $J_k(\boldsymbol{w}_{\mathcal{C}(k)})$, is associated with node k and it is assumed to be strongly convex and second-order differentiable, an example of which is the mean-square error criterion defined by

$$J_k(\boldsymbol{w}_{\mathcal{C}(k)}) = E\left\{\left|d_k(n) - \boldsymbol{x}_k^\top(n)\,\boldsymbol{w}_{\mathcal{C}(k)}\right|^2\right\}. \tag{6}$$

Relationships between tasks is a mutual information that deserves attention because it can improve the estimation accuracy. A diversity of regularizers can be considered to exploit this additional information. Promoting similarities among estimated parameter vectors via their distance to each other is simple but effective in many applications [47]. Two examples will be presented in later sections. For this purpose, we introduce the squared Euclidean distance as a possible regularizer, namely,

$$\Delta(\boldsymbol{w}_{\mathcal{C}(k)}, \boldsymbol{w}_{\mathcal{C}(\ell)}) = \|\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|^2. \tag{7}$$

Combining (6) and (7) yields the following regularized problem $\mathcal{P}_1$ at the level of the entire network:

$$(\mathcal{P}_1): \quad \min_{\boldsymbol{w}_{\mathcal{C}_1},\ldots,\boldsymbol{w}_{\mathcal{C}_Q}} \overline{J^{\text{glob}}}(\boldsymbol{w}_{\mathcal{C}_1},\ldots,\boldsymbol{w}_{\mathcal{C}_Q}),$$

where

$$\overline{J^{\text{glob}}}(\boldsymbol{w}_{\mathcal{C}_1},\ldots,\boldsymbol{w}_{\mathcal{C}_Q}) = \sum_{k=1}^{N} E\left\{\left|d_k(n) - \boldsymbol{x}_k^\top(n)\,\boldsymbol{w}_{\mathcal{C}(k)}\right|^2\right\} + \eta \sum_{k=1}^{N} \sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} \rho_{k\ell}\,\|\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|^2, \tag{8}$$

where $\boldsymbol{w}_{\mathcal{C}_i}$ is the parameter vector associated with cluster $\mathcal{C}_i$, $\eta > 0$ is a regularization parameter, and the symbol $\setminus$ is the set difference. The second term on the right-hand-side of expression (8) promotes similarities between the $\boldsymbol{w}_{\mathcal{C}_i}$ of neighboring clusters, with strength parameter η.

Observe from the right-most term in (8) that the regularization strength between two clusters is directly related to the number of edges that connect them. The non-negative coefficients $\rho_{k\ell}$ aim at adjusting the regularization strength but they do not necessarily enforce symmetry. That is, we do not require $\rho_{k\ell} = \rho_{\ell k}$ even though the regularization term $\|\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|^2$ is symmetric with respect to the weight vectors $\boldsymbol{w}_{\mathcal{C}(k)}$ and $\boldsymbol{w}_{\mathcal{C}(\ell)}$; this term will be weighted by the sum $\rho_{k\ell} + \rho_{\ell k}$ due to the summation over the N nodes. Consequently, problem formulation $\mathcal{P}_1$ inevitably leads to symmetric regularization despite the fact that $\rho_{k\ell} \neq \rho_{\ell k}$. However, we would like the design problem to benefit from the additional flexibility that is afforded by the use of asymmetric regularization coefficients. This is because asymmetry allows clusters to scale their desire for closer similarity with their neighbors differently. For example, asymmetric regularization would allow cluster $\mathcal{C}_k$ to promote similarities with cluster $\mathcal{C}_\ell$ while cluster $\mathcal{C}_\ell$ may be less inclined towards promoting similarities with $\mathcal{C}_k$. In order to enable this possiblity, we consider an alternative problem formulation $\mathcal{P}_2$ defined in terms of Q Nash equilibrium problems as follows:

$$(\mathcal{P}_2): \quad \min_{\boldsymbol{w}_{\mathcal{C}_i}} \overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i}), \qquad \text{for } i = 1,\ldots,Q$$

where

$$\overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i}) = \sum_{k\in\mathcal{C}_i} E\left\{\left|d_k(n) - \boldsymbol{x}_k^\top(n)\,\boldsymbol{w}_{\mathcal{C}(k)}\right|^2\right\} + \eta \sum_{k\in\mathcal{C}_i} \sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}_i} \rho_{k\ell}\,\|\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|^2 \tag{9}$$

where each cluster $\mathcal{C}_i$ estimates $\boldsymbol{w}_{\mathcal{C}_i}$ by minimizing $\overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i})$. Note that we have kept the notation $\boldsymbol{w}_{\mathcal{C}(k)}$ to make the role of the regularization term clearer, even though in formulation (9) we have $\boldsymbol{w}_{\mathcal{C}(k)} = \boldsymbol{w}_{\mathcal{C}_i}$ since k in $\mathcal{C}_i$. In (9), the notation $\boldsymbol{w}_{-\mathcal{C}_i}$ denotes the collection of weight vectors estimated by the other clusters. The Nash equilibrium of $\mathcal{P}_2$ satisfies the condition [50]:

$$\boldsymbol{w}_{\mathcal{C}_i}^o = \arg\min_{\boldsymbol{w}_{\mathcal{C}_i}} \overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i}^o) \tag{10}$$

for $i = 1,\ldots,Q$, where the notation $\boldsymbol{w}_{-\mathcal{C}_i}^o$ denotes the collection of the Nash equilibria by the other clusters. Problem $\mathcal{P}_2$ has the following properties:

1) An equilibrium exists for $\mathcal{P}_2$ since $\overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i})$ is convex with respect to $\boldsymbol{w}_{\mathcal{C}_i}$ for all i.

2) The equilibrium for problem $\mathcal{P}_2$ is unique since $\{\overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i})\}_{i=1}^{Q}$ satisfies the diagonal strict convexity property[1].
3) Problems $\mathcal{P}_1$ and $\mathcal{P}_2$ have the same solution by setting the value of $\rho_{k\ell}$ in $\mathcal{P}_2$ to that of $\rho_{k\ell} + \rho_{\ell k}$ from $\mathcal{P}_1$.

Properties 1) and 2) can be checked via Theorems 1 and 2 from [51]. Property 3) can be verified by the optimality conditions for the two problems, and highlights the relation between these two problems. This also shows that, in this case, the Nash equilibrium is a Pareto solution of problem $\mathcal{P}_2$.

Problem $\mathcal{P}_1$ can be solved either analytically in closed form or iteratively by using a steepest-descent algorithm. Nevertheless, we shall focus on problem $\mathcal{P}_2$ since $\mathcal{P}_1$ can be addressed as a special case of $\mathcal{P}_2$ because of Property 3). Unfortunately, there is no analytical expression for general Nash equilibrium problems. We estimate the equilibrium of problem $\mathcal{P}_2$ iteratively by the fixed point of the best response iteration [50], i.e., by using:

$$\boldsymbol{w}_{\mathcal{C}_i}(n+1) = \arg\min_{\boldsymbol{w}_{\mathcal{C}_i}} \overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i}(n)) \qquad (11)$$

for $i = 1, \ldots, Q$. Since the equilibrium is unique and the cost function for each cluster is convex, the solution of (11) can be approximated by means of a steepest-descent iteration as follows:

$$\boldsymbol{w}_{\mathcal{C}_i}(n+1) = \boldsymbol{w}_{\mathcal{C}_i}(n) - \mu\, \nabla_{\boldsymbol{w}_{\mathcal{C}_i}} \overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}(n), \boldsymbol{w}_{-\mathcal{C}_i}(n)) \quad (12)$$

for $i = 1, \ldots, Q$, with $\nabla_{\boldsymbol{w}_{\mathcal{C}_i}}$ denoting the gradient operation with respect to $\boldsymbol{w}_{\mathcal{C}_i}$, and μ a positive step-size. We have from (9) that

$$\nabla_{\boldsymbol{w}_{\mathcal{C}_i}} \overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i}(n)) \propto \sum_{k\in\mathcal{C}_i} (\boldsymbol{R}_{x,k}\boldsymbol{w}_{\mathcal{C}_i} - \boldsymbol{p}_{xd,k}) + \eta \sum_{k\in\mathcal{C}_i} \sum_{\ell\in\mathcal{N}_k\backslash\mathcal{C}_i} \rho_{k\ell}\big(\boldsymbol{w}_{\mathcal{C}_i} - \boldsymbol{w}_{\mathcal{C}(\ell)}(n)\big). \quad (13)$$

where $\boldsymbol{p}_{xd,k} = E\{\boldsymbol{x}_k(n) d_k(n)\}$ is the input-output cross-correlation vector between $\boldsymbol{x}_k(n)$ and $d_k(n)$ at node k. If it is necessary to impose some constraints on the parameters that the network is estimating, then the gradient update relation can be modified by using methods such as projection [52], fixed point iteration techniques [53], or penalty-based techniques [54]. In the body of the paper, we focus on the unconstrained case during the algorithm derivation and its analysis. However, a constrained problem will be presented in the simulation section.

In this paper, we shall consider normalized weights that satisfy

$$\sum_{\ell=1}^{N} \rho_{k\ell} = 1, \quad \text{and} \quad \rho_{k\ell} = 0 \text{ if } \ell \notin \mathcal{N}_k\backslash\mathcal{C}(k). \qquad (14)$$

The sum-to-one normalization is optional but imposed here to clarify the roles of η and $\rho_{k\ell}$, namely, η controls the regularization strength while the $\rho_{k\ell}$ adjust this strength among the neighbors that are connected. Similar to any popular regularization terms, e.g., Tikhonov [55], sparsity-inducing norms [56], TV-norm regularization [57], etc., the parameter η is usually pre-adjusted using prior information or user preference. Clearly, if η is set to 0, the algorithm degenerates to the extensively studied diffusion LMS within each cluster, that is, without information exchange between tasks. The larger the regularization constant η is, the more homogeneous the estimates over the entire network are. This regularization constant leaves the user free to drive the algorithm towards distinct solutions or a homogeneous solution, and can also depend on the structure of the network. How to set these coefficients obviously depends on the application at hand. For instance, in our simulations in later sections, we use the spatial organization of clusters or similarity measures between local reference signals $d_k(n)$ to set the parameters $\rho_{k\ell}$.

B. Local Cost Decomposition and Problem Relaxation

The solution method (12) using (13) requires that every node in the network should have access to the statistical moments $\boldsymbol{R}_{x,k}$ and $\boldsymbol{p}_{xd,k}$ over its cluster. There are two problems with this scenario. First, nodes can only be assumed to have access to information from their immediate neighborhood and the cluster of every node k may include nodes that are not direct neighbors of k. Second, nodes rarely have access to the moments $\{\boldsymbol{R}_{x,d}, \boldsymbol{p}_{xd,k}\}$; instead, they have access to data generated from distributions with these moments. Therefore, more is needed to enable a distributed solution that relies solely on local interactions within neighborhoods and that relies on measured data as opposed to statistical moments. To derive a distributed algorithm, we follow the approach of [3], [17]. The first step in this approach is to show how to express the cost (9) in terms of other local costs that only depend on data from neighborhoods.

Thus, let us introduce an $N \times N$ right stochastic matrix $\boldsymbol{C}$ with nonnegative entries $c_{\ell k}$ such that

$$\sum_{k=1}^{N} c_{\ell k} = 1, \quad \text{and} \quad c_{\ell k} = 0 \text{ if } k \notin \mathcal{N}_\ell \cap \mathcal{C}(\ell). \qquad (15)$$

where $\cap$ denotes the set intersection. With these coefficients, we associate a local cost function of the following form with each node k [3]:

$$J_k^{\text{loc}}(\boldsymbol{w}_{\mathcal{C}(k)}) = \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\, E\left\{\left|d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\,\boldsymbol{w}_{\mathcal{C}(k)}\right|^2\right\}. \qquad (16)$$

One important distinction from the local cost defined in [3] is that in [3] the summation in (16) is defined over the entire neighborhood of node k, i.e., for all $\ell \in \mathcal{N}_k$. Here we are excluding those neighbors of k that do not belong to its cluster. This is because these particular neighbors will be pursuing a different parameter vector than node k. Furthermore, we note in (16) that $\boldsymbol{w}_{\mathcal{C}(k)} = \boldsymbol{w}_{\mathcal{C}(\ell)}$ because $\ell \in \mathcal{C}(k)$. To make the notation simpler, we shall write $\boldsymbol{w}_k$ instead of $\boldsymbol{w}_{\mathcal{C}(k)}$. A consequence of this notation is that $\boldsymbol{w}_k = \boldsymbol{w}_\ell$ for all $\ell \in \mathcal{C}(k)$. Incorporating the estimates of the neighboring clusters, we modify (16) to associate

[1] Let $\boldsymbol{g}(\boldsymbol{w}, \boldsymbol{\zeta}) = [\zeta_i \nabla_{\boldsymbol{w}_{\mathcal{C}_i}} \overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i})]_{i=1}^{Q}$ arranged as a column vector with $\zeta_i > 0$. The cost functions $\{\overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i})\}_{i=1}^{Q}$ satisfy the diagonal strict convexity property, that is, $(\boldsymbol{g}(\hat{\boldsymbol{w}}, \boldsymbol{\zeta}) - \boldsymbol{g}(\boldsymbol{w}, \boldsymbol{\zeta}))^\top(\hat{\boldsymbol{w}} - \boldsymbol{w}) > 0$ for all nonequal $\boldsymbol{w}$, $\hat{\boldsymbol{w}}$.

a regularized local cost function with node k of the following form

$$\overline{J_k^{\rm loc}}(\boldsymbol{w}_k) = \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\, E\left\{\left|d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\,\boldsymbol{w}_k\right|^2\right\} + \eta \sum_{\ell\in\mathcal{N}_k\backslash\mathcal{C}(k)} \rho_{k\ell}\|\boldsymbol{w}_k - \boldsymbol{w}_\ell\|^2. \qquad (17)$$

Observe that this local cost is now solely defined in terms of information that is available to node k from its neighbors. Using this regularized local cost function, it can be verified that the global cost function for cluster $\mathcal{C}_i$ in (9) can be now expressed as

$$\begin{aligned}\overline{J_{\mathcal{C}_i}}(\boldsymbol{w}_{\mathcal{C}_i}, \boldsymbol{w}_{-\mathcal{C}_i}) &= \sum_{k\in\mathcal{C}_i}\left(\sum_{\ell\in\mathcal{C}(k)} c_{\ell k}\, E\left\{\left|d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\,\boldsymbol{w}_{\mathcal{C}(k)}\right|^2\right\}\right.\\ &\quad \left. + \,\eta \sum_{\ell\in\mathcal{N}_k\backslash\mathcal{C}_i} \rho_{k\ell}\,\|\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|^2\right)\\ &= \overline{J_k^{\rm loc}}(\boldsymbol{w}_k) + \sum_{\ell\in\mathcal{C}(k)\backslash k} \overline{J_\ell^{\rm loc}}(\boldsymbol{w}_\ell) \qquad (18)\end{aligned}$$

Let $\boldsymbol{w}_k^o$ denote the minimizer of the local cost function (17), given $\boldsymbol{w}_\ell$ for all $\ell \in \mathcal{N}_k\backslash\mathcal{C}(k)$. A completion-of-squares argument shows that each $\overline{J_k^{\rm loc}}(\boldsymbol{w}_k)$ can be expressed as

$$\overline{J_k^{\rm loc}}(\boldsymbol{w}_k) = \overline{J_k^{\rm loc}}(\boldsymbol{w}_k^o) + \|\boldsymbol{w}_k - \boldsymbol{w}_k^o\|^2_{\overline{\boldsymbol{R}}_k} \qquad (19)$$

where

$$\overline{\boldsymbol{R}}_k = \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\,\boldsymbol{R}_{x,\ell} + \eta \sum_{\ell\in\mathcal{N}_k\backslash\mathcal{C}(k)} \rho_{k\ell}\boldsymbol{I}_L. \qquad (20)$$

Substituting (19) into the second term on the right-hand-side of (18), and discarding the terms $\{\overline{J_\ell^{\rm loc}}(\boldsymbol{w}_\ell^o)\}$ because they are independent of the optimization variables in the cluster, we can consider the following equivalent cost function for cluster $\mathcal{C}(k)$ at node k:

$$\overline{J_{\mathcal{C}(k)}}(\boldsymbol{w}_k) \triangleq \overline{J_k^{\rm loc}}(\boldsymbol{w}_k) + \sum_{\ell\in\mathcal{C}(k)\backslash k} \|\boldsymbol{w}_\ell - \boldsymbol{w}_\ell^o\|^2_{\overline{\boldsymbol{R}}_\ell} \qquad (21)$$

where it holds that $\boldsymbol{w}_\ell = \boldsymbol{w}_k$ because $\ell \in \mathcal{C}(k)$. Note that we have omitted $\boldsymbol{w}_{-k}$ in the notation for $\overline{J_{\mathcal{C}(k)}}(\boldsymbol{w}_k)$ in (21) for the sake of brevity. Therefore, minimizing (21) is equivalent to minimizing the original cost (18) or (9) over $\boldsymbol{w}_k$. However the second term (21) still requires information from nodes ℓ that may not be in the direct neighborhood of node k even though they belong to the same cluster. In order to avoid access to information via multi-hop, we can relax the cost function (21) at node k by considering only information originating from its neighbors. This can be achieved by replacing the range of the index over which the summation in (21) is computed as follows:

$$\overline{J_{\mathcal{C}(k)}}'(\boldsymbol{w}_k) = \overline{J_k^{\rm loc}}(\boldsymbol{w}_k) + \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} \|\boldsymbol{w}_k - \boldsymbol{w}_\ell^o\|^2_{\overline{\boldsymbol{R}}_\ell}. \qquad (22)$$

Usually, especially in the context of adaptive learning in a non-stationary environment, the weighting matrices $\overline{\boldsymbol{R}_\ell}$ are unavailable since the covariance matrices $\boldsymbol{R}_{x,\ell}$ at each node may not be known beforehand. Following an argument based on the Rayleigh-Ritz characterization of eigenvalues, as explained in [3], [25], a useful strategy is to replace each matrix $\overline{\boldsymbol{R}}_\ell$ by a weighted multiple of the identity matrix, say, as:

$$\|\boldsymbol{w}_k - \boldsymbol{w}_\ell^o\|^2_{\overline{\boldsymbol{R}}_\ell} \approx b_{\ell k}\,\|\boldsymbol{w}_k - \boldsymbol{w}_\ell^o\|^2 \qquad (23)$$

for some nonnegative coefficients $b_{\ell k}$ that can possibly depend on the node k. As shown later, these coefficients will be incorporated into a left stochastic matrix to be defined and, therefore, the designer does not need to worry about the selection of the $b_{\ell k}$ at this stage. Based on the arguments presented so far, and using (17), the cost function (22) can then be relaxed to the following form:

$$\begin{aligned}\overline{J_{\mathcal{C}(k)}}''(\boldsymbol{w}_k) = &\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\, E\left\{\left|d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\,\boldsymbol{w}_k\right|^2\right\}\\ &+ \eta \sum_{\ell\in\mathcal{N}_k\backslash\mathcal{C}(k)} \rho_{k\ell}\,\|\boldsymbol{w}_k - \boldsymbol{w}_\ell\|^2\\ &+ \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\|\boldsymbol{w}_k - \boldsymbol{w}_\ell^o\|^2. \qquad (24)\end{aligned}$$

Observe that the two last sums on the right-hand-side of (24) divide the neighbors of node k into two exclusive sets: those that belong to its cluster (last sum) and those that do not belong to its cluster (second term). In summary, the argument so far enabled us to replace the cost (9) by the alternative cost (24) that depends only on data within the neighborhood of node k. We can now proceed to use (24) to derive distributed strategies. Subsequently, we study the stability and mean-square performance of the resulting strategies and show that they are able to perform well despite the stochastic approximations introduced in the derivation.

IV. Stochastic Approximation Algorithms

To begin with, a steepest-descent iteration can be applied by each node k to minimize the cost function (24). Let $\boldsymbol{w}_k(n)$ denote the estimate for $\boldsymbol{w}_k$ at iteration n. Using a constant step-size μ for each node, the update relation would take the following form:

$$\begin{aligned}\boldsymbol{w}_k(n+1) = \boldsymbol{w}_k(n) &- \mu \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\left(\boldsymbol{R}_{x,\ell}\boldsymbol{w}_k(n) - \boldsymbol{p}_{xd,k}\right)\\ &- \mu\eta \sum_{\ell\in\mathcal{N}_k\backslash\mathcal{C}(k)} \rho_{k\ell}\,\left(\boldsymbol{w}_k(n) - \boldsymbol{w}_\ell(n)\right)\\ &- \mu \sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\,\left(\boldsymbol{w}_k(n) - \boldsymbol{w}_\ell^o\right) \qquad (25)\end{aligned}$$

It is well-known in the stochastic adaptation literature (e.g., [58]) that there is a tradeoff between steady-state performance and convergence rate: small step-sizes lead to better mean-square-error performance at the expense of slower convergence. Schemes to switch between different step-size values to enhance the transient and steady-state behavior are possible. We continue the analysis by focusing on a constant step-size. The resulting performance expressions, when desired, can be optimized over the value of the step-size.

Now, among other possible forms, expression (25) can be evaluated in two successive update steps

$$\begin{aligned}\boldsymbol{\psi}_k(n+1) =& \boldsymbol{w}_k(n) - \mu\Big(\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\left(\boldsymbol{R}_{x,\ell}\boldsymbol{w}_k(n) - \boldsymbol{p}_{xd,k}\right)\\ &+ \eta\sum_{\ell\in\mathcal{N}_k\backslash\mathcal{C}(k)} \rho_{k\ell}\left(\boldsymbol{w}_k(n) - \boldsymbol{w}_\ell(n)\right)\Big) \qquad (26)\end{aligned}$$

$$\boldsymbol{w}_k(n+1) = \boldsymbol{\psi}_k(n+1) + \mu\sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\left(\boldsymbol{w}_\ell^o - \boldsymbol{w}_k(n)\right) \qquad (27)$$

Following the same line of reasoning from [3] in the single-task case, and extending the argument to apply to clusters, we use $\boldsymbol{\psi}_\ell(n+1)$ as a local estimate for $\boldsymbol{w}_\ell^o$ in (27) since the latter is unavailable and $\boldsymbol{\psi}_\ell(n+1)$ is an intermediate estimate for it that is available at node ℓ at time $n+1$. In addition, again in step (27), we replace $\boldsymbol{w}_k(n)$ by $\boldsymbol{\psi}_k(n+1)$ since it is a better estimate obtained by incorporating information from the neighbors according to (26). Step (27) then becomes

$$\begin{aligned}\boldsymbol{w}_k(n+1) =& \Big(1 - \mu\sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\Big)\boldsymbol{\psi}_k(n+1)\\ &+\mu\sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\,\boldsymbol{\psi}_\ell(n+1). \qquad (28)\end{aligned}$$

The coefficients in (28) can be redefined as:

$$\begin{aligned}a_{kk} &\triangleq 1 - \mu\sum_{\ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)} b_{\ell k}\\ a_{\ell k} &\triangleq \mu\, b_{\ell k}, \quad \ell\in\mathcal{N}_k^-\cap\mathcal{C}(k)\\ a_{\ell k} &\triangleq 0, \quad \ell\notin\mathcal{N}_k\cap\mathcal{C}(k) \qquad (29)\end{aligned}$$

It can be observed that the entries $\{a_{\ell k}\}$ are nonnegative for all ℓ and k (including a_{kk}) for sufficiently small step-size. Moreover, the matrix $\boldsymbol{A}$ with (ℓ,k)-th entry $a_{\ell k}$ is a left-stochastic matrix, which means that the sum of each of its columns is equal to one. With this notation, we obtain the following adapt-then-combine (ATC) diffusion strategy for solving problem (9) in a distributed manner:

$$\begin{aligned}\boldsymbol{\psi}_k(n+1) =& \boldsymbol{w}_k(n) - \mu\Big(\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\left(\boldsymbol{R}_{x,\ell}\boldsymbol{w}_k(n) - \boldsymbol{p}_{xd,k}\right)\\ &+ \eta\sum_{\ell\in\mathcal{N}_k\backslash\mathcal{C}(k)} \rho_{k\ell}\left(\boldsymbol{w}_k(n) - \boldsymbol{w}_\ell(n)\right)\Big)\\ \boldsymbol{w}_k(n+1) =& \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} a_{\ell k}\,\boldsymbol{\psi}_k(n+1). \qquad (30)\end{aligned}$$

It is also possible to arrive at a combine-then-adapt (CTA) diffusion strategy where the aggregation step is performed prior to the adaptation step [3]. In what follows, it is sufficient to focus on the ATC strategy to illustrate the main results. Employing instantaneous approximations for the required signal moments in (30), we arrive at the desired diffusion strategy for clustered multitask learning described in Algorithm 1 where the regularization factors $\rho_{k\ell}$ are chosen according to (14), and the coefficients $\{c_{\ell k}, a_{\ell k}\}$ are nonnegative scalars chosen at will by the designer to satisfy the following conditions:

$$a_{\ell k}\geq 0, \quad \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ for } \ell\notin\mathcal{N}_k\cap\mathcal{C}(k) \qquad (31)$$

$$c_{\ell k}\geq 0, \quad \sum_{k\in\mathcal{N}_\ell\cap\mathcal{C}(\ell)} c_{\ell k} = 1, \quad c_{\ell k} = 0 \text{ for } k\notin\mathcal{N}_\ell\cap\mathcal{C}(\ell) \qquad (32)$$

There are several ways to select these coefficients such as using the averaging rule, the relative variance rule, the Metropolis rule, etc., see [3] for a listing of these and other choices.

Algorithm 1: Diffusion LMS for clustered multitask networks

Start with $\boldsymbol{w}_k(0) = 0$ for all k, and repeat:

$$\begin{cases}\boldsymbol{\psi}_k(n+1) = \boldsymbol{w}_k(n)\\ \qquad + \mu\Big(\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}[d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\boldsymbol{w}_k(n)]\,\boldsymbol{x}_\ell(n)\\ \qquad + \eta\sum_{\ell\in\mathcal{N}_k\backslash\mathcal{C}(k)} \rho_{k\ell}\left(\boldsymbol{w}_\ell(n) - \boldsymbol{w}_k(n)\right)\Big)\\ \boldsymbol{w}_k(n+1) = \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} a_{\ell k}\,\boldsymbol{\psi}_k(n+1)\end{cases} \qquad (33)$$

In the case of a single-task network when there is a single cluster that consists of the entire set of nodes we get $\mathcal{N}_k\cap\mathcal{C}(k) = \mathcal{N}_k$ and $\mathcal{N}_k\backslash\mathcal{C}(k) = \emptyset$ for all k, so that expression (33) reduces to the diffusion adaptation strategy [3], [17] described in Algorithm 2.

Algorithm 2: Diffusion LMS for single-task networks **[16]**, **[17]**.

Start with $\boldsymbol{w}_k(0) = 0$ for all k, and repeat:

$$\begin{cases}\boldsymbol{\psi}_k(n+1) = \boldsymbol{w}_k(n)\\ \qquad + \mu\sum_{\ell\in\mathcal{N}_k} c_{\ell k}[d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\boldsymbol{w}_k(n)]\,\boldsymbol{x}_\ell(n)\\ \boldsymbol{w}_k(n+1) = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_k(n+1)\end{cases} \qquad (34)$$

Algorithm 3: Diffusion LMS for multitask networks

Start with $\boldsymbol{w}_k(0) = 0$ for all k, and repeat:

$$\begin{aligned}\boldsymbol{w}_k(n+1) =& \boldsymbol{w}_k(n) + \mu\,[d_k(n) - \boldsymbol{x}_k^\top(n)\boldsymbol{w}_k(n)]\,\boldsymbol{x}_k(n)\\ &+\eta\,\mu\sum_{\ell\in\mathcal{N}_k^-} \rho_{k\ell}\left(\boldsymbol{w}_\ell(n) - \boldsymbol{w}_k(n)\right) \qquad (35)\end{aligned}$$

In the case of a multitask network where the size of each cluster is one, we have $\mathcal{N}_k\cap\mathcal{C}(k) = \{k\}$ and $\mathcal{N}_k\backslash\mathcal{C}(k) = \mathcal{N}_k^-$ for all k. Then algorithm (33) degenerates into Algorithm 3. Interestingly, this is the instantaneous gradient counterpart of (12) for each node.

It is clear that there is no significant difference between the computational complexities of Algorithms 1 and 2. Algorithm 3 can be viewed as a spatially-regularized LMS algorithm, with a slight additional communication burden compared to non-cooperative LMS.

V. Mean-Square Error Performance Analysis

We now examine the stochastic behavior of the adaptive diffusion strategy (33). In order to address this question, we collect information from across the network into block vectors and matrices. In particular, let us denote by $\boldsymbol{w}(n)$, $\boldsymbol{w}^\star$ and $\boldsymbol{\psi}$ the block weight estimate vector, the block optimum weight vector, and block intermediate weight estimate vector, all of size $LN\times 1$, i.e.,

$$\boldsymbol{w}(n) = \text{col}\{\boldsymbol{w}_k(n)\}_{k=1}^N, \tag{36}$$

$$\boldsymbol{w}^\star = \text{col}\{\boldsymbol{w}_k^\star\}_{k=1}^N, \tag{37}$$

$$\boldsymbol{\psi}(n) = \text{col}\{\boldsymbol{\psi}_k(n)\}_{k=1}^N, \tag{38}$$

where $\text{col}\{\cdot\}$ stacks its vector arguments, and $\boldsymbol{w}_k^\star = \boldsymbol{w}_{\mathcal{C}(k)}^\star$. The weight error vector for each node k at iteration n is defined by $\boldsymbol{v}_k(n) = \boldsymbol{w}_k(n) - \boldsymbol{w}_k^\star$. The weight error vectors $\boldsymbol{v}_k(n)$ are also stacked on top of each other to get the block weight error vector defined as follows:

$$\boldsymbol{v}(n) = \text{col}\{\boldsymbol{v}_k(n)\}_{k=1}^N \tag{39}$$

To perform the theoretical analysis, we introduce the following independence assumption.

Assumption 1: (Independent regressors) The regression vectors $\boldsymbol{x}_k(n)$ arise from a stationary random process that is temporally stationary, temporally white and independent over space with covariance matrix $\boldsymbol{R}_{x,k} = E\{\boldsymbol{x}_k(n)\boldsymbol{x}_k^\top(n)\} > 0$.

A direct consequence is that $\boldsymbol{x}_k(n)$ is independent of $\boldsymbol{v}_\ell(m)$ for all ℓ and $m \leq n$. Although not true in general, this assumption is commonly used for analyzing adaptive constructions because it allows to simplify the derivations without constraining the conclusions [58].

A. Mean Error Behavior Analysis

The estimation error that appears in the first equation of (33) can be rewritten as

$$d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\boldsymbol{w}_k(n) = z_\ell(n) - \boldsymbol{x}_\ell^\top(n)\boldsymbol{v}_k(n) \tag{40}$$

because $\boldsymbol{w}_\ell^\star = \boldsymbol{w}_k^\star$ for all $\ell \in \mathcal{N}_k \cap \mathcal{C}(k)$. Subtracting $\boldsymbol{w}_k^\star$ from both sides of the first equation in (33), and using the above relation, the update equation for the block weight error vector of $\boldsymbol{\psi}_k(n+1)$ can be expressed as

$$\begin{aligned}\boldsymbol{\psi}(n+1) - \boldsymbol{w}^\star = \boldsymbol{v}(n) - \mu\,\boldsymbol{H}_x(n)\,\boldsymbol{v}(n) + \mu\,\boldsymbol{p}_{zx}(n)\\ -\mu\,\eta\,\boldsymbol{Q}\,(\boldsymbol{v}(n) + \boldsymbol{w}^\star)\end{aligned} \tag{41}$$

where

$$\boldsymbol{Q} = \boldsymbol{I}_{LN} - \boldsymbol{P}\otimes\boldsymbol{I}_L, \tag{42}$$

with $\otimes$ denoting the Kronecker product, and $\boldsymbol{P}$ the $N\times N$ matrix with (k,ℓ)-th entry $\rho_{k\ell}$, and $\boldsymbol{P}_{kk} = 1$ if $\mathcal{N}_k\backslash\mathcal{C}(k)$ is empty. Moreover, the matrix $\boldsymbol{H}_x(n)$ is block diagonal of size $LN\times LN$ defined as

$$\boldsymbol{H}_x(n) = \text{diag}\{\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\,\boldsymbol{x}_\ell(n)\boldsymbol{x}_\ell^\top(n)\}_{k=1}^N, \tag{43}$$

and $\boldsymbol{p}_{zx}(n)$ is the following vector of length $LN\times 1$:

$$\boldsymbol{p}_{zx}(n) = \text{col}\{\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\,\boldsymbol{x}_\ell^\top(n) z_\ell(n)\}_{k=1}^N \tag{44}$$

Let $\boldsymbol{A}_I = \boldsymbol{A}\otimes\boldsymbol{I}_L$. The second equation in (33) then allows us to write

$$\boldsymbol{w}(n+1) = \boldsymbol{A}_I^\top\,\boldsymbol{\psi}(n+1) \tag{45}$$

Subtracting $\boldsymbol{w}^\star$ from both sides of the above expression, and using (41), the update relation can be written as

$$\begin{aligned}\boldsymbol{v}(n+1) = \boldsymbol{A}_I^\top\,[\boldsymbol{I}_{LN} - \mu\,(\boldsymbol{H}_x(n) + \eta\,\boldsymbol{Q})]\,\boldsymbol{v}(n)\\ +\mu\,\boldsymbol{A}_I^\top\,\boldsymbol{p}_{zx}(n) - \mu\,\eta\,\boldsymbol{A}_I^\top\boldsymbol{Q}\,\boldsymbol{w}^\star\end{aligned} \tag{46}$$

Taking the expectation of both sides, and using Assumption 1 we get

$$\begin{aligned}E\{\boldsymbol{v}(n+1)\} = \boldsymbol{A}_I^\top\,[\boldsymbol{I}_{LN} - \mu\,(\boldsymbol{H}_R + \eta\,\boldsymbol{Q})]\,E\{\boldsymbol{v}(n)\}\\ -\mu\,\eta\,\boldsymbol{A}_I^\top\boldsymbol{Q}\,\boldsymbol{w}^\star\end{aligned} \tag{47}$$

where

$$\boldsymbol{H}_R \triangleq E\{\boldsymbol{H}_x(n)\} = \text{diag}\,\{\boldsymbol{R}_1,\ldots,\boldsymbol{R}_N\} \tag{48}$$

with

$$\boldsymbol{R}_k = \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\,\boldsymbol{R}_{x,\ell}. \tag{49}$$

Theorem 1: (Stability in the mean) Assume data model (1) and Assumption 1 hold. Then, for any initial condition, the diffusion multitask strategy (33) asymptotically converges in the mean if the step-size is chosen to satisfy

$$\rho\left(\boldsymbol{A}_I^\top\,[\boldsymbol{I}_{LN} - \mu\,(\boldsymbol{H}_R + \eta\,\boldsymbol{Q})]\right) < 1 \tag{50}$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument. A *sufficient* condition for (50) to hold is to choose μ such that

$$0 < \mu < \frac{2}{\max_k\{\lambda_{\max}(\boldsymbol{R}_k)\} + 2\,\eta} \tag{51}$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of its matrix argument. In that case, it follows from (47) that the asymptotic mean bias is given by

$$\begin{aligned}&\lim_{n\to\infty} E\{\boldsymbol{v}(n)\}\\ &= \mu\eta\left\{\boldsymbol{A}_I^\top[\boldsymbol{I}_{LN} - \mu(\boldsymbol{H}_R + \eta\boldsymbol{Q})] - \boldsymbol{I}_{LN}\right\}^{-1}\boldsymbol{A}_I^\top\boldsymbol{Q}\boldsymbol{w}^\star.\end{aligned} \tag{52}$$

Proof: Since any induced matrix norm is lower bounded by the spectral radius, we have the following relation in terms of the block maximum norm (see [3] for definition and properties of the norm):

$$\rho\left(\boldsymbol{A}_I^\top\left[\boldsymbol{I}_{LN}-\mu(\boldsymbol{H}_R+\eta\boldsymbol{Q})\right]\right)\\ \leq \|\boldsymbol{A}_I^\top(\boldsymbol{I}_{LN}-\mu\,(\boldsymbol{H}_R+\eta\boldsymbol{Q}))\,\|_{b,\infty} \quad (53)$$

Now using norm inequalities and the fact that $\boldsymbol{A}$ is a left-stochastic matrix (whose block maximum norm is equal to one), we find that:

$$\begin{aligned}&\|\boldsymbol{A}_I^\top\left[\boldsymbol{I}_{LN}-\mu\,(\boldsymbol{H}_R+\eta\,\boldsymbol{Q})\right]\|_{b,\infty}\\ &\leq \|\boldsymbol{I}_{LN}-\mu\,(\boldsymbol{H}_R+\eta\,\boldsymbol{Q})\|_{b,\infty}\\ &\leq \|\boldsymbol{I}_{LN}-\mu\,\boldsymbol{H}_R-\mu\eta\boldsymbol{I}_{LN}\|_{b,\infty}\\ &\quad+\mu\eta\,\|\boldsymbol{P}\otimes\boldsymbol{I}_L\|_{b,\infty}\end{aligned} \quad (54)$$

using (42). Now, it holds that

$$\|\boldsymbol{P}\otimes\boldsymbol{I}_L\|_{b,\infty}=\|\boldsymbol{P}\|_\infty=1 \quad (55)$$

because $\boldsymbol{P}$ is a right stochastic matrix according to condition (14). Furthermore, since $(1-\mu\eta)\,\boldsymbol{I}_{LN}-\mu\,\boldsymbol{H}_R$ is a block diagonal Hermitian matrix, its block maximum norm is equal to its spectral radius [3]. We therefore conclude that a sufficient condition for mean stability is to require

$$\rho\,((1-\mu\eta)\,\boldsymbol{I}_{LN}-\mu\,\boldsymbol{H}_R)+\mu\eta<1, \quad (56)$$

which yields condition (51). ■

Condition (51) shows that the stability limit in the mean of the multitask diffusion LMS is lower than diffusion LMS (34) due to the presence of η. The mean convergence rate of the algorithm is governed by the spectral radius $\rho(\boldsymbol{A}_I^\top[\boldsymbol{I}_{LN}-\mu\,(\boldsymbol{H}_R+\eta\,\boldsymbol{Q})])$.

B. Mean-Square Error Behavior Analysis

In order to make the presentation clearer, we shall use the following notation for terms in the weight-error expression (46):

$$\begin{aligned}\boldsymbol{B}(n)&=\boldsymbol{A}_I^\top\left[\boldsymbol{I}_{LN}-\mu\,(\boldsymbol{H}_x(n)+\eta\,\boldsymbol{Q})\right]\\ \boldsymbol{g}(n)&=\boldsymbol{A}_I^\top\,\boldsymbol{p}_{zx}(n)\\ \boldsymbol{r}&=\boldsymbol{A}_I^\top\boldsymbol{Q}\,\boldsymbol{w}^\star\end{aligned} \quad (57)$$

so that

$$\boldsymbol{v}(n+1)=\boldsymbol{B}(n)\,\boldsymbol{v}(n)+\mu\,\boldsymbol{g}(n)-\mu\,\eta\,\boldsymbol{r}. \quad (58)$$

Using Assumption 1 and $E\{\boldsymbol{g}(n)\}=0$, the mean-square of the weight error vector $\boldsymbol{v}(n+1)$, weighted by any positive semi-definite matrix $\boldsymbol{\Sigma}$ that we are free to choose, satisfies the following relation:

$$\begin{aligned}&E\{\|\boldsymbol{v}(n+1)\|^2_{\boldsymbol{\Sigma}}\}\\ &=E\{\|\boldsymbol{v}(n)\|^2_{\boldsymbol{\Sigma}'}\}+\mu^2\ \text{trace}\left\{\boldsymbol{\Sigma}E\{\boldsymbol{g}(n)\boldsymbol{g}^\top(n)\}\right\}\\ &\quad+\mu^2\eta^2\,\|\boldsymbol{r}\|^2_{\boldsymbol{\Sigma}}-2\mu\,\eta\,\boldsymbol{r}^\top\,\boldsymbol{\Sigma}\boldsymbol{B}\,E\{\boldsymbol{v}(n)\}\end{aligned} \quad (59)$$

where

$$\boldsymbol{B}\triangleq\{\boldsymbol{B}(n)\}=\boldsymbol{A}_I^\top\left[(\boldsymbol{I}_{LN}-\mu\,(\boldsymbol{H}_R+\eta\,\boldsymbol{Q})\right] \quad (60)$$

$$\boldsymbol{\Sigma}'\triangleq E\{\boldsymbol{B}^\top(n)\,\boldsymbol{\Sigma}\,\boldsymbol{B}(n)\} \quad (61)$$

In expression (59), the freedom in selecting $\boldsymbol{\Sigma}$ will allow us to derive several performance metrics. Let

$$\begin{aligned}\boldsymbol{G}&=E\{\boldsymbol{g}(n)\boldsymbol{g}^\top(n)\}\\ &=\boldsymbol{A}_I^\top\,\boldsymbol{C}_I^\top\,\text{diag}\{\sigma^2_{z,1}\boldsymbol{R}_{x,1},\ldots,\sigma^2_{z,N}\boldsymbol{R}_{x,N}\}\,\boldsymbol{C}_I\,\boldsymbol{A}_I\end{aligned} \quad (62)$$

where $\boldsymbol{C}_I=\boldsymbol{C}\otimes\boldsymbol{I}_L$. Then, relation (59) can be rewritten as

$$\begin{aligned}E\{\|\boldsymbol{v}(n+1)\|^2_{\boldsymbol{\Sigma}}\}&=E\{\|\boldsymbol{v}(n)\|^2_{\boldsymbol{\Sigma}'}\}+\mu^2\,\text{trace}\,\{\boldsymbol{\Sigma}\boldsymbol{G}\}\\ &\quad+\mu^2\eta^2\,\|\boldsymbol{r}\|^2_{\boldsymbol{\Sigma}}-2\mu\,\eta\,\boldsymbol{r}^\top\,\boldsymbol{\Sigma}\boldsymbol{B}\,E\{\boldsymbol{v}(n)\}\end{aligned} \quad (63)$$

We would like to argue that this variance relation converges for sufficiently small step-sizes and we would also like to evaluate its steady-state value in order to determine the mean-square-error of the distributed strategy. However, note that the weighting matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'$ on both sides of (63) are different, which means that (63) is still not an actual recursion. To handle this situation, we transform the weighting matrices into vector forms as follows. Let $\text{vec}(\cdot)$ denote the operator that stacks the columns of a matrix on top of each other. Vectorizing $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'$ by $\boldsymbol{\sigma}=\text{vec}(\boldsymbol{\Sigma})$ and $\boldsymbol{\sigma}'=\text{vec}(\boldsymbol{\Sigma}')$, it can be verified that relation (61) between them can be expressed as the following linear transformation:

$$\boldsymbol{\sigma}'=\boldsymbol{K}\boldsymbol{\sigma} \quad (64)$$

where $\boldsymbol{K}$ is the $(LN)^2\times(LN)^2$ matrix given by

$$\begin{aligned}\boldsymbol{K}&=E\{\boldsymbol{B}^\top(n)\otimes\boldsymbol{B}^\top(n)\}\\ &=\boldsymbol{A}_I\otimes\boldsymbol{A}_I-\mu\,(\boldsymbol{H}_R+\eta\,\boldsymbol{Q})^\top\boldsymbol{A}_I\otimes\boldsymbol{A}_I\\ &\quad-\mu\boldsymbol{A}_I\otimes(\boldsymbol{H}_R+\eta\,\boldsymbol{Q})^\top\boldsymbol{A}_I\\ &\quad+\mu^2E\{(\boldsymbol{H}(n)+\eta\,\boldsymbol{Q})^\top\boldsymbol{A}_I\otimes(\boldsymbol{H}(n)+\eta\,\boldsymbol{Q})^\top\boldsymbol{A}_I\}.\end{aligned} \quad (65)$$

Neglecting the influence of second-order terms in μ, $\boldsymbol{K}$ can be approximated by

$$\boldsymbol{K}\approx\boldsymbol{B}^\top\otimes\boldsymbol{B}^\top. \quad (66)$$

Finally, let us define $\boldsymbol{f}(\boldsymbol{\sigma},E\{\boldsymbol{v}(n)\})$ as the last two terms on the right hand side of (63), i.e.,

$$\boldsymbol{f}(\boldsymbol{\sigma},E\{\boldsymbol{v}(n)\})\triangleq\mu^2\eta^2\,\|\boldsymbol{r}\|^2_{\boldsymbol{\sigma}}-2\mu\eta\,(\boldsymbol{B}\,E\{\boldsymbol{v}(n)\}\otimes\boldsymbol{r})^\top\,\boldsymbol{\sigma} \quad (67)$$

where we will be using the notation $\|\cdot\|_\Sigma$ and $\|\cdot\|_\sigma$ interchangeably. For notational convenience, we are omitting the argument $\boldsymbol{r}$ of $\boldsymbol{f}$ since it is deterministic. Equation (63) can be expressed as follows:

$$\begin{aligned}E\{\|\boldsymbol{v}(n+1)\|^2_{\boldsymbol{\sigma}}\}&=E\{\|\boldsymbol{v}(n)\|^2_{\boldsymbol{K\sigma}}\}\\ &\quad+\mu^2\text{vec}(\boldsymbol{G}^\top)^\top\boldsymbol{\sigma}+\boldsymbol{f}(\boldsymbol{\sigma},E\{\boldsymbol{v}(n)\}).\end{aligned} \quad (68)$$

Theorem 2: (Mean-square stability) Assume data model (1) and Assumption 1 hold. Assume further that the step-size μ is sufficiently small such that approximation (66) is justified by neglecting higher-order powers of μ, and relation (68) can be used as a reasonable representation for the evolution of the (weighted) mean-square-error. Then, the diffusion multitask strategy (33) is mean-square stable if the matrix $\boldsymbol{K}$ is stable. Under approximation (66), the stability of $\boldsymbol{K}$ is guaranteed by sufficiently small step-sizes that also satisfy (51).

Proof: Iterating recursion (68) starting from $n = 0$, we find that

$$\begin{aligned} E\{\|\boldsymbol{v}(n+1)\|^2_{\boldsymbol{\sigma}}\} &= E\{\|\boldsymbol{v}(0)\|^2_{\boldsymbol{K}^{n+1}\boldsymbol{\sigma}}\} + \mu^2 \operatorname{vec}(\boldsymbol{G}^\top)^\top \sum_{i=0}^{n} \boldsymbol{K}^i\boldsymbol{\sigma} \\ &\quad + \sum_{i=0}^{n} \boldsymbol{f}(\boldsymbol{K}^i\boldsymbol{\sigma}, E\{\boldsymbol{v}(n-i)\}) \end{aligned} \tag{69}$$

with initial condition $\boldsymbol{v}(0) = \boldsymbol{w}(0) - \boldsymbol{w}^\star$. Provided that $\boldsymbol{K}$ is stable, the first and second terms on the RHS of (69) converge as $n \to \infty$, to zero for the former, and to a finite value for the latter. Consider now the third term on the RHS of (69). We know from (47) that $E\{\boldsymbol{v}(n)\}$ is uniformly bounded because (47) is a BIBO stable recursion with a bounded driving term $-\mu\,\eta\,\boldsymbol{A}_I^\top\boldsymbol{Q}\,\boldsymbol{w}^\star$. Moreover, from (67), the expression for $\boldsymbol{f}(\boldsymbol{K}^i\boldsymbol{\sigma}, E\{\boldsymbol{v}(n-i)\})$ can be written as

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{K}^i\boldsymbol{\sigma}, E\{\boldsymbol{v}(n-i)\}) &= \mu^2\eta^2 \operatorname{vec}\{\boldsymbol{r}\boldsymbol{r}^\top\}^\top\boldsymbol{K}^i\boldsymbol{\sigma} \\ &\quad -2\mu\,\eta\,(\boldsymbol{B}\,E\{\boldsymbol{v}(n-i)\}\otimes\boldsymbol{r})^\top\,\boldsymbol{K}^i\boldsymbol{\sigma}. \end{aligned} \tag{70}$$

We further know that $\boldsymbol{K}$ defined by (66) is stable. Therefore, there exists a matrix norm [3], denoted by $\|\cdot\|_\rho$, such that $\|\boldsymbol{K}\|_\rho = c_\rho < 1$. Applying this norm to $\boldsymbol{f}$ and using the triangular inequality, we can deduce that $|\boldsymbol{f}(\boldsymbol{K}^i\boldsymbol{\sigma}, E\{\boldsymbol{v}(n-i)\})| < \nu c_\rho^i$ for some positive finite constant ν. It follows that the sum appearing as the right-most term in (69) converges as $n \to \infty$. We conclude that $E\{\|\boldsymbol{v}(n+1)\|^2_\sigma\}$ converges to a bounded value as $n \to \infty$, and the algorithm is said to be mean-square stable. ■

Theorem 3: (Transient MSD) Considering a sufficiently small step-size μ that ensures mean and mean-square stability, and selecting $\boldsymbol{\Sigma} = \frac{1}{N}\boldsymbol{I}_{LN}$, then the network MSD learning curve, defined by $\zeta(n) = \frac{1}{N}E\{\|\boldsymbol{v}(n)\|\}^2$ evolves according to the following recursions for $n \geq 0$:

$$\begin{aligned} \zeta(n+1) = \zeta(n) + \frac{1}{N}\Big(&\mu^2 \operatorname{vec}(\boldsymbol{G}^\top)^\top\boldsymbol{K}^n \operatorname{vec}(\boldsymbol{I}_{LN}) \\ &- E\{\|\boldsymbol{v}(0)\|^2_{(\boldsymbol{I}_{(NL)^2}-\boldsymbol{K})\boldsymbol{K}^n \operatorname{vec}(\boldsymbol{I}_{LN})}\} \\ &+ \mu^2\eta^2\,\|\boldsymbol{r}\|^2_{\boldsymbol{K}^n \operatorname{vec}(\boldsymbol{I}_{LN})} \\ &- 2\mu\eta\,\left(\boldsymbol{\Gamma}(n) + (\boldsymbol{B}\,E\{\boldsymbol{v}(n)\}\otimes\boldsymbol{r})^\top\right)\operatorname{vec}(\boldsymbol{I}_{LN})\Big) \end{aligned} \tag{71}$$

$$\boldsymbol{\Gamma}(n+1) = \boldsymbol{\Gamma}(n)\boldsymbol{K} + (\boldsymbol{B}\,E\{\boldsymbol{v}(n)\}\otimes\boldsymbol{r})^\top(\boldsymbol{K}-\boldsymbol{I}_{(LN)^2}) \tag{72}$$

with initial condition $\zeta(0) = \frac{1}{N}\|\boldsymbol{v}(0)\|^2$ and $\boldsymbol{\Gamma}(0) = \boldsymbol{0}_{1\times(LN)}$.

Proof: Comparing (69) at instants $n+1$ and n, we can relate $E\{\|\boldsymbol{v}(n+1)\|^2_{\boldsymbol{\sigma}}\}$ to $E\{\|\boldsymbol{v}(n)\|^2_{\boldsymbol{\sigma}}\}$ as follows:

$$\begin{aligned} &E\{\|\boldsymbol{v}(n+1)\|^2_{\boldsymbol{\sigma}}\} \\ &\quad = E\{\|\boldsymbol{v}(n)\|^2_{\boldsymbol{\sigma}}\} + \mu^2 \operatorname{vec}(\boldsymbol{G}^\top)^\top\,\boldsymbol{K}^n\boldsymbol{\sigma} \\ &\qquad - E\{\|\boldsymbol{v}(0)\|^2_{(\boldsymbol{I}_{(NL)^2}-\boldsymbol{K})\boldsymbol{K}^n\boldsymbol{\sigma}}\} + \sum_{i=0}^{n}\boldsymbol{f}(\boldsymbol{K}^i\boldsymbol{\sigma}, E\{\boldsymbol{v}(n-i)\}) \\ &\qquad - \sum_{i=0}^{n-1}\boldsymbol{f}(\boldsymbol{K}^i\boldsymbol{\sigma}, E\{\boldsymbol{v}(n-1-i)\}) \end{aligned} \tag{73}$$

We can rewrite the last two terms on the RHS of (73) as

$$\begin{aligned} &\sum_{i=0}^{n}\boldsymbol{f}(\boldsymbol{K}^i\boldsymbol{\sigma}, E\{\boldsymbol{v}(n-i)\}) - \sum_{i=0}^{n-1}\boldsymbol{f}(\boldsymbol{K}^i\boldsymbol{\sigma}, E\{\boldsymbol{v}(n-1-i)\}) \\ &= \mu^2\eta^2\|\boldsymbol{r}\|^2_{\boldsymbol{K}^n\boldsymbol{\sigma}} - 2\mu\,\eta\,\left((\boldsymbol{B}\,E\{\boldsymbol{v}(n)\}\otimes\boldsymbol{r})^\top + \boldsymbol{\Gamma}(n)\right)\boldsymbol{\sigma} \end{aligned} \tag{74}$$

where

$$\begin{aligned} \boldsymbol{\Gamma}(n) = &\sum_{i=1}^{n}(\boldsymbol{B}E\{\boldsymbol{v}(n-i)\}\otimes\boldsymbol{r})^\top\,\boldsymbol{K}^i \\ &+ \sum_{i=0}^{n-1}(\boldsymbol{B}E\{\boldsymbol{v}(n-i-1)\}\otimes\boldsymbol{r})^\top\,\boldsymbol{K}^i. \end{aligned} \tag{75}$$

We can then reformulate recursion (73) as follows:

$$\begin{aligned} E\{\|\boldsymbol{v}(n+1)\|^2_{\boldsymbol{\sigma}}\} = {}& E\{\|\boldsymbol{v}(n)\|^2_{\boldsymbol{\sigma}}\} + \mu^2 \operatorname{vec}(\boldsymbol{G}^\top)^\top\,\boldsymbol{K}^n\boldsymbol{\sigma} \\ &- E\{\|\boldsymbol{v}(0)\|^2_{(\boldsymbol{I}_{(NL)^2}-\boldsymbol{K})\boldsymbol{K}^n\boldsymbol{\sigma}}\} + \mu^2\eta^2\|\boldsymbol{r}\|^2_{\boldsymbol{K}^n\boldsymbol{\sigma}} \\ &- 2\mu\eta\,(\boldsymbol{\Gamma}(n) + (\boldsymbol{B}\,E\{\boldsymbol{v}(n)\}\otimes\boldsymbol{r})^\top\,\boldsymbol{\sigma} \end{aligned} \tag{76}$$

$$\boldsymbol{\Gamma}(n+1) = \boldsymbol{\Gamma}(n)\boldsymbol{K} + (\boldsymbol{B}\,E\{\boldsymbol{v}(n)\}\otimes\boldsymbol{r})^\top\,(\boldsymbol{K}-\boldsymbol{I}_{(LN)^2}) \tag{77}$$

with $\boldsymbol{\Gamma}(0) = \boldsymbol{0}_{1\times(LN)}$. To derive the transient curve for the MSD, we replace σ by $\frac{1}{N}\operatorname{vec}(\boldsymbol{I}_{LN})$. ■

Theorem 4: (Steady-state MSD) If the step size is chosen sufficiently small to ensure mean and mean-square-error convergence, then the value of the steady-state MSD for the diffusion network (33) is given by

$$\begin{aligned} \zeta^\star = {}& \frac{\mu^2}{N}\operatorname{vec}(\boldsymbol{G}^\top)^\top\,(\boldsymbol{I}_{(LN)^2}-\boldsymbol{K})^{-1}\operatorname{vec}(\boldsymbol{I}_{LN}) \\ &+ \boldsymbol{f}\left(\frac{1}{N}(\boldsymbol{I}_{(LN)^2}-\boldsymbol{K})^{-1}\operatorname{vec}(\boldsymbol{I}_{LN}), E\{\boldsymbol{v}(\infty)\}\right) \end{aligned} \tag{78}$$

where $E\{\boldsymbol{v}(\infty)\}$ is determined by expression (52).

Proof: The steady-state MSD is the limiting value

$$\zeta^\star = \lim_{n\to\infty}\frac{1}{N}E\{\|\boldsymbol{v}(n)\|\}^2. \tag{79}$$

From the recursive expression (68) we obtain as $n \to \infty$ that

$$\begin{aligned} &\lim_{n\to\infty} E\{\|\boldsymbol{v}(n)\|^2_{(\boldsymbol{I}_{(NL)^2}-\boldsymbol{K})\boldsymbol{\sigma}}\} \\ &\qquad = \mu^2 \operatorname{vec}(\boldsymbol{G}^\top)^\top\boldsymbol{\sigma} + \boldsymbol{f}(\boldsymbol{\sigma}, E\{\boldsymbol{v}(\infty)\}). \end{aligned} \tag{80}$$

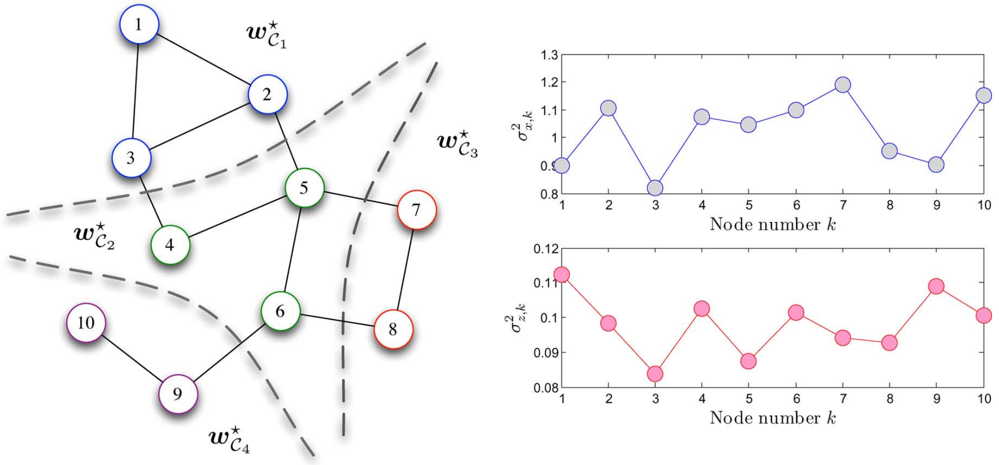
Comparing expressions (79) and (80), we observe that to arrive at the MSD requires us to choose σ to satisfy

$$(\boldsymbol{I}_{(NL)^2}-\boldsymbol{K})\,\boldsymbol{\sigma}_{\text{MSD}} = \frac{1}{N}\operatorname{vec}(\boldsymbol{I}_{LN}). \tag{81}$$

This leads to expression (78). ■

VI. Simulation Examples

In this section, we first conduct simulations on a simple network to illustrate the proposed algorithm and the analyt-



(a) (b)

Fig. 2. Experimental setup. Left: network studied in Section VI-A, with 10 nodes divided into 4 different clusters. Right: input signal and noise variances for each node.

ical performance models. Then, we provide several examples where the proposed distributed learning strategy may find applications.

A. Illustrative Numerical Example

In this subsection we provide an illustrative example to show how the proposed distributed algorithm converges over clustered multitask network. We consider a network consisting of 10 nodes with the topology depicted in Fig. 2 (left). The nodes were divided into 4 clusters: $\mathcal{C}_1 = \{1,2,3\}$, $\mathcal{C}_2 = \{4,5,6\}$, $\mathcal{C}_3 = \{7,8\}$ and $\mathcal{C}_4 = \{9,10\}$. Two-dimensional coefficient vectors of the form $\boldsymbol{w}^\star_{\mathcal{C}_i} = \boldsymbol{w}_o + \delta\boldsymbol{w}_{\mathcal{C}_i}$ were chosen as $\boldsymbol{w}_o = [0.5, -0.4]^\top$, $\delta\boldsymbol{w}_{\mathcal{C}_1} = [0.0287, -0.005]^\top$, $\delta\boldsymbol{w}_{\mathcal{C}_2} = [0.0234, 0.005]^\top$, $\delta\boldsymbol{w}_{\mathcal{C}_3} = [-0.0335, 0.0029]^\top$, and $\delta\boldsymbol{w}_{\mathcal{C}_4} = [0.0224, 0.00347]^\top$. The regression inputs $\boldsymbol{x}_k(n)$ were zero-mean 2×1 random vectors governed by a Gaussian distribution with covariance matrices $\boldsymbol{R}_{x,k} = \sigma^2_{x,k}\boldsymbol{I}_L$, and the $\sigma^2_{x,k}$ shown in the top right plot of Fig. 2. The background noises $z_k(n)$ were i.i.d. zero-mean Gaussian random variables, independent of any other signals. The corresponding variances $\sigma^2_{z,k}$ are depicted in the bottom right plot of Fig. 2.

Regularization strength $\rho_{k\ell}$ was set to $\rho_{k\ell} = |\mathcal{N}_k\backslash\mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k\backslash\mathcal{C}(k)$, and $\rho_{k\ell} = 0$ for any other ℓ. This setting usually leads to asymmetrical regularization weights. We considered the diffusion algorithm with measurement diffusion governed by a uniform matrix $\boldsymbol{C}$ such that $c_{\ell k} = |\mathcal{N}_\ell \cap \mathcal{C}(\ell)|^{-1}$ for $k \in \mathcal{N}_\ell \cap \mathcal{C}(\ell)$. Likewise, a uniform $\boldsymbol{A}$ was used such that $a_{\ell k} = |\mathcal{N}_k \cap \mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k \cap \mathcal{C}(k)$.

The algorithm was run with different step-size and regularization parameters (μ, η) such as $(0.01, 0.1)$, $(0.05, 0.1)$ and $(0.01, 1)$. Simulation results were obtained by averaging 100 Monte-Carlo runs. Transient MSD curves were obtained by (71) and (72). Steady-state MSD values were obtained by expression (78). It can be observed in the left plot of Fig. 3 that the models accurately match the simulated results.

These models were used to illustrate the performance of several learning strategies: 1) the non-cooperative LMS algorithm, 2) the multitask algorithm (Algorithm 3), and 3) the clustered multitask algorithm (Algorithm 1). The non-cooperative algorithm was obtained by assigning a cluster to each node and setting $\eta = 0$. The multitask algorithm was obtained by assigning a cluster to each node and setting $\eta \neq 0$. Note that Algorithm 2 was not considered for comparison because it is a single-task estimation method. The right plot of Fig. 3 shows that the noncooperative algorithm has the largest MSD as nodes do not collaborate for additional benefit. If estimation is performed without cluster information, but only with regularization between nodes as in the case of the multitask diffusion LMS, it can be observed that the performance is better than in the non-cooperative case. Finally, providing prior information to the clustered multitask network via an appropriate definition of clusters leads to the best performance. Clustering strategies are not discussed in this paper. This will be investigated in future work. One strategy is proposed in [32].

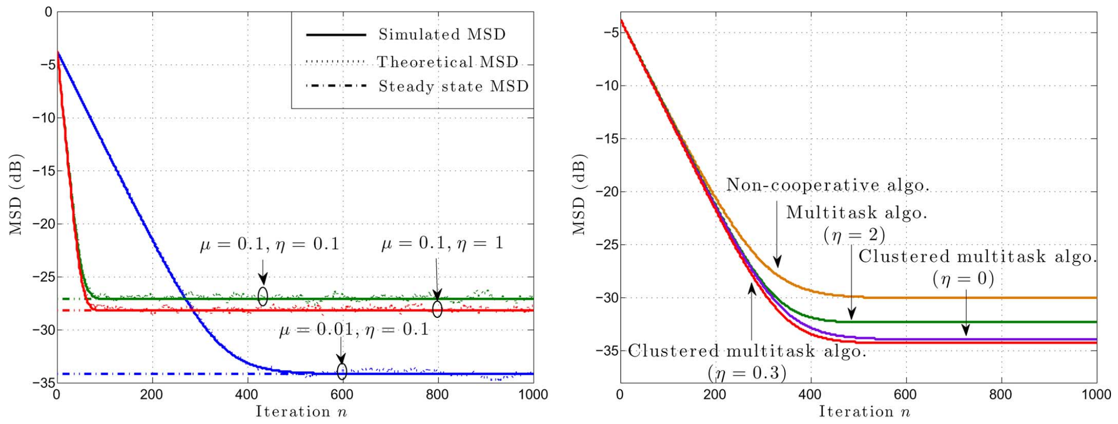
B. Distributed Non-Point Target Localization

The second application addresses the problem of target localization. Existing localization methods based on the diffusion strategy assume point targets [3], [59]. However, in some situations, targets may not be reduced to a single point such as its centroid. For instance, this includes the case where the target is a region of interest scanned by a laser light sheet. The algorithm should be able to jointly estimate a series of coordinates that characterizes the target area.

The problem we considered is shown in Fig. 4. The target was the arc of a circle with center $\boldsymbol{w}_o$. The angular resolution of the nodes was denoted by δ. This means that arcs of the circle with solid angle δ were viewed as a single point $\boldsymbol{w}_q$ by the cluster $\mathcal{C}_q$ of nodes within the cone of axis $(\boldsymbol{w}_o, \boldsymbol{w}_q)$. Note that the distance between each node $k \in \mathcal{C}_q$ and $\boldsymbol{w}_q$ can be expressed in the inner product form

$$r_{kq} = \boldsymbol{u}_{kq}^\top(\boldsymbol{w}_q - \boldsymbol{p}_k) \tag{82}$$

where $\boldsymbol{p}_k$ is the location of node k, and $\boldsymbol{u}_{kq}$ is the unit-norm vector pointing from $\boldsymbol{p}_k$ to $\boldsymbol{w}_q$. We assumed that sensors were



(a) (b)

Fig. 3. Network performance illustration. Left: transient and steady-state MSD (model vs. Monte Carlo) for different step-sizes and regularization parameters. Right: performance comparison for different strategies using theoretical models.

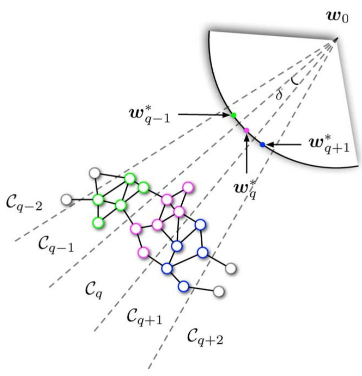


Fig. 4. Target surface localization.

aware of their location $\boldsymbol{p}_k$. Let $d_{kq} = r_{kq} + \boldsymbol{u}_{kq}^\top \boldsymbol{p}_k$, that is, $d_{kq} = \boldsymbol{u}_{kq}^\top \boldsymbol{w}_q$. The problem was thus to estimate $\boldsymbol{w}_q^\star$ from noisy input-output data $(\boldsymbol{u}_{kq}(n), d_{kq}(n))$ collected by nodes $k \in \mathcal{C}_q$. The model that was thus considered is given by [3]:

$$\begin{aligned} & d_{kq}(n) = \boldsymbol{u}_{kq}^\top(n)\, \boldsymbol{w}_q^\star + v_{kq}(n) \\ \text{with} \quad & \boldsymbol{u}_{kq}(n) = \boldsymbol{u}_{kq} + \alpha_k(n)\, \boldsymbol{u}_{kq}^\perp + \beta_k(n)\, \boldsymbol{u}_{kq} \end{aligned} \tag{83}$$

with $v_{kq}(n)$ a zero-mean temporally and spatially i.i.d. Gaussian noise of variance σ_v^2. Moreover, the measured direction $\boldsymbol{u}_{kq}(n)$ was assumed to be a noisy realization of the unit-norm vector pointing from $\boldsymbol{p}_k$ to $\boldsymbol{w}_q^\star$, with $\alpha_k(n)$ and $\beta_k(n)$ two Gaussian random variables of variances σ_α^2 and σ_β^2, respectively.

The multitask algorithm (33) was used to estimate the coordinates $\boldsymbol{w}_q^\star$ for $q \in \{1, \ldots, Q\}$, and to approximate the arc of radius R. Each node was connected to its neighbors within its cluster and the adjacent clusters. We considered two network topologies. In the first scenario, see the left-hand plot in Fig. 5 (first row), 100 nodes ranging from $3R$ to $4R$ were grouped into 10 clusters, with 10 nodes in each. The nodes were deployed uniformly with connections between neighbors. In the second scenario, see the right-hand plot in Fig. 5 (first row), 200 nodes ranging from $3R$ to $4R$ were grouped into 10 clusters, with 20 nodes in each cluster. The nodes were deployed randomly. For both experiments, the noise variances were set as follows: $\sigma_v^2 = 0.5$, $\sigma_\alpha^2 = 0.1$, and $\sigma_\beta^2 = 0.01$. We used an identity information exchange matrix $\boldsymbol{C} = \boldsymbol{I}$. The combination matrix $\boldsymbol{A}$ was defined as $a_{\ell k} = |\mathcal{N}_k \cap \mathcal{C}(k)|^{-1}$ in order to average the estimates of within-cluster neighbors. The regularization strengths $\rho_{k\ell}$ were set to $\rho_{k\ell} = |\mathcal{N}_k \backslash \mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k \backslash \mathcal{C}(k)$, with $k \neq 1$ and $k \neq Q$. Recall that $\mathcal{C}_1$ and $\mathcal{C}_Q$ are boundary clusters, and the specific regularization strengths $\rho_{1\ell} = \rho_{Q\ell} = 0$ for all ℓ were used to preserve the configuration of the group.

We ran the non-cooperative algorithm, and the clustered multitask algorithm with $\eta = 0.5$ and $\eta = 0.0005$ for each scenario, respectively. Fig. 5 (second row) shows one realization of the estimated points $\boldsymbol{w}_q$ for each arc. The cooperative algorithm clearly outperformed the non-cooperative algorithm. Fig. 5 (third row) compares the MSD of the two strategies mentioned above, with the clustered multitask algorithm with $\eta = 0$. In this case, the diffusion strategy is applied independently in each cluster, without inter-cluster interactions. This experiment clearly illustrates the advantage of fully cooperative strategies in this problem.

C. Distributed Unmixing of Hyperspectral Data

Finally, we consider the problem of distributed unmixing of hyperspectral images using the multitask learning algorithm. Hyperspectral imaging provides 2-dimensional spatial images over many contiguous bands. The high spectral resolution allows to identify and quantify distinct materials from remotely observed data. In hyperspectral images, a pixel is usually a spectral mixture of several spectral signatures of pure materials, termed endmembers, due to limited spatial resolution of devices and diversity of materials [60]. Although nonlinear mixture models have begun to support novel applications [61]–[63], the linear mixture model is still widely used for determining and quantifying materials in sensed images due to its simpler physical interpretation. With the linear mixture model, pixels can be decomposed as linear combinations of constituent spectra, weighted by fractions of abundance.

To facilitate the presentation, we shall consider that the 3-dimensional hyperspectral image under study has been reshaped

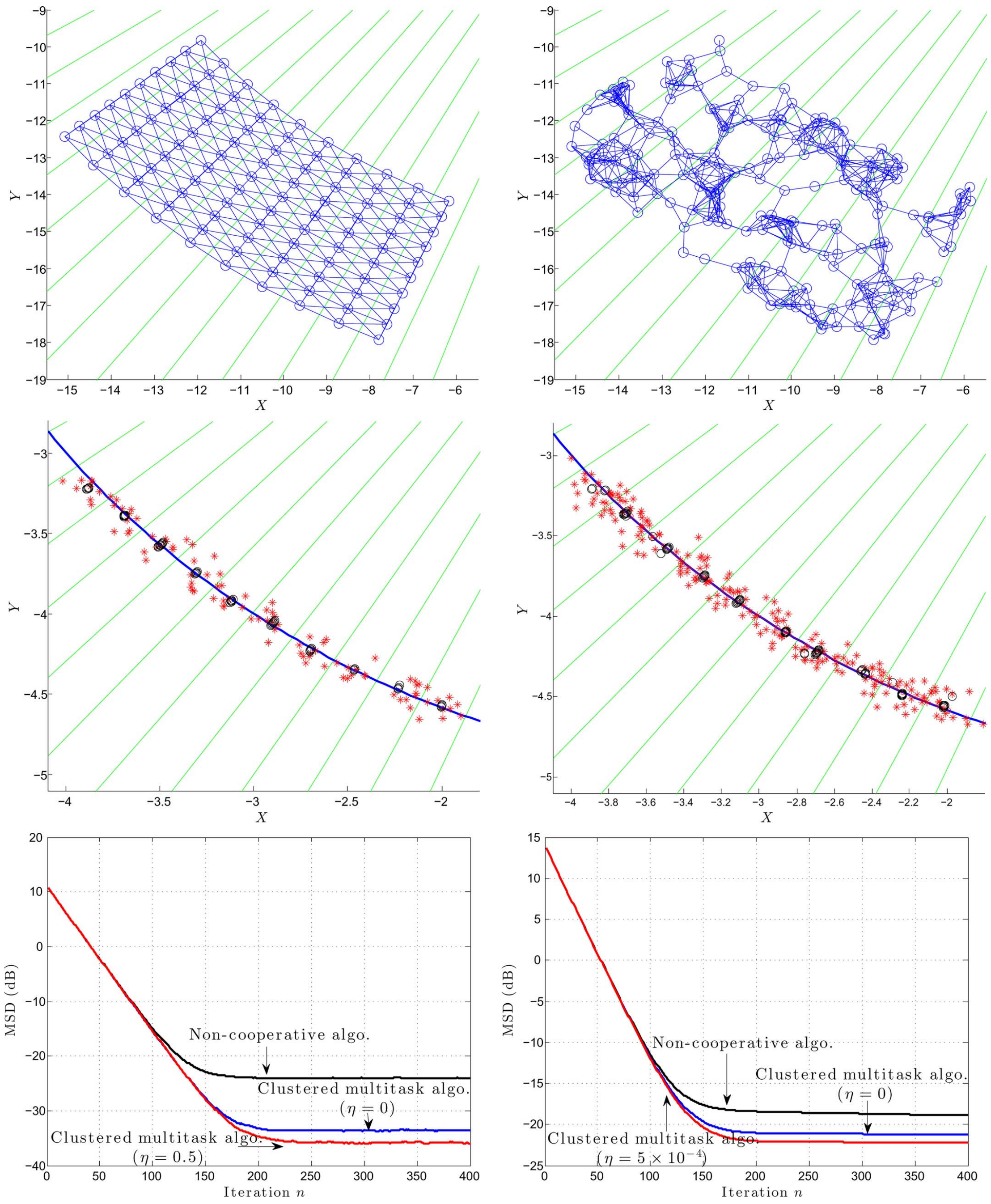


Fig. 5. Target surface localization. Left: uniform network. Right: randomly-distributed network. Row 1: network connectivity, with cluster boundaries in green. Row 2: estimation results, red crosses for the non-cooperative algorithm, black circles for the cooperative algorithm. Row 3: MSD learning curves.

into an $L \times N$ matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]$, with N the number of pixels and L the number of wavelengths. Let $\boldsymbol{M}$ be the $L \times R$ matrix of endmember spectra, with R the number of endmembers, and $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N]$ the $R \times N$ matrix of the abundance vectors of the pixels in $\boldsymbol{Y}$. The linear mixture model is expressed by

$$\boldsymbol{Y} = \boldsymbol{M}\boldsymbol{W} + \boldsymbol{V} \tag{84}$$

where $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$ is the modeling error matrix. Suppose that the material signatures (matrix $\boldsymbol{M}$) in a scene have been determined by some extraction algorithm [64]–[66]. The unmixing problem boils down to estimating the abundance vector associated with each pixel. Besides minimizing the modeling error, it is important to promote similarities of abundance vectors between neighboring pixels due to their possible correlations. Now we write the unmixing problem as follows:

$$\min_{\boldsymbol{W}} \|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{W}\|_F^2 + \eta \sum_{k=1}^{N} \sum_{j \in \mathcal{N}_k} \rho_{kj} \|\boldsymbol{w}_k - \boldsymbol{w}_j\|_1$$
$$\text{subject to} \;\; \boldsymbol{w}_k \succcurlyeq 0 \text{ and } \boldsymbol{1}^\top \boldsymbol{w}_k = 1, \text{ with } 1 \leq k \leq N, \tag{85}$$

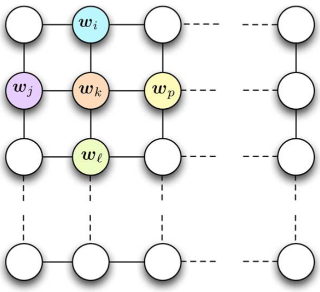


Fig. 6. Hyperspectral image unmixing problem with first-order connections between neighboring nodes.

where $\|\cdot\|_F^2$ is the matrix Frobenius norm, $\mathcal{N}_k$ is the set of neighbors of pixel k, η is the spatial regularization parameter and ρ_{kj} is the regularization weights. In the above expression, the nonnegativity constraints and sum-to-one constraints are imposed to ensure physical interpretability of the vectors of fractional abundances.

To conduct linear unmixing of large images in a distributed way, we considered each sensor of the camera as a node, and we applied the diffusion LMS for multitask problems, that is, one node per cluster – see Fig. 6. In order to exploit the spatial correlations, we defined the regularization function $\Delta(\boldsymbol{w}_k, \boldsymbol{w}_j)$ as the ℓ_1-norm of $\boldsymbol{w}_k - \boldsymbol{w}_j$ to promote piecewise constant transitions in the fractional abundance of each endmember among neighboring pixels. Similar regularization can be found in [67], [68]. This led us to the following algorithm:

$$\boldsymbol{w}_k(n+1) = \mathcal{P}_{\ell_1^+}\Bigg(\boldsymbol{w}_k(n) + \mu\,\boldsymbol{M}^\top(\boldsymbol{y}_k - \boldsymbol{M}\boldsymbol{w}_k(n)) - \mu\,\eta \sum_{j\in\mathcal{N}_k} \rho_{kj}\,\mathrm{sgn}(\boldsymbol{w}_k(n) - \boldsymbol{w}_j(n))\Bigg) \quad (86)$$

where we used that the subgradient $\partial_{\boldsymbol{x}}\|\boldsymbol{x}\|_1 = \mathrm{sgn}(\boldsymbol{x})$, with $\mathrm{sgn}(\cdot)$ the component-wise sign function. In this expression, $\mathcal{P}_{\ell_1^+}(\cdot)$ denotes the iterative operator defined in [69] that projects a vector onto the nonnegative phase of the ℓ_1-ball to satisfy the nonnegativity and sum-to-one constraint in (85). This algorithm clearly contrasts with existing batch approaches based on FISTA [70] and ADMM [71], which cannot easily address large problems (84).

The algorithm (86) was run on a data cube containing 100 × 100 mixed pixels. Each pixel was generated by the linear mixture model (85) using 9 endmember signatures randomly selected from the spectral library ASTER [72]. Each signature of this library has reflectance values measured over 224 spectral bands, uniformly distributed in the interval 3 – 12 μm. The abundance maps of the endmembers are the same as for the image DC2 in [71]. Among these 9 materials, only the 1st, 6th, 8th, and 9th abundances are considered for pictorial illustration in Fig. 8. The first row of this figure depicts the true distribution of these 5 materials. Spatially homogeneous areas with sharp transitions can be clearly observed. The generated scene was corrupted by a zero-mean white Gaussian noise $\boldsymbol{v}_n$ with an SNR level of 20 dB. In this experiment, the regularization weights ρ_{kj} were set equal to the normalized spectral similarity: $\rho_{kj} = \theta(\boldsymbol{y}_k, \boldsymbol{y}_j)/\sum_{\ell\in\mathcal{N}_k^-}\theta(\boldsymbol{y}_k, \boldsymbol{y}_\ell)$, where $\theta(\boldsymbol{y}_k, \boldsymbol{y}_j) = \boldsymbol{y}_k^\top \boldsymbol{y}_j / \|\boldsymbol{y}_k\|\|\boldsymbol{y}_j\|$. These weights emphasize the regularization between similar pixels and de-emphasize it for less similar pixels. When one knows the ground truth map, a commonly used performance measure for evaluating the performance of an unmixing algorithm is the root mean-square error (RMSE), defined as

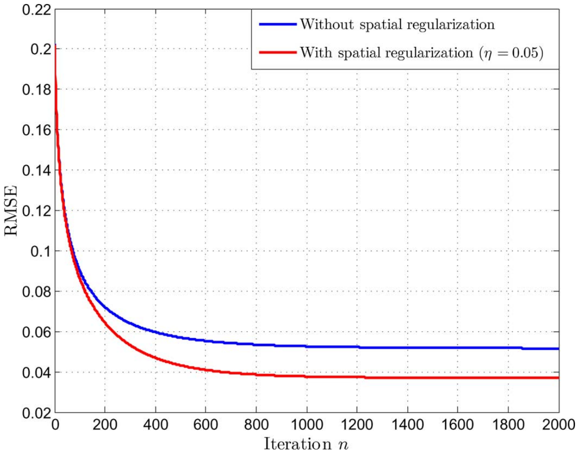


Fig. 7. RMSE curve comparison.

$$\mathrm{RMSE} = \sqrt{\frac{1}{NR}\sum_{n=1}^{N}\|\boldsymbol{w}_n - \boldsymbol{w}_n^\star\|^2}.$$

The RMSE learning curves using algorithm (86), with spatial regularization ($\eta = 0.05$) and without spatial regularization ($\eta = 0$), are depicted in Fig. 7. The corresponding abundance distributions are shown in Fig. 8. The spatial regularization results in a lower estimation error, and more homogenous abundance distribution maps with less noise.

VII. Conclusion and Perspectives

In this paper, we formulated multi-task problems where networks are able to handle situations beyond the case where the nodes estimate a unique parameter vector over the network. Considering each parameter vector estimation as a task, and possibly connecting these tasks in order that they can share information, we extended the distributed learning problem from single-task learning to clustered multitask learning. An algorithm was derived. A mean behavior analysis of the proposed algorithm was provided, in the case of the least-mean-square error criterion with ℓ_2-norm regularization. Several applications that may benefit from this framework were investigated. Several open problems still have to be solved for specific applications. For instance, it would be interesting to show which regularization can be advantageously used with our distributed multitask algorithm, and how they can be efficiently implemented in an adaptive manner. It would also be interesting to investigate how nodes can autonomously adjust regularization parameters

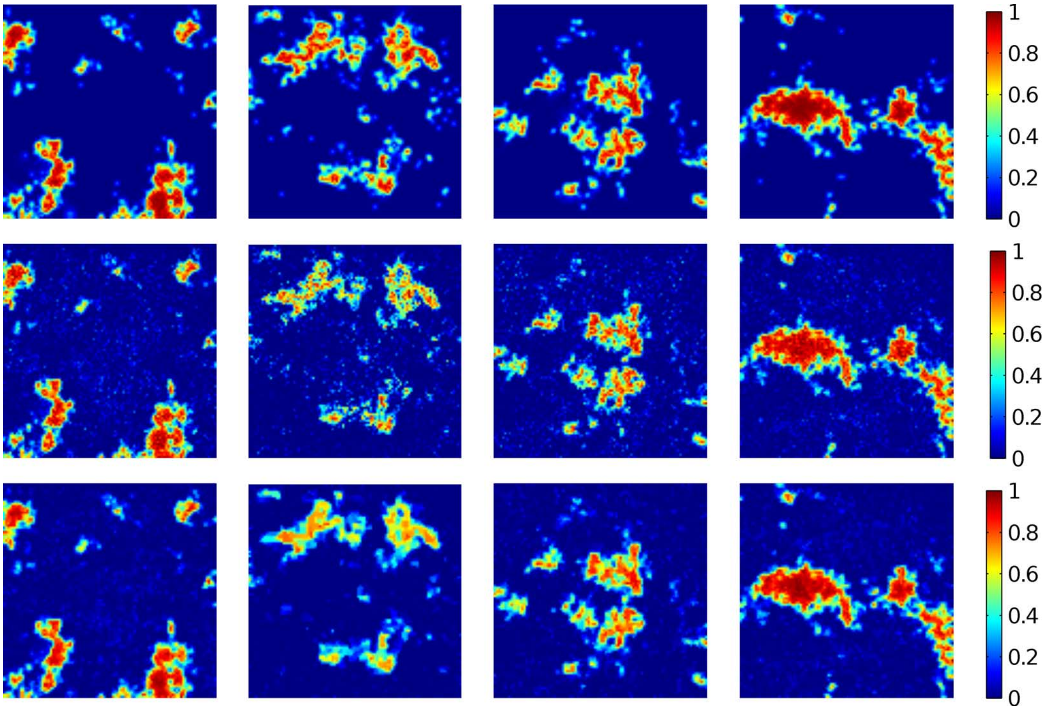


Fig. 8. Abundance maps. From left to right: 1st, 6th, 8th, and 9th abundances. From top to bottom: true abundances, estimated abundances without and with spatial regularization.

to optimize the learning performance and how they can learn the structure of the clusters in real-time.

References

[1] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS for clustered multitask networks," in *Proc. IEEE ICASSP*, Florence, Italy, May 2014, pp. 5524–5528.

[2] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Sig. Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[3] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Libraray in Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Amsterdam, The Netherlands: Elsevier, 2014, pp. 322–454.

[4] A. H. Sayed, "Adaptive networks," *Proc. of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[5] J. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Trans. Autom. Control*, vol. 29, no. 1, pp. 42–50, Jan. 1984.

[6] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 9, pp. 65–78, Sep. 2004.

[7] P. Braca, S. Marano, and V. Matta, "Running consensus in wireless sensor networks," in *Proc. FUSION*, Cologne, Germany, Jun.-Jul. 2008, pp. 1–6.

[8] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[9] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[10] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

[11] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optimiz.*, vol. 7, no. 4, pp. 913–926, Nov. 1997.

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

[13] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[14] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with constant step size," *SIAM J. Optimiz.*, vol. 18, no. 1, pp. 29–51, Feb. 2007.

[15] 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.

[16] 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, Jul. 2008.

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

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

[19] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, Apr. 2013.

[20] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[21] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers, "Steady-state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 974–979, Feb. 2012.

[22] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3460–3475, Jul. 2012.

[23] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4480–4485, Aug. 2012.

[24] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity-promoting adaptive algorithm for distributed learning," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.

[25] P. D. Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[26] S. Chouvardas, K. Kalvakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[27] F. 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.

[28] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion bias-compensated RLS estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5212–5224, Nov. 2011.

[29] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network," in *Proc. IEEE CAMSAP*, Saint Martin, France, Dec. 2013, pp. 1–4.

[30] W. Wee and I. Yamada, "A proximal splitting approach to regularized distributed adaptive estimation in diffusion network," in *Proc. IEEE ICASSP*, Vancouver, Canada, May 2013, pp. 5420–5424.

[31] S.-Y. Tu and A. H. Sayed, "Distributed decision-making over adaptive networks," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1054–1069, Mar. 2014.

[32] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. CIP*, Parador de Baiona, Spain, May 2012, pp. 1–6.

[33] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific parameter estimation over adaptive networks," in *Proc. IEEE ICASSP*, Vancouver, BC, Canada, May 2013, pp. 5425–5429.

[34] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for leaning shared structures from muliple tasks," in *Proc. ICML*, Montreal, QC, Canada, Jun. 2009, pp. 137–144.

[35] O. Chapelle, P. Shivaswmy, K. Q. Vadrevu, S. Weinberger, Y. Zhang, and B. Tseng, "Multi-task learning for boosting with application to web search ranking," in *Proc. ACM SIGKDD*, Washington, DC, USA, Jul. 2010, pp. 1189–1198.

[36] J. Zhou, L. Yuan, J. Liu, and J. Ye, "A multi-task learning formulation for predicting disease progression," in *Proc. ACM SIGKDD*, San Diego, CA, USA, Aug. 2011, pp. 814–822.

[37] J. Chen and C. Richard, "Performance analysis of diffusion LMS in multitask networks," in *Proc. IEEE CAMSAP*, Saint Martin, France, Dec. 2013, pp. 1–4.

[38] C. Jiang, Y. Chen, and K. J. Ray Liu, "Distributed adaptive networks: A graphical evolutionary game-theoretic view," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5675–5688, Nov. 2013.

[39] J. Predd, S. Kulkarni, and H. Vincent Poor, "Distributed learning in wireless sensor networks," *IEEE Sig. Process. Mag.*, vol. 23, no. 4, pp. 59–69, Jul. 2006.

[40] C. Richard, J.-C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1058–1067, 2009.

[41] P. Honeine, C. Richard, and J.-C. M. Bermudez, "Online nonlinear sparse approximation of functions," in *Proc. IEEE ISIT*, Nice, France, Jun. 2007, pp. 956–960.

[42] P. Honeine, C. Richard, H. Snoussi, J.-C. M. Bermudez, and J. Chen, "A decentralized approach for non-linear prediction of time series data in sensor networks," *EURASIP J. Wirel. Comm.*, vol. 2010, no. 1, p. 6247372, Apr. 2010.

[43] P. Honeine, C. Richard, J.-C. M. Bermudez, H. Snoussi, M. Essoloh, and F. Vincent, "Functional estimation in Hilbert space for distributed learning in wireless sensor networks," in *Proc. IEEE ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 2861–2864.

[44] P. Honeine, M. Essoloh, C. Richard, and H. Snoussi, "Distributed regression in sensor networks with a reduced-order kernel model," in *Proc. IEEE GLOBECOM*, New Orleans, LA, USA, 2008, pp. 1–5.

[45] P. Honeine, C. Richard, J.-C. M. Bermudez, and H. Snoussi, "Distributed prediction of time series data with kernels and adaptive filtering techniques in sensor networks," in *Proc. ASILOMAR*, Pacific Grove, CA, USA, 2008, pp. 246–250.

[46] J. Chen, C. Richard, P. Honeine, and J.-C. M. Bermudez, "Non-negative distributed regression for data inference in wireless sensor networks," in *Proc. ASILOMAR*, Pacific Grove, CA, USA, Nov. 2010, pp. 451–455.

[47] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *Proc. ACM SIGKDD*, Seattle, WA, USA, Aug. 2004, pp. 1–9.

[48] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proc. ICML*, Prague, Czech Republic, Aug. 2009, pp. 457–464.

[49] J. Zhou, J. Chen, and J. Ye, J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., "Clustered multi-task learning via alternating structure optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, vol. 24, pp. 702–710.

[50] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. London: Academic press, 1995.

[51] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.

[52] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[53] J. Chen, C. Richard, J.-C. M. Bermudez, and P. Honeine, "Nonnegative least-mean-square algorithm," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5225–5235, Nov. 2011.

[54] Z. Towfic and A. H. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *IEEE Trans. Signal Process.*, vol. 62, 2014, to be published.

[55] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems*, 2nd ed. Waltham, MA, USA: Academic Press, 2005.

[56] F. Bach, R. Jenatton, J. Marial, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends® in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

[57] L. I. Rudin, O. Stanley, and F. Emad, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[58] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2008.

[59] S.-Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 649–664, Aug. 2011.

[60] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Sig. Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan. 2002.

[61] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approches," *IEEE J. Sel. Topics Appl. Earth Observ.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

[62] J. Chen, C. Richard, and P. Honeine, "Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 480–492, Jan. 2013.

[63] J. Chen, C. Richard, and P. Honeine, "Estimating abundance fractions of materials in hyperspectral images by fitting a post-nonlinear mixing model," in *Proc. IEEE WHISPERS*, Gainesville, FL, USA, Jun. 2013, pp. 1–4.

[64] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Proc. SPIE 3753, Imaging Spectrometry V*, Oct. 1999, vol. 266, pp. 266–275.

[65] J. M. P. Nascimento and J. M. Bioucas-Dias, "Vertex Component Analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.

[66] P. Honeine and C. Richard, "Geometric unmixing of large hyperspectral images: A barycentric coordinate approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 6, pp. 2185–2195, Jun. 2012.

[67] J. Chen, C. Richard, and P. Honeine, "Nonlinear estimation of material abundances in hyperspectral images with ℓ_1-norm spatial regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2654–2655, May 2014.

[68] J. Chen, C. Richard, A. Ferrari, and P. Honeine, "Nonlinear unmixing of hyperspectral data with partially linear least-squares support vector regression," in *Proc. IEEE ICASSP*, Vancouver, BC, Canada, May 2013, pp. 2174–2178.

[69] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proc. ICML*, Helsinki, Finland, Jul. 2008, pp. 272–279.

[70] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[71] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Total variation spatial regularization for sparse hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4484–4502, Nov. 2012.

[72] A. M. Baldridge, S. J. Hook, C. I. Grove, and G. Rivera, "The ASTER spectral library version 2.0," *Remote Sens. Environ.*, vol. 113, no. 4, pp. 711–715, Apr. 2009.

Jie Chen (S'12–M'14) was born in Xi'an, China, in 1984. He received the B.S. degree in information and telecommunication engineering in 2006 from the Xi'an Jiaotong University, Xi'an, and the Dipl.-Ing. and the M.S. degrees in information and telecommunication engineering in 2009 from the University of Technology of Troyes (UTT), Troyes, France, and from the Xi'an Jiaotong University, respectively. In 2013, he received the Ph.D. degree in systems optimization and security from the UTT.

From April 2013 to March 2014, he was a Postdoctoral Research Fellow at the Côte d'Azur Observatory, University of Nice Sophia Antipolis, Nice, France. Since April 2014, he has been a Postdoctoral Research Fellow with the department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, USA.

His current research interests include adaptive signal processing, supervised and unsupervised learning, distributed optimization, hyperspectral image analysis, and bio-signal processing.

Cédric Richard (S'98–M'01–SM'07) was born January 24, 1970 in Sarrebourg, France. He received the Dipl.-Ing. and the M.S. degrees in 1994 and the Ph.D. degree in 1998 from the University of Technology of Compiègne, France, all in electrical and computer engineering. From 1999 to 2003, he was an Associate Professor at the University of Technology of Troyes (UTT), France. From 2003 to 2009, he was a Full Professor at UTT. Since september 2009, he is a Full Professor in the Lagrange Laboratory (University of Nice Sophia Antipolis, CNRS, Observatoire de la Côte d'Azur). In Winter 2009 and 2014, and Fall 2010, 2011 and 2013, he was a Visiting Researcher with the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianopolis, Brazil. He is a junior member of the Institut Universitaire de France since October 2010.

His current research interests include statistical signal processing and machine learning.

Cédric Richard is the author of over 220 papers. He was the General Chair of the XXIth Francophone conference GRETSI on Signal and Image Processing that was held in Troyes, France, in 2007, and of the IEEE Statistical Signal Processing Workshop (IEEE SSP'11) that was held in Nice, France, in 2011. He will be the Technical Chair of EUSIPCO 2015. Since 2005, he is a member of GRETSI association board and of the EURASIP society, and Senior Member of the IEEE. In 2006–2010, he served as an associate editor of the IEEE Transactions on Signal Processing. Currently, he serves as an Associate Editor of Signal Processing Elsevier, and of the IEEE Signal Processing Letters. He is an Eurasip liaison local officer. He is a member of the Signal Processing Theory and Methods (SPTM TC) Technical Committee, and of the Machine Learning for Signal Processing (MLSP TC) Technical Committee, of the IEEE Signal Processing Society.

Paul Honeine and and Cédric Richard won the Best Paper Award for "Solving the preimage problem in kernel machines: a direct method" at the 2009 IEEE International Workshop on Machine Learning for Signal Processing.

Ali H. Sayed (S'90–M'92–SM'99–F'01) is Professor and former Chairman of electrical engineering at the University of California, Los Angeles, where he directs the UCLA Adaptive Systems Laboratory. His research activities involve several areas including adaptation and learning, network science, information processing theories, distributed processing, statistical data analysis, and biologically-inspired designs. He is coauthor of the textbook Linear Estimation (Englewood Cliffs, NJ: Prentice Hall, 2000), of the research monograph Indefinite Quadratic Estimation and Control (Philadelphia, PA: SIAM, 1999), and coeditor of Fast Algorithms for Matrices with Structure (Philadelphia, PA: SIAM, 1999). He is also the author of the textbooks Fundamentals of Adaptive Filtering (Hoboken, NJ: Wiley, 2003), and Adaptive Filters (Hoboken, NJ: Wiley, 2008). He has contributed several encyclopedia and handbook articles.

Prof. Sayed is also a Fellow of the American Association for the Advancement of Science (AAAS). His work has been recognized with several awards including the 2014 Athanasios Papoulis Award from the European Association for Signal Processing, the 2013 Meritorious Service Award and the 2012 Technical Achievement Award from the IEEE Signal Processing Society, the 2005 Terman Award from the American Society for Engineering Education, the 2003 Kuwait Prize, and the 1996 Donald G. Fink Prize from IEEE. He served as a Distinguished Lecturer of the IEEE Signal Processing Society in 2005. His articles received best paper awards from the IEEE Signal Processing Society in 2002, 2005, and 2012. He has been active in serving the Signal Processing community in various roles. Among other activities, he served as Editor-in-Chief of the IEEE Transactions on Signal Processing (2003–2005), Editor-in-Chief of the EURASIP J. on Advances in Signal Processing (2006– 2007), General Chairman of ICASSP (Las, Vegas, 2008), and Vice-President of Publications of the IEEE Signal Processing Society (2009–2011). He also served as member of the Board of Governors (2007–2011), Awards Board (2005), Publications Board (2003–2005), Conference Board (2007–2011), and Technical Directions Board (2008–2009) of the same Society.