

# Multitask Diffusion Adaptation over Networks

Jie Chen<sup>†</sup>, *Student Member, IEEE*, Cédric Richard<sup>†</sup>, *Senior Member, IEEE*

Ali H. Sayed<sup>‡</sup>, *Fellow Member, IEEE*

<sup>†</sup> Université de Nice Sophia-Antipolis, UMR CNRS 7293, Observatoire de la Côte d'Azur

Laboratoire Lagrange, Parc Valrose, 06102 Nice - France

phone: (33) 492 076 394      fax: (33) 492 076 321

jie.chen@unice.fr      cedric.richard@unice.fr

<sup>‡</sup> Electrical Engineering Department

University of California, Los Angeles, USA

phone: (310) 267 2142      fax: (310) 206 8495

sayed@ee.ucla.edu

**EDICS:** NET-ADEG, NET-DISP, MLR-DIST, SSP-PERF

## Abstract

Adaptive networks are suitable for decentralized inference tasks, e.g., to monitor complex natural phenomena. Recent research 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  $\ell_2$ -regularization. The stability and convergence of the algorithm in the mean and in the mean-square sense is 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 spectral sensing, target localization, and hyperspectral data unmixing.

## Index Terms

Multitask learning, distributed optimization, diffusion strategy, collaborative processing, asymmetric regularization, spectral sensing, target localization, data unmixing.

This work was partly supported by the Agence Nationale pour la Recherche, France, (Hypanema project, ANR-12-BS03-003), and the Centre National de la Recherche Scientifique, France (Display project, Mastodons). The work of A. H. Sayed was supported in part by NSF grant CCF-1011918.

## 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 [1], [2]. In adaptive networks, the interconnected nodes have to continually learn and adapt, as well as perform preassigned tasks such as parameter estimation from observations collected by the dispersed agents. Although centralized strategies with a fusion center can benefit more fully from information collected throughout the network but stored at a single point, in most cases, distributed strategies are more attractive to solve inference problems in a collaborative and autonomous manner. Scalability, robustness, and low-power consumption are key characteristics of these strategies. Applications include environment monitoring, but also modeling of self-organized behavior observed in nature such as bird flight in formation and fish schooling [1], [3].

There are several useful distributed strategies for sequential data processing over networks including consensus strategies [4]–[9], incremental strategies [10]–[14], and diffusion strategies [1], [2], [15]–[18]. 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 [19] than consensus-based implementations. Consequently, we shall focus on diffusion-type implementations in the sequel. The diffusion LMS strategy was proposed and studied in [15], [16]. Its performance in the presence of imperfect information exchange and model non-stationarity was analyzed in [20]–[22]. Diffusion LMS with  $\ell_1$ -norm regularization was considered in [23]–[26] to promote sparsity in the model. In [27], the problem of distributed learning in diffusion networks was addressed by deriving projection algorithms onto convex sets. Diffusion RLS over adaptive networks was studied in [28], [29]. More recently, a distributed dictionary learning algorithm based on a diffusion strategy was derived in [30]–[32]. This literature mainly considers quadratic cost functions and linear models where systems are characterized by a parameter vector in the Euclidean space. Extensions to more general cost functions that are not necessarily quadratic and to more general data models are studied in [17], [18] in the context of adaptation and learning over networks. Moreover, several other works explored distributed estimation for nonlinear input-output relationships defined in a functional space, such as reproducing kernel Hilbert spaces. For instance, in [33], inference is performed with a regularized kernel least-squares estimator, where the distributed information-sharing strategy consisted of successive orthogonal projections. Distributed estimation based on adaptive kernel regression [34], [35] is also studied in [36]–[39], with information passed from node to node in an incremental manner. In [40], non-negative distributed regression is considered for nonlinear model inference subject to non-negativity constraints, where the diffusion strategy is used to conduct information exchange.

An inspection of the existing literature on distributed algorithms shows that most works focus primarily, though not exclusively [41]–[43], 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 been studied in the machine learning community in several contexts. For example, the problem finds applications in web page categorization [44], web-search ranking [45], and disease progression modeling [46], 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 [42], [47]. 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. Obviously, a limit case of this problem is the situation where all clusters are of equal size one, that is, each node has its own parameter vector to estimate but shares information with its neighbors. Another limit case is when the size of the cluster agrees with the size of the network in which case all nodes have the same parameter vector to estimate. The aim of this paper is to derive diffusion strategies that are able to solve this general multitask estimation problem, and to analyze their performance in terms of mean-square error and convergence rate. Simulations are also conducted to illustrate the theoretical analysis, and to apply the algorithms to three useful applications involving spectral sensing, target localization, and hyperspectral data unmixing.

This paper is organized as follows. Section II formulates the distributed estimation problem for multitask learning. Section III presents a relaxation strategy for optimizing local cost functions over the network. Section IV derives a stochastic gradient algorithm for distributed adaptive learning in a multitask-oriented environment. Section V analyzes the theoretical performance of the proposed algorithm, in the mean and mean-square-error sense. In Section VI, experiments and applications are presented to illustrate the performance of the approach. Section VII concludes this paper and gives perspectives on future work.

## II. NETWORK MODELS AND MULTITASK LEARNING

Before starting our presentation, we provide a summary of some of the main symbols used in the article. Other symbols will be defined in the context where they are used:

$x$	Normal font denotes scalars.
$\mathbf{x}$	Boldface small letters denote vectors. All vectors are column vectors.
$\mathbf{R}$	Boldface capital letters denote matrices.
$(\cdot)^\top$	Matrix transpose.
$\mathbf{I}_N$	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., $\mathcal{C}(k) = \{\mathcal{C}_i : k \in \mathcal{C}_i\}$ .
$J(\cdot), \bar{J}(\cdot)$	Cost functions without/with regularization.
$\mathbf{w}^*, \mathbf{w}^o$	Optimum parameter vectors without/with regularization.

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  $\{d_k(n), \mathbf{x}_k(n)\}$ , with  $d_k(n)$  denoting a scalar zero-mean reference signal, and  $\mathbf{x}_k(n)$  denoting an  $L \times 1$  regression vector with a positive-definite covariance matrix,  $\mathbf{R}_{\mathbf{x},k} = E\{\mathbf{x}_k(n)\mathbf{x}_k^\top(n)\} > 0$ . The data at node  $k$  are assumed to be related via the linear regression model:

$$d_k(n) = \mathbf{x}_k^\top(n) \mathbf{w}_k^* + z_k(n) \quad (1)$$

where  $\mathbf{w}_k^*$  is an unknown parameter vector at node  $k$ , and  $z_k(n)$  is a zero-mean i.i.d. 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 multitask oriented. We therefore distinguish among the following three types of networks, as illustrated by Figure 1, depending on how the parameter vectors  $\mathbf{w}_k^*$  across the nodes are related:

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

$$\mathbf{w}_k^* = \mathbf{w}^*, \quad \forall k \in \{1, \dots, N\} \quad (2)$$

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

$$\mathbf{w}_k^* \sim \mathbf{w}_\ell^* \quad \text{if } \ell \in \mathcal{N}_k \quad (3)$$

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. Within the area of machine learning, the relation between tasks can be promoted in several ways, e.g., through mean regularization [48], low rank regularization [49], or clustered regularization [50]. We note that 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 Figure 1(b) can represent neighboring relationships between adjacent pixels. We will consider this application in greater detail in the simulation section.

- 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,

$$\mathbf{w}_k^* = \mathbf{w}_{\mathcal{C}_q}^*, \quad \text{whenever } k \in \mathcal{C}_q \quad (4)$$

$$\mathbf{w}_{\mathcal{C}_p}^* \sim \mathbf{w}_{\mathcal{C}_q}^*, \quad \text{if } \mathcal{C}_p, \mathcal{C}_q \text{ are connected} \quad (5)$$

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 the 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,  $J_k(\mathbf{w}_{\mathcal{C}(k)})$ , 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

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

In order to promote similarities among adjacent clusters, appropriate regularization can be used. For this purpose, we introduce the squared Euclidean distance as a possible regularizer, namely,

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

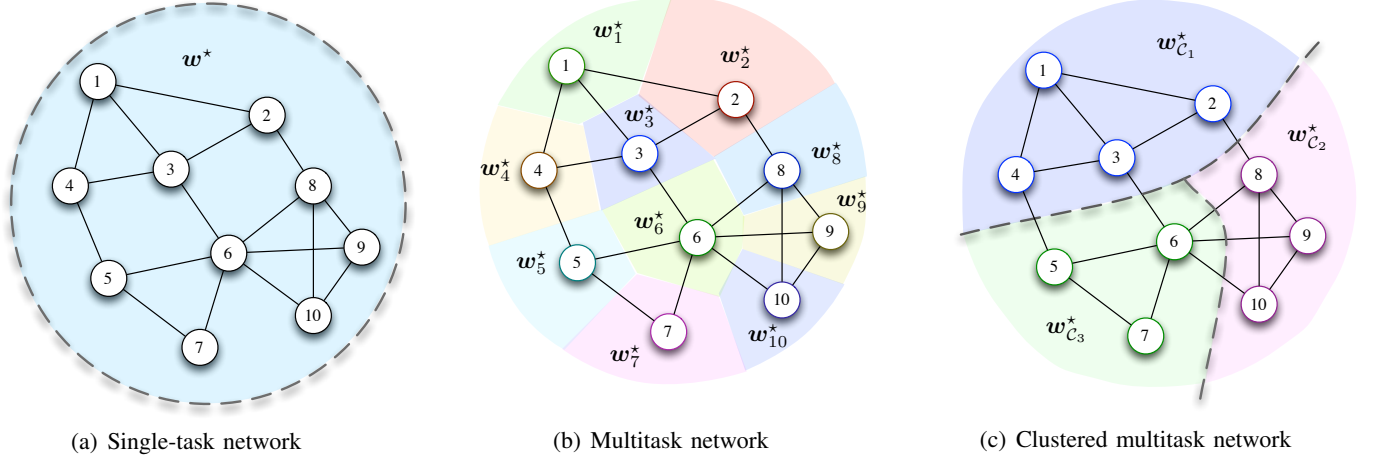


Fig. 1. Three types of networks. The single-task and multitask networks can be viewed as special cases of the clustered multitask network.

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

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

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

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  $\|\mathbf{w}_{\mathcal{C}(k)} - \mathbf{w}_{\mathcal{C}(\ell)}\|^2$  is symmetric with respect to the weight vectors  $\mathbf{w}_{\mathcal{C}(k)}$  and  $\mathbf{w}_{\mathcal{C}(\ell)}$ ; 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 exploit this flexibility more fully, we consider an alternative problem formulation  $\mathcal{P}_2$  defined in terms of  $Q$  Nash equilibrium problems as follows:

$$(\mathcal{P}_2) \quad \begin{cases} \min_{\mathbf{w}_{\mathcal{C}_i}} \overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i}) & \text{for } i = 1, \dots, Q \\ \text{with } \overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i}) = \sum_{k \in \mathcal{C}_i} E \left\{ |d_k(n) - \mathbf{x}_k^\top(n) \mathbf{w}_{\mathcal{C}(k)}|^2 \right\} + \eta \sum_{k \in \mathcal{C}_i} \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}_i} \rho_{k\ell} \|\mathbf{w}_{\mathcal{C}(k)} - \mathbf{w}_{\mathcal{C}(\ell)}\|^2 \end{cases} \quad (9)$$

where each cluster  $\mathcal{C}_i$  estimates  $\mathbf{w}_{\mathcal{C}_i}$  by minimizing  $\overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i})$ . Note that we have kept the notation  $\mathbf{w}_{\mathcal{C}(k)}$  to make the role of the regularization term clearer, even though in formulation (9) we have  $\mathbf{w}_{\mathcal{C}(k)} = \mathbf{w}_{\mathcal{C}_i}$  for all  $k$  in  $\mathcal{C}_i$ . In (9), the notation  $\mathbf{w}_{-\mathcal{C}_i}$  denotes the collection of weight vectors estimated by the other clusters, i.e.,  $\mathbf{w}_{-\mathcal{C}_i} = \{\mathbf{w}_{\mathcal{C}_k} : k = 1, \dots, Q\} - \{\mathbf{w}_{\mathcal{C}_i}\}$ .

The Nash equilibrium of  $\mathcal{P}_2$  satisfies the condition [51]:

$$\mathbf{w}_{\mathcal{C}_i}^o = \arg \min_{\mathbf{w}_{\mathcal{C}_i}} \overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i}^o) \quad (10)$$

for  $i = 1, \dots, Q$ , where the notation  $\mathbf{w}_{-\mathcal{C}_i}^o$  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  $\overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i})$  is convex with respect to  $\mathbf{w}_{\mathcal{C}_i}$  for all  $i$ .
- 2) The equilibrium for  $\mathcal{P}_2$  is unique since  $\{\overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i})\}_{i=1}^Q$  satisfies the diagonal strict convexity property.<sup>1</sup>
- 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 in [52]. Property 3) can be verified by the optimality conditions for the two problems.

Problem  $\mathcal{P}_1$  can be solved either analytically in closed form or iteratively by using a steepest-descent algorithm. 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 [51], that is,

$$\mathbf{w}_{\mathcal{C}_i}(n+1) = \arg \min_{\mathbf{w}_{\mathcal{C}_i}} \overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i}(n)) \quad (11)$$

for  $i = 1, \dots, Q$ , and leads to the solution of (9). Since the equilibrium is unique and the cost function for each cluster is convex, the solution of (9) can also be approached by means of a steepest-descent iteration as follows:

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

for  $i = 1, \dots, Q$ , with  $\nabla_{\mathbf{w}_{\mathcal{C}_i}}$  denoting the gradient operation with respect to  $\mathbf{w}_{\mathcal{C}_i}$ , and  $\mu$  a positive step-size. We have

$$\nabla_{\mathbf{w}_{\mathcal{C}_i}} \overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i}(n)) \propto \sum_{k \in \mathcal{C}_i} (\mathbf{R}_{x,k} \mathbf{w}_{\mathcal{C}_i} - \mathbf{p}_{x,d,k}) + \eta \sum_{k \in \mathcal{C}_i} \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}_i} \rho_{k\ell} (\mathbf{w}_{\mathcal{C}_i} - \mathbf{w}_{\mathcal{C}(\ell)}(n)). \quad (13)$$

where  $\mathbf{p}_{x,d,k} = E\{\mathbf{x}_k(n)d_k(n)\}$  is the input-output cross-correlation vector between  $\mathbf{x}_k(n)$  and  $d_k(n)$  at node  $k$ . If some additional constraints are imposed on the parameters to estimate, the gradient update relation can be modified using methods such as projection [53] or fixed point iteration 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. Since  $\mathcal{P}_1$  is equivalent to  $\mathcal{P}_2$  with proper setting of the weights  $\rho_{k\ell}$ , we shall now derive a distributed algorithm for solving problem  $\mathcal{P}_2$ . In this paper, we shall consider normalized weights that satisfy

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

### 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  $\mathbf{R}_{x,k}$  and  $\mathbf{p}_{x,d,k}$  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  $\{\mathbf{R}_{x,d}, \mathbf{p}_{x,d,k}\}$ ; 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 [2], [16]. 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.

<sup>1</sup>Let  $\mathbf{g}(\mathbf{w}, \boldsymbol{\zeta}) = [\zeta_i \nabla_{\mathbf{w}_{\mathcal{C}_i}} \overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i})]_{i=1}^Q$  arranged as a row vector with  $\zeta_i > 0$ . The cost functions  $\{\overline{\mathcal{J}}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i})\}_{i=1}^Q$  satisfy the diagonal strict convexity property if  $\mathbf{g}(\mathbf{w}, \boldsymbol{\zeta})$  is strictly decreasing in  $\mathbf{w}$  for some positive vector  $\boldsymbol{\zeta}$ , that is,  $(\mathbf{g}(\hat{\mathbf{w}}, \boldsymbol{\zeta}) - \mathbf{g}(\mathbf{w}, \boldsymbol{\zeta}))^\top (\hat{\mathbf{w}} - \mathbf{w}) < 0$  for all nonequal  $\mathbf{w}, \hat{\mathbf{w}}$ .

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

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

With these coefficients, we associate a local cost function of the following form with each node  $k$  [2]:

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

One important distinction from the local cost defined in [2] is that in [2] 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  $\mathbf{w}_{\mathcal{C}(k)} = \mathbf{w}_{\mathcal{C}(\ell)}$  because  $\ell \in \mathcal{C}(k)$ . To make the notation simpler, we shall write  $\mathbf{w}_k$  instead of  $\mathbf{w}_{\mathcal{C}(k)}$ . A consequence of this notation is that  $\mathbf{w}_k = \mathbf{w}_\ell$  for all  $\ell \in \mathcal{C}(k)$ . Incorporating the estimates of the neighboring clusters, we modify (16) to associate a regularized local cost function with node  $k$  of the following form

$$\overline{J}_k^{\text{loc}}(\mathbf{w}_k) = \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k} E \left\{ |d_\ell(n) - \mathbf{x}_\ell^\top(n) \mathbf{w}_k|^2 \right\} + \eta \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \rho_{k\ell} \|\mathbf{w}_k - \mathbf{w}_\ell\|^2. \quad (17)$$

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

$$\begin{aligned} \overline{J}_{\mathcal{C}_i}(\mathbf{w}_{\mathcal{C}_i}, \mathbf{w}_{-\mathcal{C}_i}) &= \sum_{k \in \mathcal{C}_i} \left( \sum_{\ell \in \mathcal{C}(k)} c_{\ell k} E \left\{ |d_\ell(n) - \mathbf{x}_\ell^\top(n) \mathbf{w}_{\mathcal{C}(k)}|^2 \right\} + \eta \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}_i} \rho_{k\ell} \|\mathbf{w}_{\mathcal{C}(k)} - \mathbf{w}_{\mathcal{C}(\ell)}\|^2 \right) \\ &= \sum_{k \in \mathcal{C}_i} \overline{J}_k^{\text{loc}}(\mathbf{w}_k) \\ &= \overline{J}_k^{\text{loc}}(\mathbf{w}_k) + \sum_{\ell \in \mathcal{C}(k) \setminus k} \overline{J}_\ell^{\text{loc}}(\mathbf{w}_\ell) \end{aligned} \quad (18)$$

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

$$\overline{J}_k^{\text{loc}}(\mathbf{w}_k) = \overline{J}_k^{\text{loc}}(\mathbf{w}_k^o) + \|\mathbf{w}_k - \mathbf{w}_k^o\|_{\overline{\mathbf{R}}_k}^2 \quad (19)$$

where

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

Substituting equation (19) into the second term on the right-hand-side of (18), and discarding the terms  $\{\overline{J}_\ell^{\text{loc}}(\mathbf{w}_\ell^o)\}$  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  $k$ :

$$\overline{J}_{\mathcal{C}(k)}(\mathbf{w}_k) \triangleq \overline{J}_k^{\text{loc}}(\mathbf{w}_k) + \sum_{\ell \in \mathcal{C}(k) \setminus k} \|\mathbf{w}_\ell - \mathbf{w}_k^o\|_{\overline{\mathbf{R}}_\ell}^2 \quad (21)$$

where it holds that  $\mathbf{w}_k = \mathbf{w}_\ell$  because  $\ell \in \mathcal{C}(k)$ . Note that we have omitted  $\mathbf{w}_{-k}$  in the notation for  $\overline{J}_{\mathcal{C}(k)}(\mathbf{w}_k)$  for the sake of brevity. Therefore, minimizing (21) is equivalent to minimizing the original cost (18) or (9) over  $\mathbf{w}_k$ . However the second term (21) still requires information from nodes  $\ell$  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:

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

Usually, especially in the context of adaptive learning in a non-stationary environment, the weighting matrices  $\overline{\mathbf{R}}_\ell$  are unavailable since the covariance matrices  $\mathbf{R}_{x,\ell}$  at each node may not be known beforehand. Following an argument based on the Rayleigh-Ritz characterization of eigenvalues, it was explained in [2] that a useful strategy is to replace each matrix  $\overline{\mathbf{R}}_\ell$  by a weighted multiple of the identity matrix, say, as:

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

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 global cost function (22) can then be relaxed to the following form:

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

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 approximation introduced in steps.

#### 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  $\mathbf{w}_k(n)$  denote the estimate for  $\mathbf{w}_k$  at iteration  $n$ . Using a constant step-size  $\mu$  for each node, the update relation would take the following form:

$$\begin{aligned} \mathbf{w}_k(n+1) = & \mathbf{w}_k(n) - \mu \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k} (\mathbf{R}_{x,\ell} \mathbf{w}_k(n) - \mathbf{p}_{x d,k}) - \mu \eta \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \rho_{k\ell} (\mathbf{w}_k(n) - \mathbf{w}_\ell(n)) \\ & - \mu \sum_{\ell \in \mathcal{N}_k^- \cap \mathcal{C}(k)} b_{\ell k} (\mathbf{w}_k(n) - \mathbf{w}_\ell^o) \end{aligned} \quad (25)$$

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

$$\boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) - \mu \left( \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k} (\mathbf{R}_{x,\ell} \mathbf{w}_k(n) - \mathbf{p}_{x d,k}) + \eta \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \rho_{k\ell} (\mathbf{w}_k(n) - \mathbf{w}_\ell(n)) \right) \quad (26)$$

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

Following the same line of reasoning from [2] 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  $\mathbf{w}_\ell^o$  in (27) since the latter is unavailable and  $\boldsymbol{\psi}_\ell(n+1)$  is an intermediate estimate for it



that is available at node  $\ell$  at time  $n + 1$ . In addition, again in step (27), we replace  $\mathbf{w}_k(n)$  by  $\psi_k(n + 1)$  since it is a better estimate obtained by incorporating information from the neighbors according to (26). Step (27) then becomes

$$\mathbf{w}_k(n + 1) = \left( 1 - \mu \sum_{\ell \in \mathcal{N}_k^- \cap \mathcal{C}(k)} b_{\ell k} \right) \psi_k(n + 1) + \mu \sum_{\ell \in \mathcal{N}_k^- \cap \mathcal{C}(k)} b_{\ell k} \psi_\ell(n + 1). \quad (28)$$

The coefficients in (28) can be redefined as:

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

It can be observed that the entries  $\{a_{\ell k}\}$  are nonnegative for all  $\ell$  and  $k$  (including  $a_{kk}$ ) for sufficiently small step-size. Moreover, the matrix  $\mathbf{A}$  with  $(\ell, k)$ -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:

$$\begin{aligned} \psi_k(n + 1) &= \mathbf{w}_k(n) - \mu \left( \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k} (\mathbf{R}_{x,\ell} \mathbf{w}_k(n) - \mathbf{p}_{x,d,k}) + \eta \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \rho_{k\ell} (\mathbf{w}_k(n) - \mathbf{w}_\ell(n)) \right) \\ \mathbf{w}_k(n + 1) &= \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} a_{\ell k} \psi_\ell(n + 1) \end{aligned} \quad (30)$$

At each instant  $n + 1$ , node  $k$  updates the intermediate value  $\psi_k(n + 1)$  with a local steepest descent iteration. This step involves a regularization term in the case where the set of inter-cluster neighbors of node  $k$  is not empty. Next, an aggregation step is performed where node  $k$  combines its intermediate value  $\psi_k(n + 1)$  with the intermediate values  $\psi_\ell(n + 1)$  from its cluster neighbors. 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 [2]. 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:

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

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

There are several ways to select these coefficients such as using the averaging rule or the Metropolis rule (see [2] for a listing of these and other choices).

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  $\mathcal{N}_k \setminus \mathcal{C}(k) = \emptyset$  for all  $k$ , so that expression (33) reduces to the diffusion adaptation strategy [2], [16] described in Algorithm 2.

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  $\mathcal{N}_k \setminus \mathcal{C}(k) = \mathcal{N}_k^-$  for all  $k$ , the algorithm and (34) degenerate into Algorithm 3. Interestingly, this is the instantaneous gradient counterpart of equation (12) for each node.

---

**Algorithm 1:** Diffusion LMS for clustered multitask networks
 

---

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

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


---

---

**Algorithm 2:** Diffusion LMS for single-task networks [15], [16].
 

---

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

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


---

---

**Algorithm 3:** Diffusion LMS for multitask networks
 

---

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

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) + \mu \left( d_k(n) - \mathbf{x}_k^\top(n) \mathbf{w}_k(n) \right) \mathbf{x}_k(n) + \eta \mu \sum_{\ell \in \mathcal{N}_k^-} \rho_{k\ell} (\mathbf{w}_\ell(n) - \mathbf{w}_k(n)) \quad (35)$$


---

## 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  $\mathbf{w}(n)$ ,  $\mathbf{w}^*$  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.,

$$\mathbf{w}(n) = \begin{pmatrix} \mathbf{w}_1(n) \\ \vdots \\ \mathbf{w}_N(n) \end{pmatrix}, \quad \mathbf{w}^* = \begin{pmatrix} \mathbf{w}_1^* \\ \vdots \\ \mathbf{w}_N^* \end{pmatrix}, \quad \boldsymbol{\psi}(n) = \begin{pmatrix} \boldsymbol{\psi}_1(n) \\ \vdots \\ \boldsymbol{\psi}_N(n) \end{pmatrix} \quad (36)$$

with  $\mathbf{w}_k^* = \mathbf{w}_{\mathcal{C}(k)}^*$ . The weight error vector for each node  $k$  at iteration  $n$  is defined by  $\mathbf{v}_k(n) = \mathbf{w}_k(n) - \mathbf{w}_k^*$ . The weight error vectors  $\mathbf{v}_k(n)$  are also stacked on top of each other to get the block weight error vector defined as follows:

$$\mathbf{v}(n) = \begin{pmatrix} \mathbf{v}_1(n) \\ \vdots \\ \mathbf{v}_N(n) \end{pmatrix} \quad (37)$$

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

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

A direct consequence is that  $\mathbf{x}_k(n)$  is independent of  $\mathbf{v}_\ell(m)$  for all  $\ell$  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. Moreover, various analyses in the literature have already shown that performance results obtained under this assumption match well the actual performance of adaptive algorithms when the step-size is sufficiently small [55].

### A. Mean error behavior analysis

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

$$\begin{aligned} d_\ell(n) - \mathbf{x}_\ell^\top(n) \mathbf{w}_k(n) &= \mathbf{x}_\ell^\top(n) \mathbf{w}_k^* + z_\ell(n) - \mathbf{x}_\ell^\top(n) \mathbf{w}_k(n) \\ &= z_\ell(n) - \mathbf{x}_\ell^\top(n) \mathbf{v}_k(n) \end{aligned} \quad (38)$$

because  $\mathbf{w}_\ell^* = \mathbf{w}_k^*$  for all  $\ell \in \mathcal{N}_k \cap \mathcal{C}(k)$ . Subtracting  $\mathbf{w}_k^*$  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

$$\boldsymbol{\psi}(n+1) - \mathbf{w}^* = \mathbf{v}(n) - \mu \mathbf{H}_x(n) \mathbf{v}(n) + \mu \mathbf{p}_{zx}(n) - \mu \eta \mathbf{Q} (\mathbf{v}(n) + \mathbf{w}^*) \quad (39)$$

where

$$\mathbf{Q} = \mathbf{I}_{LN} - \mathbf{P} \otimes \mathbf{I}_L, \quad (40)$$

with  $\otimes$  denoting the Kronecker product, and  $\mathbf{P}$  the  $N \times N$  matrix with  $(k, \ell)$ -th entry  $\rho_{k\ell}$ . Moreover, the matrix  $\mathbf{H}_x(n)$  is block diagonal of size  $LN \times LN$  defined as

$$\mathbf{H}_x(n) = \text{diag} \left\{ \sum_{\ell \in \mathcal{N}_1 \cap \mathcal{C}(1)} c_{\ell 1} \mathbf{x}_\ell(n) \mathbf{x}_\ell^\top(n), \dots, \sum_{\ell \in \mathcal{N}_N \cap \mathcal{C}(N)} c_{\ell N} \mathbf{x}_\ell(n) \mathbf{x}_\ell^\top(n) \right\}, \quad (41)$$

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

$$\mathbf{p}_{zx}(n) = \left( \sum_{\ell \in \mathcal{N}_1 \cap \mathcal{C}(1)} c_{\ell 1} \mathbf{x}_\ell^\top(n) z_\ell(n), \dots, \sum_{\ell \in \mathcal{N}_N \cap \mathcal{C}(N)} c_{\ell N} \mathbf{x}_\ell^\top(n) z_\ell(n) \right)^\top. \quad (42)$$

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

$$\mathbf{w}(n+1) = \mathbf{A}_I^\top \boldsymbol{\psi}(n+1) \quad (43)$$

Subtracting  $\mathbf{w}^*$  from both sides of the above expression, and using equation (39), the update relation can be written in a single expression as follows

$$\mathbf{v}(n+1) = \mathbf{A}_I^\top [\mathbf{I}_{LN} - \mu (\mathbf{H}_x(n) + \eta \mathbf{Q})] \mathbf{v}(n) + \mu \mathbf{A}_I^\top \mathbf{p}_{zx}(n) - \mu \eta \mathbf{A}_I^\top \mathbf{Q} \mathbf{w}^* \quad (44)$$

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

$$E\{\mathbf{v}(n+1)\} = \mathbf{A}_I^\top [\mathbf{I}_{LN} - \mu (\mathbf{H}_R + \eta \mathbf{Q})] E\{\mathbf{v}(n)\} - \mu \eta \mathbf{A}_I^\top \mathbf{Q} \mathbf{w}^* \quad (45)$$

where

$$\mathbf{H}_R \triangleq E\{\mathbf{H}_x(n)\} = \text{diag} \{ \mathbf{R}_1, \dots, \mathbf{R}_N \} \quad (46)$$

with

$$\mathbf{R}_k = \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k} \mathbf{R}_{x,\ell}. \quad (47)$$

*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

$$\rho\left(\mathbf{A}_I^\top [\mathbf{I}_{LN} - \mu(\mathbf{H}_R + \eta\mathbf{Q})]\right) < 1 \quad (48)$$

where  $\rho(\cdot)$  denotes the spectral radius of its matrix argument. A *sufficient* condition for (48) to hold is to choose  $\mu$  such that

$$0 < \mu < \frac{2}{\max_k \{\lambda_{\max}(\mathbf{R}_k)\} + 2\eta} \quad (49)$$

where  $\lambda_{\max}(\cdot)$  denotes the maximum eigenvalue of its matrix argument. In that case, it follows from (45) that the asymptotic mean bias is given by

$$\lim_{n \rightarrow \infty} E\{\mathbf{v}(n)\} = \mu\eta \left\{ \mathbf{A}_I^\top [\mathbf{I}_{LN} - \mu(\mathbf{H}_R + \eta\mathbf{Q})] - \mathbf{I}_{LN} \right\}^{-1} \mathbf{A}_I^\top \mathbf{Q} \mathbf{w}^*. \quad (50)$$

*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 [2] for definition and properties of the norm):

$$\rho\left(\mathbf{A}_I^\top [\mathbf{I}_{LN} - \mu(\mathbf{H}_R + \eta\mathbf{Q})]\right) \leq \|\mathbf{A}_I^\top (\mathbf{I}_{LN} - \mu(\mathbf{H}_R + \eta\mathbf{Q}))\|_{b,\infty} \quad (51)$$

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

$$\begin{aligned} \|\mathbf{A}_I^\top [\mathbf{I}_{LN} - \mu(\mathbf{H}_R + \eta\mathbf{Q})]\|_{b,\infty} &\leq \|\mathbf{A}_I^\top\|_{b,\infty} \cdot \|\mathbf{I}_{LN} - \mu(\mathbf{H}_R + \eta\mathbf{Q})\|_{b,\infty} \\ &= \|\mathbf{I}_{LN} - \mu(\mathbf{H}_R + \eta\mathbf{Q})\|_{b,\infty} \\ &\leq \|\mathbf{I}_{LN} - \mu\mathbf{H}_R - \mu\eta\mathbf{I}_{LN}\|_{b,\infty} + \mu\eta\|\mathbf{P} \otimes \mathbf{I}_L\|_{b,\infty} \end{aligned} \quad (52)$$

using the definition  $\mathbf{Q} = (\mathbf{I}_N - \mathbf{P}) \otimes \mathbf{I}_L$  and the triangle inequality. Now, it holds that

$$\|\mathbf{P} \otimes \mathbf{I}_L\|_{b,\infty} = \|\mathbf{P}\|_\infty = 1 \quad (53)$$

because  $\mathbf{P}$  is a right stochastic matrix according to condition (14). Furthermore, since  $(1 - \mu\eta)\mathbf{I}_{LN} - \mu\mathbf{H}_R$  is a block diagonal Hermitian matrix, its block maximum norm is equal to its spectral radius [2], namely,

$$\|(1 - \mu\eta)\mathbf{I}_{LN} - \mu\mathbf{H}_R\|_{b,\infty} = \rho((1 - \mu\eta)\mathbf{I}_{LN} - \mu\mathbf{H}_R). \quad (54)$$

Using (53)–(54) in (52) we conclude that a sufficient condition for mean stability is to require

$$\rho((1 - \mu\eta)\mathbf{I}_{LN} - \mu\mathbf{H}_R) + \mu\eta \leq 1, \quad (55)$$

which yields condition (49). ■

### 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 (44):

$$\begin{aligned} \mathbf{B}(n) &= \mathbf{A}_I^\top [\mathbf{I}_{LN} - \mu(\mathbf{H}_x(n) + \eta\mathbf{Q})] \\ \mathbf{g}(n) &= \mathbf{A}_I^\top \mathbf{p}_{zx}(n) \\ \mathbf{r} &= \mathbf{A}_I^\top \mathbf{Q} \mathbf{w}^* \end{aligned} \quad (56)$$

so that

$$\mathbf{v}(n+1) = \mathbf{B}(n) \mathbf{v}(n) + \mu \mathbf{g}(n) - \mu \eta \mathbf{r}. \quad (57)$$

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

$$\begin{aligned} E\{\|\mathbf{v}(n+1)\|_{\Sigma}^2\} &= E\{\|\mathbf{v}(n)\|_{\Sigma'}^2\} + \mu^2 \text{trace}\{\Sigma E\{\mathbf{g}(n)\mathbf{g}^{\top}(n)\}\} + \mu^2 \eta^2 \|\mathbf{r}\|_{\Sigma}^2 \\ &\quad - 2\mu \eta \mathbf{r}^{\top} \Sigma \mathbf{B} E\{\mathbf{v}(n)\} \end{aligned} \quad (58)$$

where

$$\mathbf{B} \triangleq \{\mathbf{B}(n)\} = \mathbf{A}_I^{\top} [(I_{LN} - \mu(\mathbf{H}_R + \eta \mathbf{Q}))] \quad (59)$$

$$\Sigma' \triangleq E\{\mathbf{B}^{\top}(n) \Sigma \mathbf{B}(n)\} \quad (60)$$

In expression (58), the freedom in selecting  $\Sigma$  will allow us to derive several performance metrics. Let

$$\begin{aligned} \mathbf{G} &= E\{\mathbf{g}(n)\mathbf{g}^{\top}(n)\} \\ &= \mathbf{A}_I^{\top} \mathbf{C}_I^{\top} \text{diag}\{\sigma_{z,1}^2 \mathbf{R}_{x,1}, \dots, \sigma_{z,N}^2 \mathbf{R}_{x,N}\} \mathbf{C}_I \mathbf{A}_I \end{aligned} \quad (61)$$

where  $\mathbf{C}_I = \mathbf{C} \otimes \mathbf{I}_L$ . Then, relation (58) can be rewritten as

$$E\{\|\mathbf{v}(n+1)\|_{\Sigma}^2\} = E\{\|\mathbf{v}(n)\|_{\Sigma'}^2\} + \mu^2 \text{trace}\{\Sigma \mathbf{G}\} + \mu^2 \eta^2 \|\mathbf{r}\|_{\Sigma}^2 - 2\mu \eta \mathbf{r}^{\top} \Sigma \mathbf{B} E\{\mathbf{v}(n)\} \quad (62)$$

We would like to show 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  $\Sigma$  and  $\Sigma'$  on both sides of (62) are different, which means that (62) 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 the matrices  $\Sigma$  and  $\Sigma'$  by  $\boldsymbol{\sigma} = \text{vec}(\Sigma)$  and  $\boldsymbol{\sigma}' = \text{vec}(\Sigma')$ , it can be verified that the relation between them can be expressed as the following linear transformation:

$$\boldsymbol{\sigma}' = \mathbf{K} \boldsymbol{\sigma} \quad (63)$$

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

$$\begin{aligned} \mathbf{K} &= E\{\mathbf{B}^{\top}(n) \otimes \mathbf{B}^{\top}(n)\} \\ &= \mathbf{A}_I \otimes \mathbf{A}_I - \mu(\mathbf{H}_R + \eta \mathbf{Q})^{\top} \mathbf{A}_I \otimes \mathbf{A}_I - \mu \mathbf{A}_I \otimes (\mathbf{H}_R + \eta \mathbf{Q})^{\top} \mathbf{A}_I \\ &\quad + \mu^2 E\{(\mathbf{H}(n) + \eta \mathbf{Q})^{\top} \mathbf{A}_I \otimes (\mathbf{H}(n) + \eta \mathbf{Q})^{\top} \mathbf{A}_I\}. \end{aligned} \quad (64)$$

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

$$\mathbf{K} \approx \mathbf{B}^{\top} \otimes \mathbf{B}^{\top}. \quad (65)$$

Finally, let us define  $\mathbf{f}(\boldsymbol{\sigma}, E\{\mathbf{v}(n)\})$  as the last two terms on the right hand side of (62), i.e.,

$$\mathbf{f}(\boldsymbol{\sigma}, E\{\mathbf{v}(n)\}) \triangleq \mu^2 \eta^2 \|\mathbf{r}\|_{\boldsymbol{\sigma}}^2 - 2\mu \eta (\mathbf{B} E\{\mathbf{v}(n)\} \otimes \mathbf{r})^{\top} \boldsymbol{\sigma}. \quad (66)$$

For notational convenience, we are omitting the argument  $\mathbf{r}$  of  $\mathbf{f}$  since it is deterministic. Equation (62) can be expressed as follows:

$$E\{\|\mathbf{v}(n+1)\|_{\boldsymbol{\sigma}}^2\} = E\{\|\mathbf{v}(n)\|_{\mathbf{K}\boldsymbol{\sigma}}^2\} + \mu^2 \text{vec}(\mathbf{G}^{\top})^{\top} \boldsymbol{\sigma} + \mathbf{f}(\boldsymbol{\sigma}, E\{\mathbf{v}(n)\}) \quad (67)$$

where we will be using the notations  $\|\cdot\|_{\Sigma}$  and  $\|\cdot\|_{\sigma}$  interchangeably.

*Theorem 2:* (Mean-square stability) Assume data model (1) and Assumption 1 hold. Assume further that the step-size  $\mu$  is sufficiently small such that approximation (49) is justified by neglecting higher-order powers of  $\mu$ , and relation (67) 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  $\mathbf{K}$  is stable. Under approximation (65), the stability of  $\mathbf{K}$  is guaranteed by sufficiently small step-sizes that also satisfy (49).

*Proof:* Iterating recursion (67) starting from  $n = 0$ , we find that

$$E\{\|\mathbf{v}(n+1)\|_{\sigma}^2\} = E\{\|\mathbf{v}(0)\|_{\mathbf{K}^{n+1}\sigma}^2\} + \mu^2 \text{vec}(\mathbf{G}^{\top})^{\top} \sum_{i=0}^n \mathbf{K}^i \sigma + \sum_{i=0}^n \mathbf{f}(\mathbf{K}^i \sigma, E\{\mathbf{v}(n-i)\}) \quad (68)$$

with initial condition  $\mathbf{v}(0) = \mathbf{w}(0) - \mathbf{w}^*$ . Provided that  $\mathbf{K}$  is stable, the first and the second term on the RHS of (68) converge as  $n \rightarrow \infty$ , to zero for the former, and to a finite value for the latter. Consider now the third term on the RHS of (68). We know from (45) that  $E\{\mathbf{v}(n)\}$  is uniformly bounded because (45) is a BIBO stable recursion with a bounded driving term  $-\mu\eta \mathbf{A}_I^{\top} \mathbf{Q} \mathbf{w}^*$ . Moreover, from (66), the expression for  $\mathbf{f}(\mathbf{K}^i \sigma, E\{\mathbf{v}(n-i)\})$  can be written as

$$\mathbf{f}(\mathbf{K}^i \sigma, E\{\mathbf{v}(n-i)\}) = \mu^2 \eta^2 \text{vec}\{\mathbf{r}\mathbf{r}^{\top}\}^{\top} \mathbf{K}^i \sigma - 2\mu\eta (\mathbf{B} E\{\mathbf{v}(n-i)\} \otimes \mathbf{r})^{\top} \mathbf{K}^i \sigma. \quad (69)$$

We further know that  $\mathbf{K}$  is stable. Therefore, there exists a matrix norm [2], denoted by  $\|\cdot\|_{\rho}$  such that  $\|\mathbf{K}\|_{\rho} = c_{\rho} < 1$ . Applying this norm to  $\mathbf{f}$  and using the triangular inequality, we can deduce that

$$|\mathbf{f}(\mathbf{K}^i \sigma, E\{\mathbf{v}(n-i)\})| < \nu c_{\rho}^i \quad (70)$$

for some positive finite constant  $\nu$ . It follows that the sum appearing as the right-most term in (68) converges as  $n \rightarrow \infty$ . We conclude that  $E\{\|\mathbf{v}(n+1)\|_{\sigma}^2\}$  converges to a bounded value as  $n \rightarrow \infty$ , and the algorithm is said to be mean-square stable.  $\blacksquare$

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

$$\zeta(n+1) = \zeta(n) + \frac{1}{N} \left( \mu^2 \text{vec}(\mathbf{G}^{\top})^{\top} \mathbf{K}^n \text{vec}(\mathbf{I}_{LN}) - E\{\|\mathbf{v}(0)\|_{(\mathbf{I}_{(NL)^2} - \mathbf{K})\mathbf{K}^n \text{vec}(\mathbf{I}_{LN})}^2\} + \mu^2 \eta^2 \|\mathbf{r}\|_{\mathbf{K}^n \text{vec}(\mathbf{I}_{LN})}^2 \right. \\ \left. - 2\mu\eta (\mathbf{\Gamma}(n) + (\mathbf{B} E\{\mathbf{v}(n)\} \otimes \mathbf{r})^{\top} \text{vec}(\mathbf{I}_{LN})) \right) \quad (71)$$

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

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

*Proof:* Comparing (68) at instants  $n+1$  and  $n$ , we can relate  $E\{\|\mathbf{v}(n+1)\|_{\sigma}^2\}$  to  $E\{\|\mathbf{v}(n)\|_{\sigma}^2\}$  as follows:

$$E\{\|\mathbf{v}(n+1)\|_{\sigma}^2\} = E\{\|\mathbf{v}(n)\|_{\sigma}^2\} + \mu^2 \text{vec}(\mathbf{G}^{\top})^{\top} \mathbf{K}^n \sigma - E\{\|\mathbf{v}(0)\|_{(\mathbf{I}_{(NL)^2} - \mathbf{K})\mathbf{K}^n \sigma}^2\} \\ + \sum_{i=0}^n \mathbf{f}(\mathbf{K}^i \sigma, E\{\mathbf{v}(n-i)\}) - \sum_{i=0}^{n-1} \mathbf{f}(\mathbf{K}^i \sigma, E\{\mathbf{v}(n-1-i)\}) \quad (73)$$

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

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

Introducing the following notation

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

we can reformulate recursion (73) as follows:

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

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

with  $\boldsymbol{\Gamma}(0) = \mathbf{0}_{(LN)^2}$ . To derive the transient curve for the MSD, we replace  $\boldsymbol{\sigma}$  by  $\frac{1}{N} \text{vec}(\mathbf{I}_{LN})$ . ■

*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

$$\zeta^* = \frac{\mu^2}{N} \text{vec}(\mathbf{G}^\top)^\top (\mathbf{I}_{(LN)^2} - \mathbf{K})^{-1} \text{vec}(\mathbf{I}_{LN}) + \mathbf{f}\left(\frac{1}{N} (\mathbf{I}_{(LN)^2} - \mathbf{K})^{-1} \text{vec}(\mathbf{I}_{LN}), E\{\mathbf{v}(\infty)\}\right) \tag{78}$$

where  $E\{\mathbf{v}(\infty)\}$  is determined by expression (50).

*Proof:* The steady-state MSD is the limiting value

$$\zeta^* = \lim_{n \rightarrow \infty} \frac{1}{N} E\{\|\mathbf{v}(n)\|^2\}. \tag{79}$$

From the recursive expression (67) we obtain as  $n \rightarrow \infty$  that

$$\lim_{n \rightarrow \infty} E\{\|\mathbf{v}(n)\|_{(\mathbf{I}_{(NL)^2} - \mathbf{K})\boldsymbol{\sigma}}^2\} = \mu^2 \text{vec}(\mathbf{G}^\top)^\top \boldsymbol{\sigma} + \mathbf{f}(\boldsymbol{\sigma}, E\{\mathbf{v}(\infty)\}). \tag{80}$$

Comparing expressions (79) and (80), we observe that to arrive at the MSD requires us to choose  $\boldsymbol{\sigma}$  to satisfy

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

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 analytical performance models. Then, we provide several application-oriented examples where the proposed distributed learning strategy may find applications. These experiments will illustrate the behavior and the advantage of the proposed strategy.

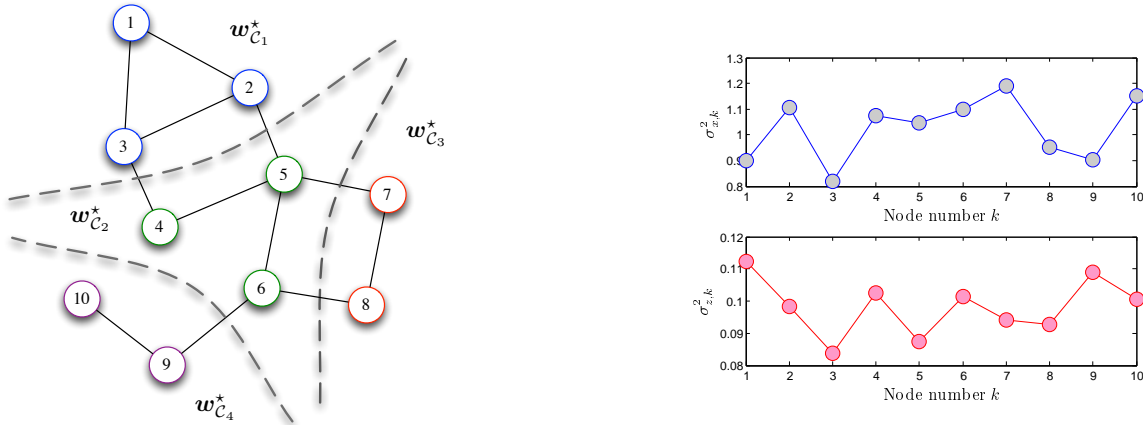


Fig. 2. Experimental setup. Left: network studied in Section VI-A, with 10 nodes divided into 4 different clusters. Right: input signal and noise variances for each node.

### 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 Figure 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  $w_{\mathcal{C}_i}^* = w_o + \delta w_{\mathcal{C}_i}$  were chosen as  $w_o = [0.5, -0.4]^\top$ ,  $\delta w_{\mathcal{C}_1} = [0.0287, -0.005]^\top$ ,  $\delta w_{\mathcal{C}_2} = [0.0234, 0.005]^\top$ ,  $\delta w_{\mathcal{C}_3} = [-0.0335, 0.0029]^\top$ , and  $\delta w_{\mathcal{C}_4} = [0.0224, 0.00347]^\top$ . The regression inputs  $x_k(n)$  were zero-mean  $2 \times 1$  random vectors governed by a Gaussian distribution with covariance matrices  $R_{x,k} = \sigma_{x,k}^2 \mathbf{I}_L$ , and the  $\sigma_{x,k}^2$  shown in the top right plot of Figure 2. The background noises  $z_k(n)$  were i.i.d. zero-mean Gaussian random variables, independent of any other signals. The corresponding variances  $\sigma_{z,k}^2$  are depicted in the bottom right plot of Figure 2.

Regularization strength  $\rho_{k\ell}$  was chosen as  $\rho_{k\ell} = |\mathcal{N}_k \setminus \mathcal{C}(k)|^{-1}$  for  $\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)$ , and  $\rho_{k\ell} = 0$  for any other  $\ell$ . This setting usually leads to asymmetrical regularization weights. We considered the diffusion algorithm with measurement diffusion governed by a uniform matrix  $\mathbf{C}$  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  $\mathbf{A}$  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  $(\mu, \eta)$  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 Figure 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 and 3) the clustered multitask algorithm. 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  $\eta \neq 0$ . The right plot of Figure 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 [42].

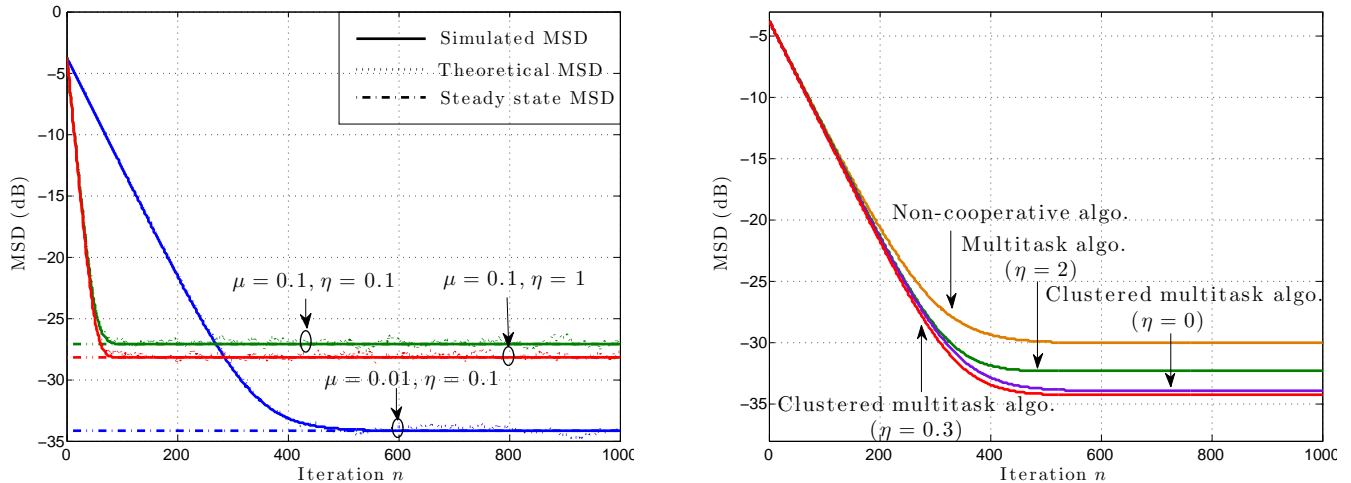


Fig. 3. Network performance illustration. Left: transient and steady-state MSD (model vs. Monte Carlo) for different step-sizes and regularization parameters. Right: performance comparison for different strategies using theoretical models.

### B. Distributed spectrum estimation with multi-antenna devices

We now consider an example of distributed spectral sensing using the clustered multitask diffusion LMS. Cognitive radio systems involve two types of users: primary users (PU) and secondary users (SU). Secondary users are allowed to detect and temporarily occupy unused frequency bands provided that they do not cause harmful interference to primary users [56]. Therefore, secondary users need to sense spectral bands that are occupied by active primary users. In [2], [57], collaborative spectral sensing was studied with single-antenna nodes via the diffusion strategy. In this subsection, we explore a distributed spectral sensing method over the network with multi-antenna devices.

Consider a communication environment consisting of  $N_P$  primary users and  $N_S$  secondary users, each secondary user being equipped with  $N_R$  antennas. We assume that the power spectrum of the signal transmitted by each primary user  $q$  can be represented as a linear combination of  $N_B$  basis functions weighted by the weights  $\alpha_{qm}$ , namely,

$$\mathbf{S}_q(f) = \sum_{m=1}^{N_B} \alpha_{qm} \phi_m(f) \quad (82)$$

The power spectrum of the signal received at the  $\ell$ -th antenna of device  $k$  is given by

$$\mathbf{R}_{k\ell}(f) = \sum_{q=1}^{N_P} p_{k\ell,q} \left( \sum_{m=1}^{N_B} \alpha_{qm} \phi_m(f) \right) + \sigma_{k\ell}^2 \quad (83)$$

where  $p_{k\ell,q}$  is the path loss factor between the primary user  $q$  and the  $\ell$ -th antenna of node  $k$ , and  $\sigma_{k\ell}^2$  is the receiver noise power.

At each time instant, the  $\ell$ -th antenna of device  $k$  gets measurements of the power spectrum  $\mathbf{R}_{k\ell}(f)$  over  $N_F$  frequency samples. Assume that the receiver noise power can be estimated with high accuracy using energy detection over an idle band. Then, at instant  $n$ , we can define the reference signal on this antenna element at the  $j$ -th frequency band by

$$r_{k\ell,j}(n) = \sum_{q=1}^{N_P} p_{k\ell,q} \left( \sum_{m=1}^{N_B} \alpha_{qm} \phi_m(f_j) \right) + z_{k\ell,j}(n) \quad \text{for } j = 1, \dots, N_F \quad (84)$$

where  $z_{k\ell,j}(n)$  is the sampling noise, assumed to be zero-mean Gaussian with variance  $\sigma_{z_{k\ell,j}}^2$ . We denote by  $\Phi$  the  $N_F \times N_B$  matrix of basis functions with  $j$ -th row defined by  $[\phi_1(f_j), \dots, \phi_{N_B}(f_j)]$ , and let

$$\Phi_{k\ell} = [p_{k\ell,1}, \dots, p_{k\ell,N_P}] \otimes \Phi. \quad (85)$$

We collect coefficients  $\alpha_{qm}$  into the vectors  $\alpha_q = [\alpha_{q1}, \dots, \alpha_{qN_B}]^\top$  and  $\alpha = [\alpha_1^\top, \dots, \alpha_{N_P}^\top]^\top$ , measurements  $r_{k\ell,j}(n)$  into the vector  $\mathbf{r}_{k\ell}(n) = [r_{k\ell,1}(n), \dots, r_{k\ell,N_F}(n)]^\top$ , and noise samples  $z_{k\ell,j}(n)$  into the vector  $\mathbf{z}_{k\ell}(n) = [z_{k\ell,1}(n), \dots, z_{k\ell,N_F}(n)]^\top$ . At time instant  $n$ , the model (84) can then be expressed in the following vector form

$$\mathbf{r}_{k\ell}(n) = \Phi_{k\ell} \alpha + \mathbf{z}_{k\ell}(n) \quad (86)$$

for each antenna  $\ell$  of each sensor node  $k$ , at each time instant  $n$ . Inverting the linear model (86) should allow each pair  $(k, \ell)$  to estimate the solution  $\alpha(n)$ . However, in practice, the path loss factor  $p_{k\ell,q}$  cannot be estimated accurately, or even estimated due to failures of synchronization if the power of the received signal is lower than a certain threshold. Thus,  $\hat{\Phi}_{k\ell}$ , depending on the estimated path loss factors  $\hat{p}_{k\ell,q}$ , should be used instead of  $\Phi_{k\ell}$  in the model (86). In the experiment described hereafter, we treat each multi-antenna device as a cluster of  $N_R$  antennas because they are supposed to sense the same local spectrum. In addition, we assume that each cluster is fully connected, i.e., with links between each pair of antennas. Existence of connections between devices depends on their distance from one another. As a consequence, the number of intra-cluster neighbors for each antenna element is  $N_R$ , including itself, and the number of extra-cluster neighbors for each antenna element is  $N_R$  times the number of neighboring devices.

Estimating the spectrum in a noncooperative manner would lead to a local profile of the spectrum occupation, with hidden node effects. We used our algorithm to circumvent this drawback. We considered a connected network composed of  $N_P = 2$  primary users and  $N_S = 10$  secondary users. Each secondary user was equipped with an antenna array of  $N_R$  elements. The topology of the network is shown in Figure 4. The secondary users sampled  $N_F = 80$  frequency bins. We used  $N_B = 16$  Gaussian basis functions defined as

$$\phi_m(f) = e^{-\frac{\|f - f_{c_i}\|^2}{2\sigma_b^2}} \quad (87)$$

with centers  $f_{c_i}$  uniformly distributed along the frequency axis, and variance  $\sigma_b^2 = 0.0025$ . Vectors  $\alpha_1$  and  $\alpha_2$  were arbitrarily set to  $\alpha_1 = [\mathbf{0}_{10}^\top \ 0.4 \ 0.38 \ 0.4 \ \mathbf{0}_3^\top]^\top$  and  $\alpha_2 = [\mathbf{0}_3^\top \ 0.4 \ 0.38 \ 0.4 \ \mathbf{0}_{10}^\top]^\top$ , where  $\mathbf{0}_q$  is a  $q \times 1$  vector of zeros. The path loss factor at instant  $n$  between the primary user  $q$  and the  $\ell$ -th antenna of the secondary user  $k$  was set to  $p_{k\ell,q}(n) = \bar{p}_{k,q} + \delta p_{k\ell,q}(n)$ , where  $\bar{p}_{k,q}$  is the deterministic path loss factor determined by the distance between  $k$  and  $q$  via the free space propagation model, i.e., the received signal power is inversely proportional to the squared distance to the transmitter, and  $\delta p_{k\ell,q}(n)$  is a zero-mean Gaussian variable with standard deviation  $0.2\bar{p}_{k,q}$ , which served as the random fading among antenna elements.

In practice, the path loss factors have to be estimated. We considered that each antenna of the secondary user  $k$  was able to estimate  $\hat{p}_{k\ell,q}(n) = \bar{p}_{k,q}$  if  $\bar{p}_{k,q} \geq p_0$ , otherwise  $\hat{p}_{k\ell,q}(n) = 0$  due to the loss of the synchronization. The noise  $z_{k\ell,j}(n)$  was assumed to be zero-mean Gaussian with the standard deviation 0.01. The antenna elements were considered as fully connected on each device. The information exchange matrix  $\mathbf{C}$  and the combination matrix  $\mathbf{A}$  were arbitrarily set to uniform matrices, with entries thus equal to  $\frac{1}{N_R}$  as each multi-antenna device was considered as a cluster. Within each cluster, the regularization parameter  $\rho_{k\ell}$  was uniformly set to one over  $N_R$  multiplied by the number of neighboring devices.

Our algorithm was run on the multi-antenna device network with different settings. In the left plot of Figure 5, a diffusion strategy was applied within each device, without cooperation between devices ( $\eta = 0$ ), for several numbers  $N_R$  of antennas per device. In the right plot of Figure 5, a diffusion strategy was applied within each device, and cooperation between devices was

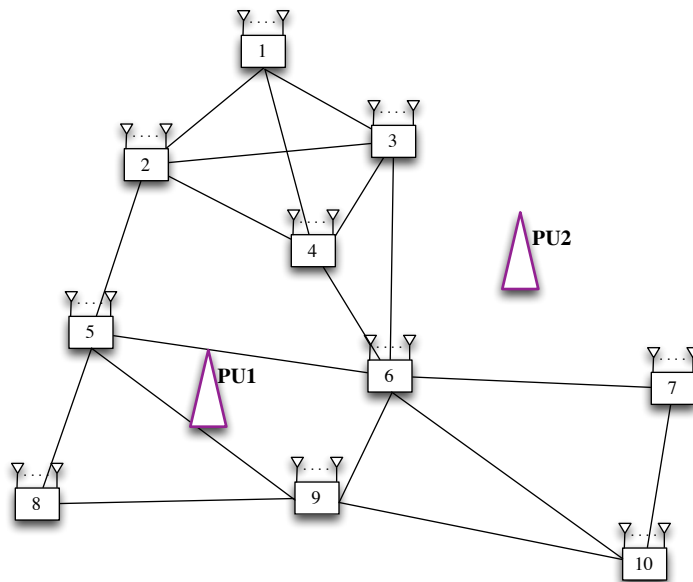


Fig. 4. Cognitive radio network with multi-antenna devices.

promoted ( $\eta = 0.01$ ). The step-sizes were adjusted so that the initial convergence rates were equivalent. It can be observed that increasing the number of antennas, and promoting cooperation between the sets of antennas, allow to improve the performance notably. Figure 6 provides the estimated power spectra for 4 of the 10 devices. Three learning strategies are considered: a non-cooperative strategy ( $\eta = 0$ ) with single-antenna devices ( $N_R = 1$ ), and a cooperative strategy ( $\eta = 0.01$ ) with single-antenna devices ( $N_R = 1$ ) and multi-antenna devices ( $N_R = 4$ ).

Red dashed curves represent ground-truth spectra transmitted by the primary users. An inspection of the device locations in Figure 4 shows that the non-cooperative strategy is highly influenced by the local profile of power spectra. For instance, device 8 was close to PU1 and far from PU2, and it poorly estimated the power spectra generated by PU2. Device 10 was not able to estimate any spectrum because it was out of range for PU1 and PU2. Channel allocation relying on such local spectral profile may have led to a hidden node effect. Cooperative strategies clearly provide more consistent results, and using multiple antennas provides additional gain.

### C. 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 [2], [3]. 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 Figure 7. The target was the arc of a circle with center  $\mathbf{w}_o$ . The angular resolution of the nodes was denoted by  $\delta$ . This means that arcs of the circle with solid angle  $\delta$  were viewed as a single point  $\mathbf{w}_q$  by the cluster  $\mathcal{C}_q$  of nodes within the cone of axis  $(\mathbf{w}_o, \mathbf{w}_q)$ . Note that the distance between each node  $k \in \mathcal{C}_q$  and  $\mathbf{w}_q$  can be expressed in the inner product form

$$r_{kq} = \mathbf{u}_{kq}^\top (\mathbf{w}_q - \mathbf{p}_k) \quad (88)$$

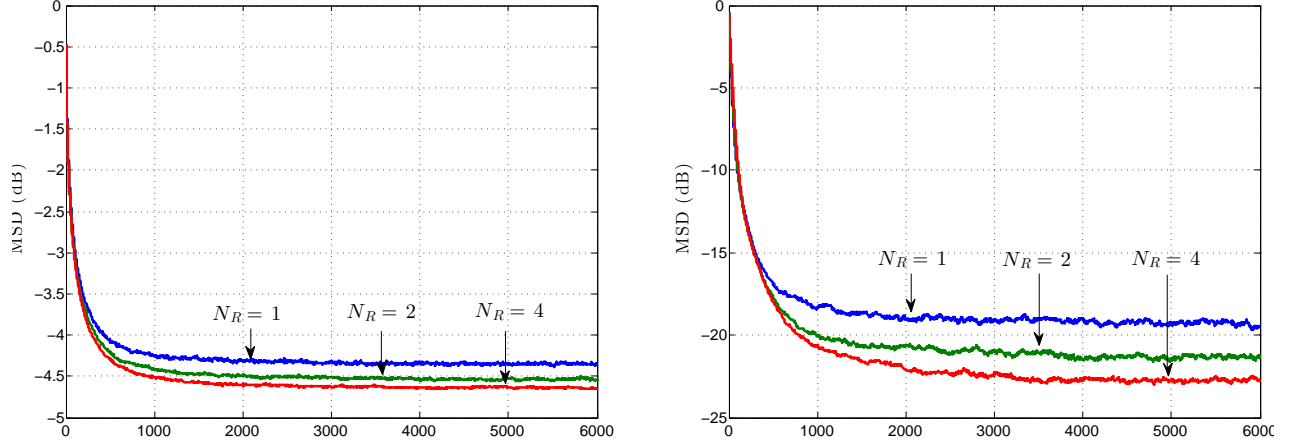


Fig. 5. MSD learning curves. Left: non-cooperating devices ( $\eta = 0$ ). Right: cooperating devices ( $\eta = 0.01$ ).

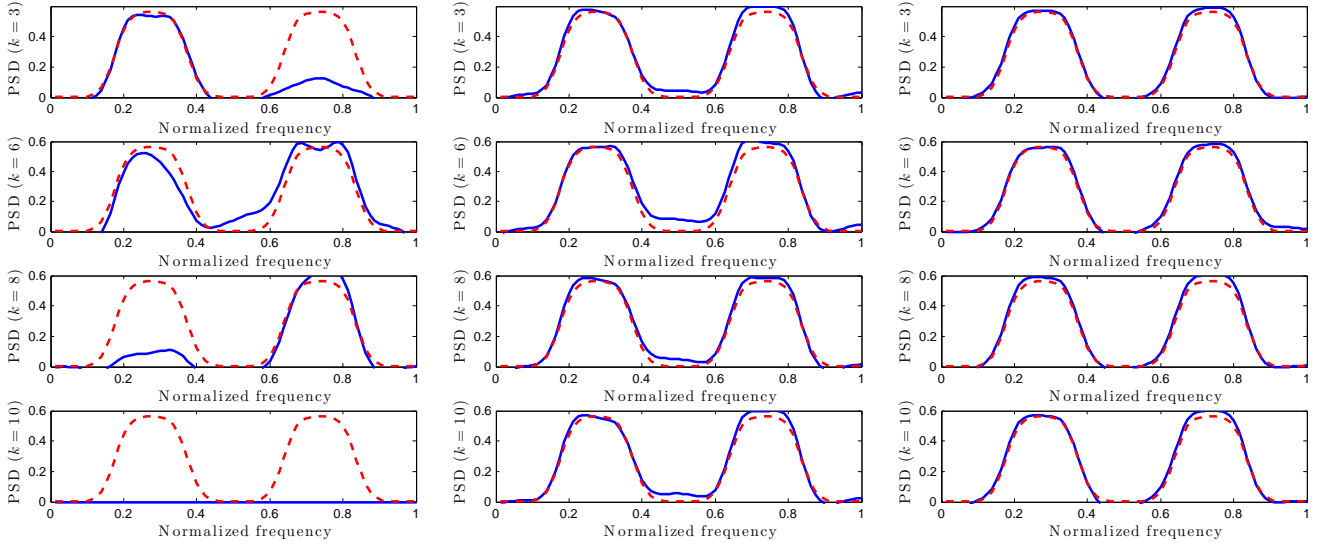


Fig. 6. PSD estimation. From top to bottom: estimate on nodes 3, 6, 8, and 10. From left to right: non-cooperative single-antenna system ( $N_R = 1$ ,  $\eta = 0$ ), cooperative single-antenna system ( $N_R = 1$ ,  $\eta = 0.01$ ), cooperative multi-antenna system ( $N_R = 4$ ,  $\eta = 0.01$ ).

where  $\mathbf{p}_k$  is the location of node  $k$ , and  $\mathbf{u}_{kq}$  is the unit-norm vector pointing from  $\mathbf{p}_k$  to  $\mathbf{w}_q$ . We assumed that sensors were aware of their location  $\mathbf{p}_k$ . Let  $d_{kq} = r_{kq} + \mathbf{u}_{kq}^\top \mathbf{p}_k$ , that is,  $d_{kq} = \mathbf{u}_{kq}^\top \mathbf{w}_q$ . The problem was thus to estimate  $\mathbf{w}_q^*$  from noisy input-output data  $(\mathbf{u}_{kq}(n), d_{kq}(n))$  collected by nodes  $k \in \mathcal{C}_q$ . The model that was thus considered is given by [2]:

$$d_{kq}(n) = \mathbf{u}_{kq}^\top(n) \mathbf{w}_q^* + v_{kq}(n) \quad (89)$$

$$\text{with } \mathbf{u}_{kq}(n) = \mathbf{u}_{kq} + \alpha_k(n) \mathbf{u}_{kq}^\perp + \beta_k(n) \mathbf{u}_{kq}$$

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

The multitask algorithm (33) was used to estimate the coordinates  $w_q^*$  for  $q \in \{1, \dots, Q\}$ , 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 Figure 8 (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 Figure 8 (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  $C = I$ . The combination matrix  $A$  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  $\rho_{k\ell} = |\mathcal{N}_k \setminus \mathcal{C}(k)|^{-1}$  for  $\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)$ , 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  $\ell$  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. Figure 8 (second row) shows one realization of the estimated points  $w_q$  for each arc. The cooperative algorithm clearly outperformed the non-cooperative algorithm. Figure 8 (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.

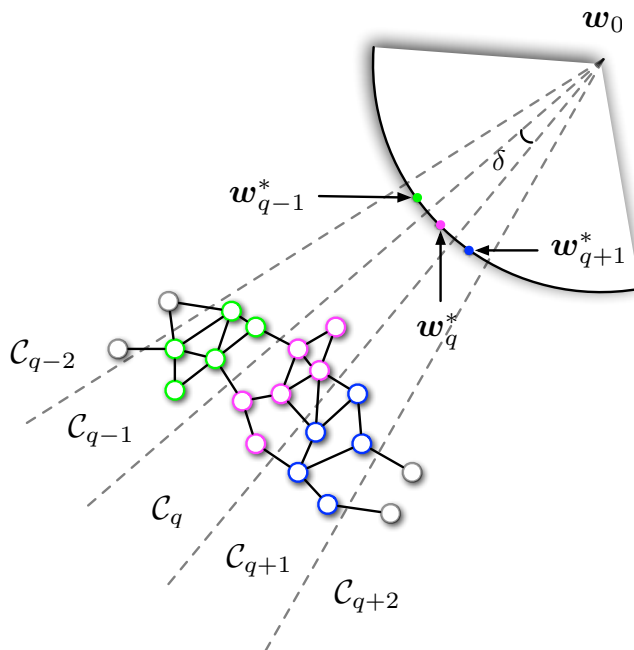


Fig. 7. Target surface localization.

#### D. 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

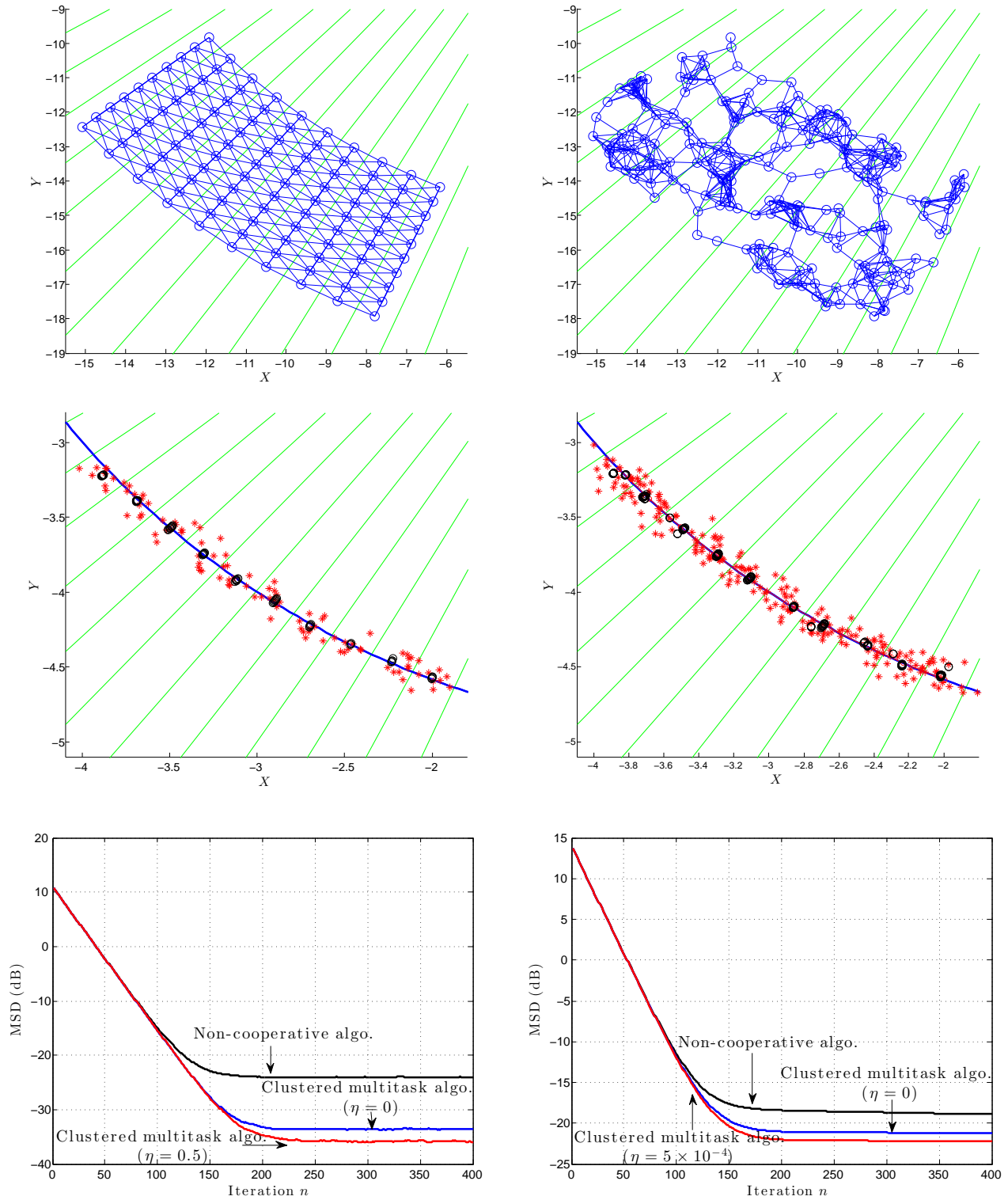


Fig. 8. Target surface localization. Left: uniform network. Right: randomly-distributed network. Row 1: network connectivity, with cluster boundaries in green. Row 2: estimation results, red crosses for the non-cooperative algorithm, black circles for the cooperative algorithm. Row 3: MSD learning curves.

mixture of several spectral signatures of pure materials, termed endmembers, due to limited spatial resolution of devices and diversity of materials [58]. Although nonlinear mixture models have begun to support novel applications [59]–[61], 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 into an  $L \times N$  matrix  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$ , with  $N$  the number of pixels and  $L$  the number of wavelengths. Let  $\mathbf{M}$  be the  $L \times R$  matrix of endmember spectra, with  $R$  the number of endmembers, and  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_N]$  the  $R \times N$  matrix of the abundance vectors of the pixels in  $\mathbf{Y}$ . The linear mixture model is expressed by

$$\mathbf{Y} = \mathbf{M}\mathbf{W} + \mathbf{V} \quad (90)$$

where  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$  is the modeling error matrix. Suppose that the material signatures (matrix  $\mathbf{M}$ ) in a scene have been determined by some endmember extraction algorithm [62]–[64]. 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:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \|\mathbf{Y} - \mathbf{M}\mathbf{W}\|_F^2 + \eta \sum_{k=1}^N \sum_{j \in \mathcal{N}_k} \rho_{kj} \|\mathbf{w}_k - \mathbf{w}_j\|_1 \\ \text{subject to} \quad & \mathbf{w}_k \succeq 0 \quad \text{and} \quad \mathbf{1}^\top \mathbf{w}_k = 1 \quad \text{with} \quad 1 \leq k \leq N, \end{aligned} \quad (91)$$

where  $\|\cdot\|_F^2$  is the matrix Frobenius norm,  $\mathcal{N}_k$  is the set of neighbors of pixel  $k$ ,  $\eta$  is the spatial regularization parameter and  $\rho_{kj}$  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 Figure 9. In order to exploit the spatial correlations, we defined the regularization function  $\Delta(\mathbf{w}_k, \mathbf{w}_j)$  as the  $\ell_1$ -norm of  $\mathbf{w}_k - \mathbf{w}_j$  to promote piecewise constant transitions in the fractional abundance of each endmember among neighboring pixels. Similar regularization can be found in [65], [66]. This led us to the following algorithm:

$$\mathbf{w}_k(n+1) = \mathcal{P}_{\ell_1^+} \left( \mathbf{w}_k(n) + \mu \mathbf{M}^\top (\mathbf{y}_k - \mathbf{M}\mathbf{w}_k(n)) - \mu \eta \sum_{j \in \mathcal{N}_k} \rho_{kj} \text{sgn}(\mathbf{w}_k(n) - \mathbf{w}_j(n)) \right) \quad (92)$$

where we used that the subgradient  $\partial_{\mathbf{x}} \|\mathbf{x}\|_1 = \text{sgn}(\mathbf{x})$ , with  $\text{sgn}(\cdot)$  the component-wise sign function. In this expression,  $\mathcal{P}_{\ell_1^+}(\cdot)$  denotes the iterative operator defined in [67] that projects a vector onto the nonnegative phase of the  $\ell_1$ -ball to satisfy the nonnegativity and sum-to-one constraint in (91). This algorithm clearly contrasts with existing batch approaches based on FISTA [68] and ADMM [69], which cannot easily address large problems (90).

The algorithm (92) was run on a data cube containing  $100 \times 100$  mixed pixels. Each pixel was generated by the linear mixture model (91) using 9 endmember signatures randomly selected from the spectral library ASTER [70]. Each signature of this library has reflectance values measured over 224 spectral bands, uniformly distributed in the interval  $3 - 12 \mu\text{m}$ . The abundance maps of the endmembers are the same as for the image DC2 in [69]. Among these 9 materials, only the 1st, 6th, 8th, and 9th abundances are considered for pictorial illustration in Figure 11. 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

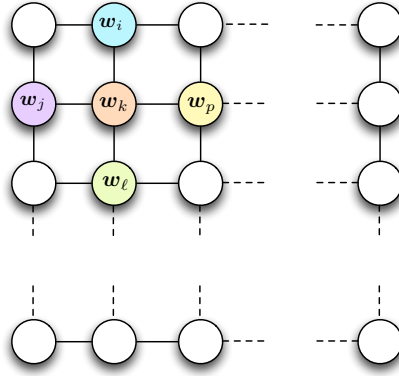


Fig. 9. Hyperspectral image unmixing problem with first-order connections between neighboring nodes.

corrupted by a zero-mean white Gaussian noise  $\mathbf{v}_n$  with an SNR level of 20 dB. In this experiment, the regularization weights  $\rho_{kj}$  were set equal to the normalized spectral similarity:

$$\rho_{kj} = \frac{\theta(\mathbf{y}_k, \mathbf{y}_j)}{\sum_{\ell \in \mathcal{N}_k^-} \theta(\mathbf{y}_k, \mathbf{y}_\ell)} \quad (93)$$

where  $\theta(\mathbf{y}_k, \mathbf{y}_j) = \frac{\mathbf{y}_k^\top \mathbf{y}_j}{\|\mathbf{y}_k\| \|\mathbf{y}_j\|}$ . 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

$$\text{RMSE} = \sqrt{\frac{1}{NR} \sum_{n=1}^N \|\mathbf{w}_n - \mathbf{w}_n^*\|^2}.$$

The RMSE learning curves using algorithm (92), with spatial regularization ( $\eta = 0.05$ ) and without spatial regularization ( $\eta = 0$ ), are depicted in Figure 10. The corresponding abundance distributions are shown in Figure 11. 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  $\ell_2$ -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 to optimize the learning performance and how they can learn the structure of the clusters in real-time.



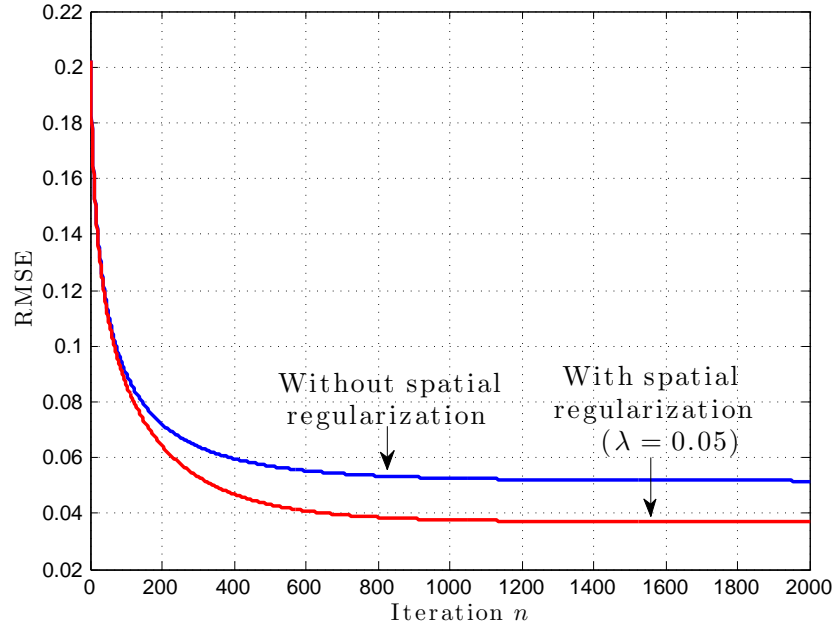


Fig. 10. RMSE curve comparison.

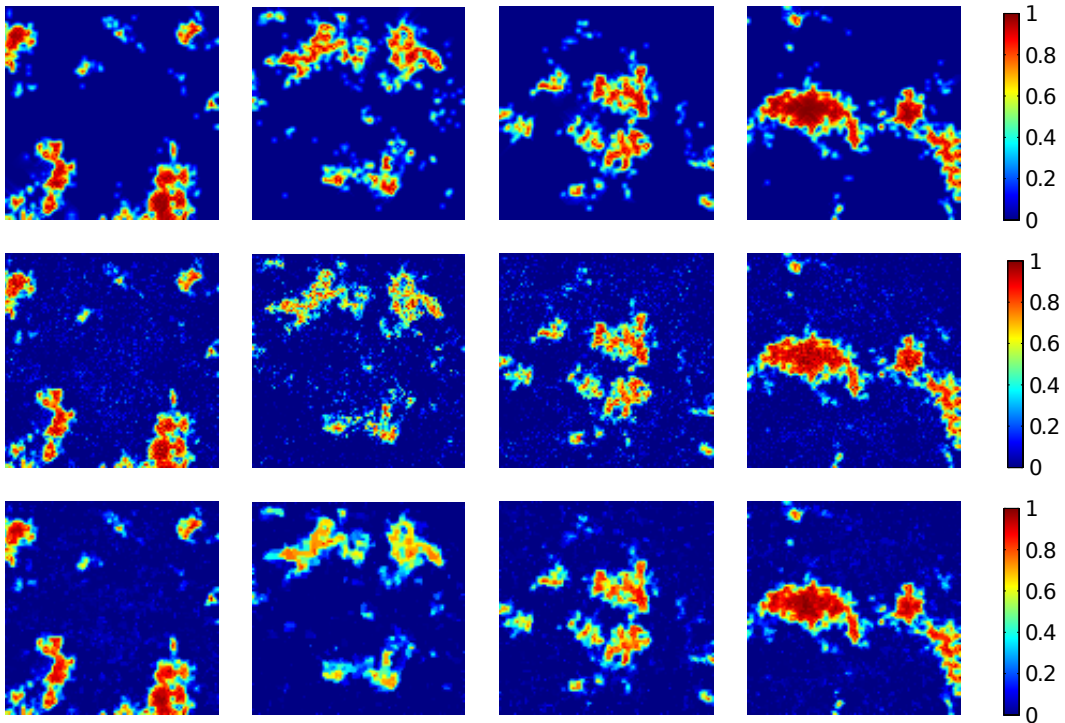


Fig. 11. Abundance maps. From left to right: 1st, 6th, 8th, and 9th abundances. From top to bottom: true abundances, estimated abundances without and with spatial regularization.

## REFERENCES

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

- [2] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, R. Chellapa and S. Theodoridis, Eds., pp. 322–454. Elsevier, 2013. Also available as arXiv:1205.4220 [cs.MA], May 2012.
- [3] S.-Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 649–664, August 2011.
- [4] J. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Transactions on Automatic Control*, vol. 29, no. 1, pp. 42–50, January 1984.
- [5] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *System & Control letters*, vol. 53, no. 9, pp. 65–78, September 2004.
- [6] P. Braca, S. Marano, and V. Matta, "Running consensus in wireless sensor networks," in *Proc. International Conference on Information Fusion (FUSION)*, Cologne, Germany, June-July 2008, pp. 1–6.
- [7] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, January 2009.
- [8] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, January 2009.
- [9] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, August 2011.
- [10] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 913–926, November 1997.
- [11] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, July 2001.
- [12] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal of Selected Topics in Areas in Communications*, vol. 23, no. 4, pp. 798–808, April 2005.
- [13] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with constant step size," *SIAM Journal on Optimization*, vol. 18, no. 1, pp. 29–51, February 2007.
- [14] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, August 2007.
- [15] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.
- [16] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, March 2010.
- [17] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, August 2012.
- [18] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, April 2013.
- [19] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, December 2012.
- [20] S.-Y. Tu and A. H. Sayed, "Adaptive networks with noisy links," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Houston, TX, December 2011, pp. 1–5.
- [21] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers, "Steady-state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 974–979, February 2012.
- [22] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, July 2012.
- [23] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Sparse diffusion lms for distributed adaptive estimation," in *Proc. 37th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012, pp. 3281–3284.

- [24] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4480–4485, August 2012.
- [25] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity-promoting adaptive algorithm for distributed learning," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5412–5425, October 2012.
- [26] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Transactions on Signal Processing*, vol. 61, no. 6, pp. 1419–1433, March 2013.
- [27] S. Chouvardas, K. Kalvakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, October 2011.
- [28] F. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.
- [29] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion bias-compensated RLS estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5212–5224, November 2011.
- [30] P. Chainais and C. Richard, "Distributed dictionary learning over a sensor network," in *Proc. Conférence sur l'Apprentissage Automatique (CAP)*, Lille, France, July 2013, pp. 1–6.
- [31] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network," in *Proc. IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Saint Martin, France, December 2013, pp. 1–4.
- [32] W. Wee and I. Yamada, "A proximal splitting approach to regularized distributed adaptive estimation in diffusion network," in *Proc. 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 5420–5424.
- [33] J. Predd, S. Kulkarni, and H. Vincent Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 59–69, July 2006.
- [34] C. Richard, J.-C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2009.
- [35] P. Honeine, C. Richard, and J.-C. M. Bermudez, "Online nonlinear sparse approximation of functions," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Nice, France, June 2007, pp. 956–960.
- [36] 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 Journal on Wireless Communications and Networking*, vol. 2010, no. 1, pp. 6247372, April 2010.
- [37] 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 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009, pp. 2861–2864.
- [38] P. Honeine, M. Essoloh, C. Richard, and H. Snoussi, "Distributed regression in sensor networks with a reduced-order kernel model," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, New Orleans, LO, USA, 2008, pp. 1–5.
- [39] 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. 44th Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Pacific Grove, CA, USA, 2008, pp. 246–250.
- [40] J. Chen, C. Richard, P. Honeine, and J.-C. M. Bermudez, "Non-negative distributed regression for data inference in wireless sensor networks," in *Proc. 44th Asilomar Conference on Signals, Systems, and Computers (ASILOMAR)*, Pacific Grove, CA, USA, November 2010, pp. 451–455.
- [41] S.-Y. Tu and A. H. Sayed, "Adaptive decision making over complex networks," in *Proc. 46th Asilomar Conference on Signals, Systems, and Computers (ASILOMAR)*, Pacific Grove, CA, USA, November 2012, pp. 525–530.
- [42] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. International Workshop on Cognitive Information Processing (CIP)*, Parador de Baiona, Spain, May 2012, pp. 1–6.

- [43] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, “Distributed incremental-based LMS for node-specific parameter estimation over adaptive networks,” in *Proc. 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 5425–5429.
- [44] J. Chen, L. Tang, J. Liu, and J. Ye, “A convex formulation for leaning shared structures from multiple tasks,” in *Proc. 26th Annual International Conference on Machine Learning (ICML)*, Montreal, Canada, June 2009, pp. 137–144.
- [45] O. Chapelle, P. Shivaswamy, K. Q. Vadrevu, S. Weinberger, Y. Zhang, and B. Tseng, “Multi-task learning for boosting with application to web search ranking,” in *Proc. 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington DC, USA, July 2010, pp. 1189–1198.
- [46] J. Zhou, L. Yuan, J. Liu, and J. Ye, “A multi-task learning formulation for predicting disease progression,” in *Proc. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, USA, August 2011, pp. 814–822.
- [47] J. Chen and C. Richard, “Performance analysis of diffusion LMS in multitask networks,” in *Proc. IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Saint Martin, France, December 2013, pp. 1–4.
- [48] T. Evgeniou and M. Pontil, “Regularized multi-task learning,” in *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, August 2004.
- [49] S. Ji and J. Ye, “An accelerated gradient method for trace norm minimization,” in *Proc. 26th Annual International Conference on Machine Learning (ICML)*, Prague, Czech Republic, August 2009, pp. 457–464.
- [50] J. Zhou, J. Chen, and Ye, “Clustered multi-task learning via alternating structure optimization,” in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., pp. 702–710. 2011.
- [51] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, London : Academic press, 2nd edition, 1995.
- [52] J. B. Rosen, “Existence and uniqueness of equilibrium points for concave n-person games,” *Econometrica: Journal of the Econometric Society*, vol. 33, no. 3, pp. 520–534, 1965.
- [53] S. Theodoridis, K. Slavakis, and I. Yamada, “Adaptive learning in a world of projections,” *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 97–123, January 2011.
- [54] J. Chen, C. Richard, J.-C. M. Bermudez, and P. Honeine, “Nonnegative least-mean-square algorithm,” *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5225–5235, November 2011.
- [55] A. H. Sayed, *Adaptive Filters*, John Wiley & Sons, 2008.
- [56] M. Joseph and J. Q. Maguire, “Cognitive radio: Making software radios more personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, August 1999.
- [57] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, “Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks,” *IEEE Transactions on Signal Processing*, vol. 61, no. 12, pp. 3183–3197, June 2013.
- [58] N. Keshava and J. F. Mustard, “Spectral unmixing,” *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, January 2002.
- [59] 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 approaches,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, April 2012.
- [60] J. Chen, C. Richard, and P. Honeine, “Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model,” *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 480–492, January 2013.
- [61] J. Chen, C. Richard, and P. Honeine, “Estimating abundance fractions of materials in hyperspectral images by fitting a post-nonlinear mixing model,” in *Proc. 5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Gainesville, F, USA, June 2013, pp. 1–4.
- [62] M. E. Winter, “N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data,” in *Proc. SPIE 3753, Imaging Spectrometry V*, October 1999, vol. 266, pp. 266–275.
- [63] J. M. P. Nascimento and J. M. Bioucas-Dias, “Vertex Component Analysis: A fast algorithm to unmix hyperspectral data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, April 2005.

- [64] P. Honeine and C. Richard, “Geometric unmixing of large hyperspectral images: A barycentric coordinate approach,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 6, pp. 2185–2195, June 2012.
- [65] J. Chen, C. Richard, and P. Honeine, “Nonlinear estimation of material abundances in hyperspectral images with  $\ell_1$ -norm spatial regularization,” *IEEE Transactions on Geoscience and Remote Sensing*, 2013, (to appear).
- [66] J. Chen, C. Richard, A. Ferrari, and P. Honeine, “Nonlinear unmixing of hyperspectral data with partially linear least-squares support vector regression,” in *Proc. 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 2174–2178.
- [67] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, “Efficient projections onto the  $\ell_1$ -ball for learning in high dimensions,” in *Proc. 25th International Conference on Machine Learning*, Helsinki, Finland, July 2008, pp. 272–279.
- [68] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, March 2009.
- [69] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, “Total variation spatial regularization for sparse hyperspectral unmixing,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4484–4502, November 2012.
- [70] A. M. Baldridge, S. J. Hook, C. I. Grove, and G. Rivera, “The ASTER spectral library version 2.0,” *Remote Sensing of Environment*, vol. 113, no. 4, pp. 711–715, April 2009.